UNIVERSITY OF SOUTHAMPTON

FACULTY OF PHYSICAL AND APPLIED SCIENCES

School of Electronics and Computer Science

Quantum Error Correction Codes

by

Zunaira Babar

BEng., MSc

Thesis for the degree of Doctor of Philosophy

June 2015

SUPERVISORS
Dr Soon Xin Ng
and Professor Lajos Hanzo

UNIVERSITY OF SOUTHAMPTON

ABSTRACT

FACULTY OF PHYSICAL AND APPLIED SCIENCES
School of Electronics and Computer Science

Doctor of Philosophy

Quantum Error Correction Codes

by Zunaira Babar

Quantum parallel processing techniques are capable of solving certain complex problems at a substantially lower complexity than their classical counterparts. From the perspective of telecommunications, this quantum-domain parallel processing provides a plausible solution for achieving full-search based multi-stream detection, which is vital for future gigabit-wireless systems. The peculiar laws of quantum mechanics have also spurred interest in the absolutely-secure quantum-based communication systems. Unfortunately, quantum decoherence imposes a hitherto insurmountable impairment on the practical implementation of quantum computation as well as on quantum communication systems, which may be overcome with the aid of efficient error correction codes. In this thesis, we design error correction codes for the quantum domain, which is an intricate journey from the realm of classical channel coding theory to that of the Quantum Error Correction Codes (QECCs).

Since quantum-based communication systems are capable of supporting the transmission of both classical and quantum information, we initially focus our attention on the code design for entanglement-assisted classical communication over the quantum depolarizing channel. We conceive an EXtrinsic Information Transfer (EXIT) chart aided near-capacity classical-quantum code design, which invokes a classical Irregular Convolutional Code (IRCC) and a Unity Rate Code (URC) in conjunction with our proposed soft-decision aided SuperDense Code (SD). Hence, it is referred to as an 'IRCC-URC-SD' arrangement. The proposed scheme is intrinsically amalgamated both with 2-qubit as well as 3-qubit SD coding protocols and it is benchmarked against the corresponding entanglement-assisted classical capacity. Since the IRCC-URC-SD scheme is a bit-based design, it incurs a capacity loss. As a further advance, we design a symbol-based concatenated code design, referred to as a symbol-based 'CC-URC-SD', which relies on a single-component classical Convolutional Code (CC). Additionally, for the sake of reducing the associated decoding complexity, we also investigate the impact of the constraint length of the convolutional code on the achievable performance.

Our initial designs, namely IRCC-URC-SD and CC-URC-SD, exploit redundancy in the classical domain. By contrast, QECCs relying on the quantum-domain redundancy are indispensable for conceiving a quantum communication system supporting the transmission of quantum information and also for quantum computing. Therefore, we next provide insights into the transformation from the family of classical codes to the class of quantum codes known as 'Quantum Stabilizer Codes' (QSC), which invoke the classical syndrome decoding. Particularly, we detail the underlying quantum-to-classical isomorphism, which facilitates the design of meritorious families of QECCs from the known classical codes. We further study the syndrome decoding techniques operating over classical channels, which may be exploited for decoding QSCs. In this context, we conceive a syndrome-based block decoding approach for the classical Turbo Trellis Coded Modulation (TTCM), whose performance is investigated for transmission over an Additive White Gaussian Noise (AWGN) channel as well as over an uncorrelated Rayleigh fading channel.

Pursuing our objective of designing efficient QECCs, we next consider the construction of Hashing-bound-approaching concatenated quantum codes. In this quest, we appropriately adapt the conventional non-binary EXIT charts for Quantum Turbo Codes (QTCs) by exploiting the intrinsic quantum-to-classical isomorphism. We further demonstrate the explicit benefit of our EXIT-chart technique for achieving a Hashing-bound-approaching code design. We also propose a generically applicable structure for Quantum Irregular Convolutional Codes (QIRCCs), which can be dynamically adapted to a specific application scenario with the aid of the EXIT charts. More explicitly, we provide a detailed design example by constructing a 10-subcode QIRCC and use it as an outer code in a concatenated quantum code structure for evaluating its performance.

Working further in the direction of iterative code structures, we survey Quantum Low Density Parity Check (QLPDC) codes from the perspective of code design as well as in terms of their decoding algorithms. Furthermore, we propose a radically new class of high-rate row-circulant Quasi-Cyclic QLDPC (QC-QLDPC) codes, which can be constructed from arbitrary row-circulant classical QC-LDPC matrices. We also conceive a modified non-binary decoding algorithm for homogeneous Calderbank-Shor-Steane (CSS)-type QLDPC codes, which is capable of alleviating the problems imposed by the unavoidable length-4 cycles. Our modified decoder outperforms the state-of-the-art decoders in terms of their Word Error Rate (WER) performance, despite imposing a reduced decoding complexity. Finally, we intricately amalgamate our modified decoder with the classic Uniformly-ReWeighted Belief Propagation (URW-BP) for the sake of achieving further performance improvement.

Declaration of Authorship

I, Zunaira Babar, declare that the thesis entitled *Quantum Error Correction Codes* and the work presented in the thesis are both my own, and have been generated by me as the result of my own original research. I confirm that:

- this work was done wholly or mainly while in candidature for a research degree at this University;
- where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated;
- where I have consulted the published work of others, this is always clearly attributed;
- where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work;
- I have acknowledged all main sources of help;
- where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself;
- parts of this work have been published as: [1, 2, 3, 4, 5, 6, 7]

Signed:..

Date:..

Acknowledgements

I would like to take this opportunity to express my deepest gratitude to my supervisors Dr. Soon Xin Ng and Prof. Lajos Hanzo. Without their insightful guidance, continuous support and inexhaustible enthusiasm, I would not have been able to accomplish this formidable task. Their inspirational motivation has helped me grow both as a researcher as well as a person. I would also like to thank the faculty members of the Southampton Wireless Group, namely Dr. Robert G. Maunder, Dr Mohammed El-Hajjar, Dr Rong Zhang, Prof. Lie Liang Yang and Prof. Sheng Chen for the useful discussions/suggestions throughout my research. Special thanks are also due to Dr. Mark Wilde (Louisiana State University) for the valuable discussions. I also extend my gratitude to all my colleagues at Southampton for their continuous support and inspiration. I am also indebted to the School of Electronics and Computer Science, University of Southampton, for supporting my studies under the auspices of the Pan-European Concerto project. Finally, I extend my heartfelt appreciation to my family and friends for their unconditional support and encouragement.

List of Publications

Journal Papers:

1. **Zunaira Babar**, Panagiotis Botsinis, Dimitrios Alanis, Soon Xin Ng and Lajos Hanzo, "Fifteen Years of Quantum LDPC Coding and Improved Decoding Strategies", *IEEE Access (to be submitted).*

2. **Zunaira Babar**, Panagiotis Botsinis, Dimitrios Alanis, Soon Xin Ng and Lajos Hanzo, "Construction of Quantum LDPC Codes from Classical Row-Circulant QC-LDPCs", *IEEE Communications Letters (to be submitted).*

3. **Zunaira Babar**, Panagiotis Botsinis, Dimitrios Alanis, Soon Xin Ng and Lajos Hanzo, "The Road from Classical to Quantum Codes: a Hashing Bound Approaching Design Procedure", *IEEE Access*, vol.3, pp. 146-176, Mar. 2015.

4. **Zunaira Babar**, Soon Xin Ng and Lajos Hanzo, "EXIT-Chart Aided Near-Capacity Quantum Turbo Code Design", *IEEE Transactions on Vehicular Technology*, vol.64, no.3, pp. 866-875, Mar. 2015.

5. **Zunaira Babar**, Soon Xin Ng and Lajos Hanzo, "Near-Capacity Code Design for Entanglement-Assisted Classical Communication over Quantum Depolarizing Channels", *IEEE Transactions on Communications*, vol. 61, no. 12, pp. 4801-4807, Dec. 2013.

6. **Zunaira Babar**, Soon Xin Ng and Lajos Hanzo, "Reduced-Complexity Syndrome-Based TTCM Decoding", *IEEE Communications Letters*, vol. 17, no. 6, pp. 1220-1223, Jun. 2013.

7. Panagiotis Botsinis, Dimitrios Alanis, **Zunaira Babar**, Soon Xin Ng and Lajos Hanzo, "Non-Coherent Quantum Multiple Symbol Differential Detection for Wireless Systems", *IEEE Access*, vol.3, pp. 569-598, May 2015.

8. Mark Wilde, Min-Hsiu Hsieh and **Zunaira Babar**, "Entanglement-Assisted Quantum Turbo Codes", *IEEE Transactions on Information Theory*, vol. 60, no. 2, pp. 1203-1222, Feb. 2014.

9. Panagiotis Botsinis, Dimitrios Alanis, **Zunaira Babar**, Soon Xin Ng and Lajos Hanzo, "Iterative Quantum-Assisted Multi-User Detection for Multi-Carrier Interleave Division Multiple Access Systems", *IEEE Transactions on Communications (under review).*

10. Abdulah Jeza Aljohani, **Zunaira Babar**, Soon Xin Ng, and Lajos Hanzo, "Distributed Source-Channel Coding using Reduced-Complexity Syndrome-Based TTCM", *IEEE Communications Letters (to be submitted).*

11. Dimitrios Alanis, Panagiotis Botsinis, **Zunaira Babar**, Soon Xin Ng and Lajos Hanzo, "Non-Dominated Quantum Iterative Routing Optimization for Wireless Multihop Networks", *IEEE Access (to be submitted).*

Conference Papers:

1. **Zunaira Babar**, Soon Xin Ng and Lajos Hanzo, "EXIT-Chart Aided Code Design for Symbol-Based Entanglement-Assisted Classical Communication over Quantum Channels", *IEEE Vehicular Technology Conference (VTC-Fall)*, Sep. 2014, Vancouver, BC.

2. Hung Nguyen, **Zunaira Babar**, Soon Xin Ng, Matteo Mazzotti, Lorenzo Iacobelli and Lajos Hanzo, "Network Coded MIMO Aided Cooperative Communications in the Ambulance-and-Emergency Area", *International Conference on Selected Topics in Mobile and Wireless Networking (MoWNet)*, Sep. 2014, Rome, IT.

Poster Presentation:

1. **Zunaira Babar**, Panagiotis Botsinis, Dimitrios Alanis, Soon Xin Ng and Lajos Hanzo, "Quantum-Aided Solutions in Wireless Systems", *International Workshop on Quantum Communication Networks*, Jan. 2014, Leeds, UK.

Contents

Abstract **iii**

Declaration of Authorship **v**

Acknowledgement **vii**

List of Publications **ix**

List of Symbols **xvii**

1 Introduction **1**

1.1 Motivation 1

1.2 Historical Overview of QECCs 5

1.2.1 Quantum Convolutional Codes 6

1.2.2 Quantum Low Density Parity Check Codes 9

1.2.3 Quantum Turbo Codes 10

1.2.4 Entanglement-Assisted Quantum Codes 13

1.3 Outline of the Thesis 13

1.4 Novel Contributions of the Thesis 17

2 Preliminaries of Quantum Information **19**

2.1 Introduction 19

2.2 Quantum Bits 19

2.3 N-Qubit Composite System 21

2.4 No-Cloning Theorem 22

2.5 Entanglement 23

2.6 Unitary Operators 24

2.7 The Pauli Group 26

2.8 Quantum Channels 27

2.9 Summary and Conclusions 29

3 Near-Capacity Code Designs for Entanglement-Assisted Classical Communication 31

3.1 Introduction 31

3.2 Review of the Superdense Coding Protocol 33

3.2.1 2-Qubit Superdense Coding 33

3.2.2 N-Qubit Superdense Coding 36

3.3 Entanglement-Assisted Classical Capacity 37

3.4 Bit-Based Code Structure 39

3.5 Near-Capacity Design 42

3.5.1 EXIT Charts 42

3.5.2 Near-Capacity IRCC-URC-SD Design 43

3.6 Results and Discussions I 47

3.6.1 Performance of IRCC-URC-2SD 47

3.6.2 Performance of IRCC-URC-3SD 51

3.7 Symbol-Based Code Structure 54

3.8 Results and Discussions II 55

3.9 Summary and Conclusions 60

4 From Classical to Quantum Error Correction 65

4.1 Introduction 65

4.2 Review of Classical Linear Block Codes 66

4.3 Quantum Stabilizer Codes 68

4.3.1 Stabilizer Formalism 68

4.3.2 Quantum-to-Classical Isomorphism 74

4.3.2.1 Pauli-to-Binary Isomorphism 74

4.3.2.2 Pauli-to-Quaternary Isomorphism 79

4.3.3 Classification of Quantum Errors 83

4.4 Quantum Convolutional Codes 84

4.5 Entanglement-Assisted Quantum Codes 87

4.6 Summary and Conclusions 89

5 Classical Syndrome Decoding 91

5.1 Introduction 91

5.2 Look-Up Table-based Syndrome Decoding 93

5.3 Trellis-based Syndrome Decoding 95

5.3.1 Linear Block Codes 95

5.3.2 Convolutional Codes 98

5.4 Block Syndrome Decoding 105

5.4.1 General Formalism 105

5.4.2 Block Syndrome Decoder for TTCM 105

5.4.2.1 System Model 106

5.4.2.2 Syndrome-Based MAP Decoder 108

5.4.2.3 Error Estimation 109

5.4.2.4 Syndrome-based Blocking 110

5.5 Results and Discussions 110

5.5.1 Performance of BSD-TTCM over AWGN Channel 110

5.5.2 Performance of BSD-TTCM over Uncorrelated Rayleigh Fading Channel 112

5.5.3 Effect of Frame Length on the Performance of BSD-TTCM 116

5.6 Summary and Conclusions 117

6 EXIT-Chart Aided Hashing Bound Approaching Concatenated Quantum Codes 121

6.1 Introduction 121

6.2 Design Objectives 122

6.3 Circuit-Based Representation of Stabilizer Codes 125

6.4 Concatenated Quantum Codes 133

6.4.1 System Model 133

6.4.2 Degenerate Iterative Decoding 135

6.5 EXIT-Chart Aided Quantum Code Design 138

6.6 Results and Discussions I 143

6.6.1 Accuracy of EXIT Chart Predictions 143

6.6.2 Entanglement-Assisted and Unassisted Inner Codes 143

6.6.3 Optimized Quantum Turbo Code Design 147

6.7 Quantum Irregular Convolutional Codes 151

6.8 Results and Discussions II 155

6.9 Summary and Conclusions 158

7 Quantum Low Density Parity Check Codes 163

7.1 Introduction 163

7.2 Quantum LDPC Code Designs 164

7.2.1 Calderbank-Shor-Steane Codes 164

7.2.2 Non-CSS Codes 171

7.2.3 Entanglement-Assisted QLDPC Codes 175

7.3 Iterative Decoding of Quantum LDPC Codes 176

7.3.1 Binary Decoding 176

7.3.2 Non-Binary Decoding 180

7.3.3 Decoding Issues & Heuristic Methods for Improvement 182

7.4 High-Rate QLDPC Codes from Row-Circulant Classical LDPCs 186

7.5 Results and Discussions I 190

7.6 Modified Non-Binary Decoding 193

7.7 Reweighted BP for Graphs Exhibiting Cycles 200

7.8 Results and Discussions II 202

7.8.1 Modified Non-Binary Decoding 202

7.8.2 Uniformly-Reweighted BP 207

7.9 Summary and Conclusions 212

8 Conclusions and Future Directions 217

8.1 Summary and Conclusions 217

8.2 Future Research Directions 228

Appendices 231

A Construction of Syndrome Former 231

A.1 Convolutional Codes 231

A.2 Turbo Trellis Coded Modulation 233

B Simulation of QLDPC Decoding 235

List of Abbreviations 237

Glossary 239

Bibliography 245

List of Symbols

General Notation

- The notation $|.\rangle$ is used to indicate a quantum state. Therefore, $|\psi\rangle$ represents a qubit having the state ψ.

- The notation $|.|$ is used to indicate a magnitude operation. Therefore, $|\alpha|$ represents the magnitude of a complex number α.

- The notation $[.]$ is used to indicate the effective Pauli operation. Therefore, $[\mathcal{P}]$ represents the effective Pauli vector for the Pauli vector $\mathcal{P}$.

- The notation $\star$ is used to indicate the symplectic product.

- The notation $\otimes$ is used to indicate the tensor product.

- The notation $\bigotimes$ is used to indicate the convolution operation.

- The notation $\prod$ is used to indicate the product operation.

- The notation $\sum$ is used to indicate the sum operation.

- The notation $\langle , \rangle$ is used to represent the inner product.

- The GF(4) variables are represented with a $\hat{}$ on top, e.g. $\hat{x}$.

- The notation (n, k) is used for a classical code, while the notation $[n, k]$ is used for a quantum code.

- The superscript ^{T} is used to indicate the matrix transpose operation. Therefore, $\mathbf{x}^T$ represents the transpose of the matrix $\mathbf{x}$.

Special Symbols

a	The number of auxiliary qubits.
$A(.)$	The *a priori* probability.
c	The number of pre-share entangled qubts (ebits).
C	The capacity of a classical channel.
$\mathbb{C}^d$	The d-dimensional Hilbert space.
$\mathrm{C}_Q(.)$	The quantum channel capacity.
$E(.)$	The *extrinsic* probability.
$\mathrm{E}[.]$	The expectation (or time average) operation.
E	The entanglement consumption rate.
$\mathbb{F}_q$	The Galois field $\mathrm{GF}(q)$.
$\mathcal{F}$	The fast fourier transform operation.
$\mathcal{F}^{-1}$	The inverse fast fourier transform operation.
G	The generator matrix.
$\mathcal{G}_n$	The n-qubit Pauli group.
g_i	The ith stabilizer generator.
H	The parity check matrix.
$\mathcal{H}$	The stabilizer group.
$\mathrm{H}_2(.)$	The binary entropy function.
$\mathbf{H}$	The Hadamard gate.
I	The mutual information.
$\mathbf{I}$	The Pauli-**I** operator.
k	The length of information word.
n	The length of codeword.
$\mathcal{L}$	The Pauli error inflicted on the information word.
$\mathbf{M}$	The measurement operation.
p	The channel error (or flip) rate, e.g. channel depolarizing probability.
p^*	The noise limit (or maximum tolerable channel error rate).
$\mathrm{P}(.)$	The probability function.
$\mathcal{P}$	The Pauli error inflicted on the transmitted codeword.
π	The bit or qubit interleaver.
π^{-1}	The bit or qubit de-interleaver.
π_s	The symbol-based interleaver.
π_s^{-1}	The symbol-based de-interleaver.
R_c	The equivalent classical coding rate of a quantum code.
R_Q	The quantum coding rate.
$\mathcal{S}$	The Pauli error inflicted on the auxiliary qubits.
$\mathbf{S}$	The phase gate.
$T[.]$	The transfer function.

$Tr[.]$	The trace operation.
$\mathcal{V}$	The Clifford encoder.
$\sum$	The sum operation.
$T[.]$	The transfer function.
$\mathrm{Tr}[.]$	The trace operation.
$\mathcal{V}$	The Clifford encoder.
$\mathbf{X}$	The Pauli-$\mathbf{X}$ operator.
$\mathbf{Y}$	The Pauli-$\mathbf{Y}$ operator.
$\mathbf{Z}$	The Pauli-$\mathbf{Z}$ operator.

Chapter 1

Introduction

If computers that you build are quantum,
Then spies everywhere will all want 'em.
Our codes will all fail,
And they'll read our email,
Till we get crypto that's quantum, and daunt 'em.
Jennifer and Peter Shor

1.1 Motivation

The laws of quantum mechanics provide a promising solution to our quest for miniaturization and increased processing power, as implicitly predicted by Moore's law formulated four decades ago [8]. This can be attributed to the inherent parallelism associated with the quantum bits (qubits). More specifically, in contrast to the classical bits, which can either assume a value of 0 or 1, qubits can exist in a superposition of the two states. Consequently, while an N-bit classical register can store only a single N-bit value, an N-qubit quantum register can store all the 2^N states concurrently, allowing parallel evaluations of certain functions with regular global structure at a complexity cost that is equivalent to a single classical evaluation [9, 10], as illustrated in Figure 1.1. Therefore, as exemplified by Shor's factorization algorithm [13] and Grover's search algorithm [14], quantum-based computation is capable of solving certain complex problems at a substantially lower complexity, as compared to its classical counterpart. From the perspective of telecommunications, this quantum domain parallel processing seems to be a plausible solution for the massive parallel processing required for achieving joint optimization in large-scale communication systems, for example with the aid of quantum assisted multi-user detection [10, 15, 16] and quantum-assisted routing optimization for self-organizing networks [17]. More explicitly, provided that we can create a sufficiently high number of parallel

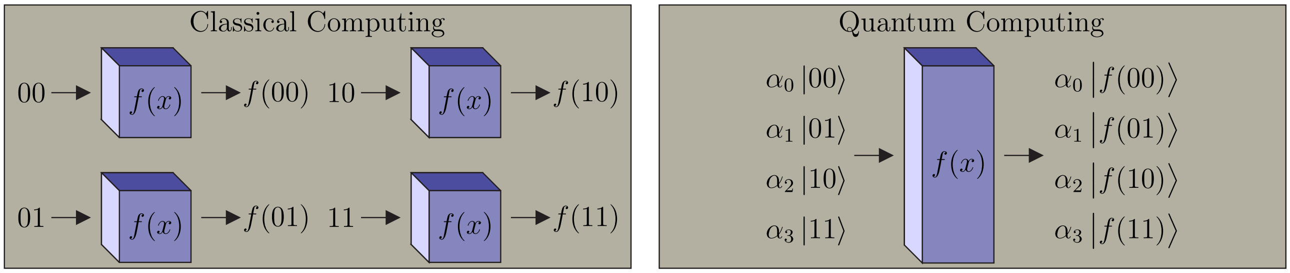


Figure 1.1: Quantum Parallelism: Given a function $f(x)$, which has a regular global structure such that $f(x) : \{0,1\}^2 \rightarrow \{0,1\}^2$, a classical system requires four evaluations to compute $f(x)$ for all possible $x \in \{00, 01, 10, 11\}$. By contrast, since a 2-qubit quantum register can be in a superposition of all the four states concurrently, i.e. $|\psi\rangle = \alpha_0|00\rangle + \alpha_1|01\rangle + \alpha_2|10\rangle + \alpha_3|11\rangle$, quantum computing requires only a single classical evaluation to yield the outcome, which is also in a superposition of all the four possibilities, i.e. $\alpha_0|f(00)\rangle + \alpha_1|f(01)\rangle + \alpha_2|f(10)\rangle + \alpha_3|f(11)\rangle$. However, it is not possible to read all the four states because the quantum register collapses to one of the four superimposed states upon measurement. Nevertheless, we may manipulate the resultant superposition of the four possible states before observing the quantum register for the sake of determining a desired property of the function, as in [11, 12, 13, 14].

streams and that we have a low-complexity full-search-based multi-stream detector, the throughput of the wireless system may be increased linearly with the transmit power [10]. Unfortunately, the associated optimal full-search-based multi-stream detectors of classical systems have an excessive complexity, which increases exponentially both with the number of users as well as the antennas. Since quantum-based computation is capable of solving certain complex problems at a substantially lower complexity as compared to its classical counterpart, quantum parallel processing techniques may be invoked here [10, 15, 16, 17].

Against this background, quantum computation may jeopardize the most trusted methods of classical public key encryption, which derive their security from the computational complexity associated with the underlying mathematical functions. While classical cryptography is at risk of being deciphered due to quantum computing, quantum-based communication is capable of supporting secure data dissemination, where any 'measurement' or 'observation' by an eavesdropper perturbs the quantum superposition, hence intimating the parties concerned [18, 9]. Quantum-based communication has given rise to a new range of security paradigms, which cannot be created using a classical communication system. In this context, quantum key distribution techniques [19, 20], quantum secure direct communication [21, 22, 23] and the recently proposed unconditional quantum location verification [24] are of particular significance. In particular, quantum-based communication supports the transmission of classical as well as quantum information over an absolutely secure quantum channel, which exploits a classical side-channel for detecting eavesdropping.

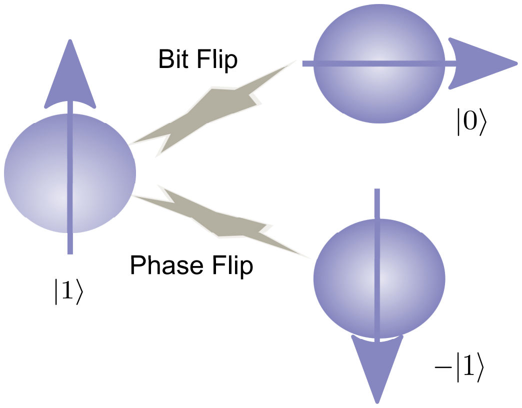


Figure 1.2: Quantum decoherence characterized by bit-flips and phase-flips. *The vertical polarization represents the state* $|1\rangle$*, while the horizontal polarization represents the state* $|0\rangle$.

System Size (cm)	Cosmic Radiation	Room Temperature	Sunlight	Vacuum (10^6 particles/cm^3)	Air
10^{-3}	10^{-7}	10^{-14}	10^{-16}	10^{-18}	10^{-35}
10^{-5}	10^{15}	10^{-3}	10^{-8}	10^{-10}	10^{-23}
10^{-6}	10^{25}	10^{5}	10^{-2}	10^{-6}	10^{-19}

Table 1.1: Coherence time (seconds) under different environmental conditions for systems of varying sizes [27, p. 176].

Unfortunately, a major impediment to the practical realization of quantum computation as well as communication systems is quantum noise, which is conventionally termed as ‘decoherence’ (loss of the coherent quantum state). More explicitly, decoherence is the undesirable interaction of the qubits with the environment [25, 26]. It may be viewed as the undesirable entanglement of qubits with the environment, which perturbs the fragile superposition of states, thus leading to the detrimental effects of noise. The overall decoherence process may be characterized either by bit-flips or phase-flips or in fact possibly both, inflicted on the qubits [25], as depicted in Figure 1.2[1]. The longer a qubit retains its coherent state, known as the coherence time, the better. In particular, the coherence time is affected both by the size of the system as well as the environment. Table 1.1 depicts the coherence time for systems of varying sizes under different environmental conditions, i.e. when subjected to cosmic radiation, room temperature, sunlight, vacuum and air. Ideally, a quantum computing algorithm (and similarly quantum communication) should be designed such that the computational process (or transmission) finishes before the qubits decohere. Since different quantum systems have different

[1] A qubit may be realized in different ways, e.g. two different photon polarizations, different alignments of a nuclear spin, two electronic levels of an atom or the charge/current/energy of a Josephson junction.

System	Coherence Time (s)	Time per Gate Operation (s)	Max. No. of Reliable Operations
Electrons from gold atom	10^{-8}	10^{-14}	10^{6}
Trapped indium atoms	10^{-1}	10^{-14}	10^{13}
Optical microcavity	10^{-5}	10^{-14}	10^{9}
Electron spin	10^{-3}	10^{-7}	10^{4}
Electron quantum dot	10^{-3}	10^{-6}	10^{3}
Nuclear spin	10^{4}	10^{-3}	10^{7}

Table 1.2: Maximum number of reliable operations for various systems [27, p. 177].

coherence times as well as different quantum gate operation times, the maximum number of reliable operations varies for different systems, as illustrated in Table 1.2, where the maximum number of reliable operations is computed as:

$$\text{Max. no. of reliable operations} = \frac{\text{Coherence time}}{\text{Time per gate operation}}. \tag{1.1}$$

We may further observe in Table 1.2 that for the first three systems, which have the same time per gate operation, the system having the longest coherence time, i.e. trapped indium atoms, permits the maximum number of reliable operations. This is also evident from Eq. (1.1). Hence, it is desirable to increase the coherence time for reliable quantum computation. Analogously, a longer coherence time is also necessary for reliable quantum communication.

Recall that quantum-based communication systems support the transmission of both classical as well as of quantum information. When the information to be transmitted is classical, we may invoke the family of classical error correction techniques for counteracting the impact of decoherence, as depicted in the system model of Figure 1.3. More specifically, the classical information is first encoded using a classical error correction code. The encoded bits are then *mapped* onto the qubits, which are transmitted over a quantum channel. The mapping of classical bits to qubits may be carried out for example by the so-called superdense coding protocol [28]. By contrast, for a more general communication system, which supports the transmission of classical as well as quantum information, and for reliable quantum computation, we have to resort to the class of Quantum Error Correction Codes (QECCs), which exploit redundancy in the quantum domain in contrast to the classical redundancy of Figure 1.3. Analogous to the decoherence phenomenon, QECCs also rely on the peculiar phenomenon of entanglement - hence John Preskill eloquently pointed out that we are “fighting entanglement with entanglement” [29]. More explicitly, similar to the classical channel coding techniques, QECCs rectify the impact of quantum noise (bit-flips and phase-flips) for the sake of ensuring that the qubits

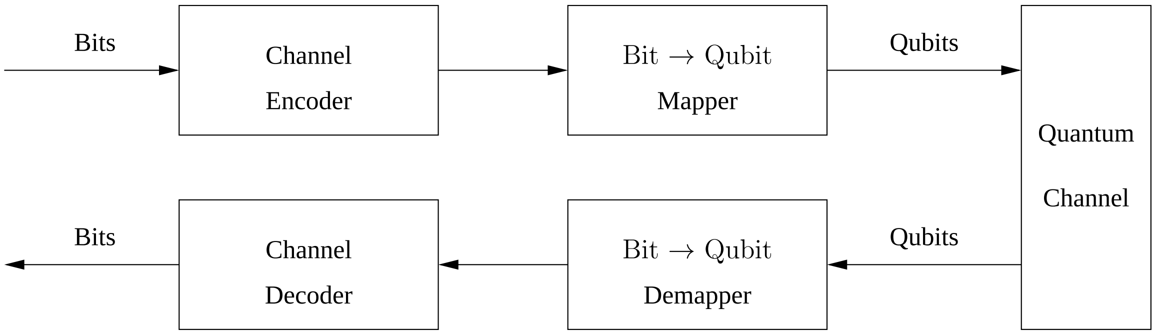


Figure 1.3: Quantum-based communication system for the transmission of classical information, relying on classical error correction codes, where bit-to-qubit mapping may be carried out using the superdense coding protocol [28].

retain their coherent quantum state for longer durations with a high probability, thus in effect beneficially increasing the coherence time of the unperturbed quantum state. This has been experimentally demonstrated in [30, 31, 32].

Against this background, in this thesis, we will focus our attention on the design of classical codes for the system of Figure 1.3 as well as on QECCs, which are indispensable for conceiving a quantum communication system and also for quantum computing.

1.2 Historical Overview of QECCs

A major breakthrough in the field of quantum information processing was marked by Shor's pioneering work on QECCs, which dispelled the notion that conceiving QECCs was infeasible due to the existence of the no-cloning theorem. Inspired by the classical 3-bit repetition codes, Shor conceived the first quantum code in his seminal paper [25], which was published in 1995. The proposed code had a coding rate of 1/9 and was capable of correcting only single qubit errors. This was followed by Calderbank-Shor-Steane (CSS) codes, invented independently by Calderbank and Shor [33] as well as by Steane [34, 35], which facilitated the design of good quantum codes from the known classical binary linear codes. More explicitly, CSS codes may be defined as follows:

An $[n, k_1 - k_2]$ CSS code, which is capable of correcting t bit errors as well as phase errors, can be constructed from classical linear block codes $C_1(n, k_1)$ and $C_2(n, k_2)$, if $C_2 \subset C_1$ and both C_1 as well as the dual of C_2, i.e. $C_2^{\perp}$, can correct t errors. Here, C_1 is used for correcting bit errors, while $C_2^{\perp}$ is used for phase-error correction.

Therefore, with the aid of CSS construction, the overall problem of finding good quantum codes was reduced to finding good dual-containing or self-orthogonal classical codes. Following these principles, the classical $[7, 4, 3]$ Hamming code was used to design a 7-qubit Steane code [35] having a coding rate of 1/7, which is capable of correcting single isolated errors inflicted on the transmitted codewords. Finally, Laflamme *et al.* [36] and Bennett *et al.* [37] independently proposed the optimal single error correcting code in 1996, which required only four redundant qubits.

Following these developments, Gottesman formalized the notion of constructing quantum codes from the classical binary and quaternary codes by establishing the theory of Quantum Stabilizer Codes (QSCs) [38] in his Ph.D thesis [39]. In contrast to the CSS construction, the stabilizer formalism defines a more general class of quantum codes, which imposes a more relaxed constraint than the CSS codes. Explicitly, the resultant quantum code structure can either assume a CSS or a non-CSS (also called unrestricted) structure, but it has to meet the symplectic product criterion. More specifically, stabilizer codes constitute a broad class of quantum codes, which subsumes CSS codes as a subclass and has undoubtedly provided a firm foundation for a wide variety of quantum codes developed, including for example quantum Bose-Chaudhuri-Hocquenghem (BCH) codes [40, 41, 42, 43], quantum Reed-Solomon codes [44, 45], Quantum Low Density Parity Check (QLDPC) codes [46, 47, 48, 49], Quantum Convolutional Codes (QCCs) [50, 51, 52, 53], Quantum Turbo Codes (QTCs) [54, 55] as well as quantum polar codes [56, 57, 58]. These major milestones achieved in the history of QECCs are chronologically arranged in Figure 1.4. Let us now look deeper into the development of QCCs, QLDPC codes and QTCs, which have been the prime focus of most recent research both in the classical as well as in the quantum domains.

1.2.1 Quantum Convolutional Codes

The inception of QCCs dates back to 1998. Inspired by the higher coding efficiencies of Classical Convolutional Codes (CCCs) as compared to the comparable block codes and the low latency associated with the online encoding and decoding of CCCs [59], Chau conceived the first QCC in [60]. He also generalized the classical Viterbi decoding algorithm for the class of quantum codes in [61], but he overlooked some crucial encoding and decoding aspects. Later, Ollivier *et al.* [50, 51] revisited the class of stabilizer-based convolutional codes. Similar to the classical Viterbi decoding philosophy, they also conceived a Look-Up Table (LUT) based quantum Viterbi algorithm for the maximum likelihood decoding of QCCs, whose complexity increases linearly with the number of encoded qubits. Ollivier *et al.* also derived the corresponding online encoding and decoding circuits having complexity which increased linearly with the number of encoded qubits. Unfortunately, their proposed rate-1/5 single-error correcting QCC did not provide any performance or decoding complexity gain over the rate-1/5 single-error correcting block code of [36]. Pursuing this line of research, Almeida *et al.* [62] constructed a rate-1/4 single-error correcting Shor-type concatenated QCC from a CCC$(2, 1, 2)$ and invoked the classical syndrome-based trellis decoding for the quantum domain. Hence, the proposed QCC had a higher coding rate than the QCC of [50, 51]. However, this coding efficiency was achieved at the cost of a relatively high encoding complexity associated with the concatenated trellis structure. It must be pointed out here that the pair of independent trellises used for decoding the bit-flips and phase-flips impose a lower complexity than a large joint trellis would. Finally, Forney *et al.* [52, 53]

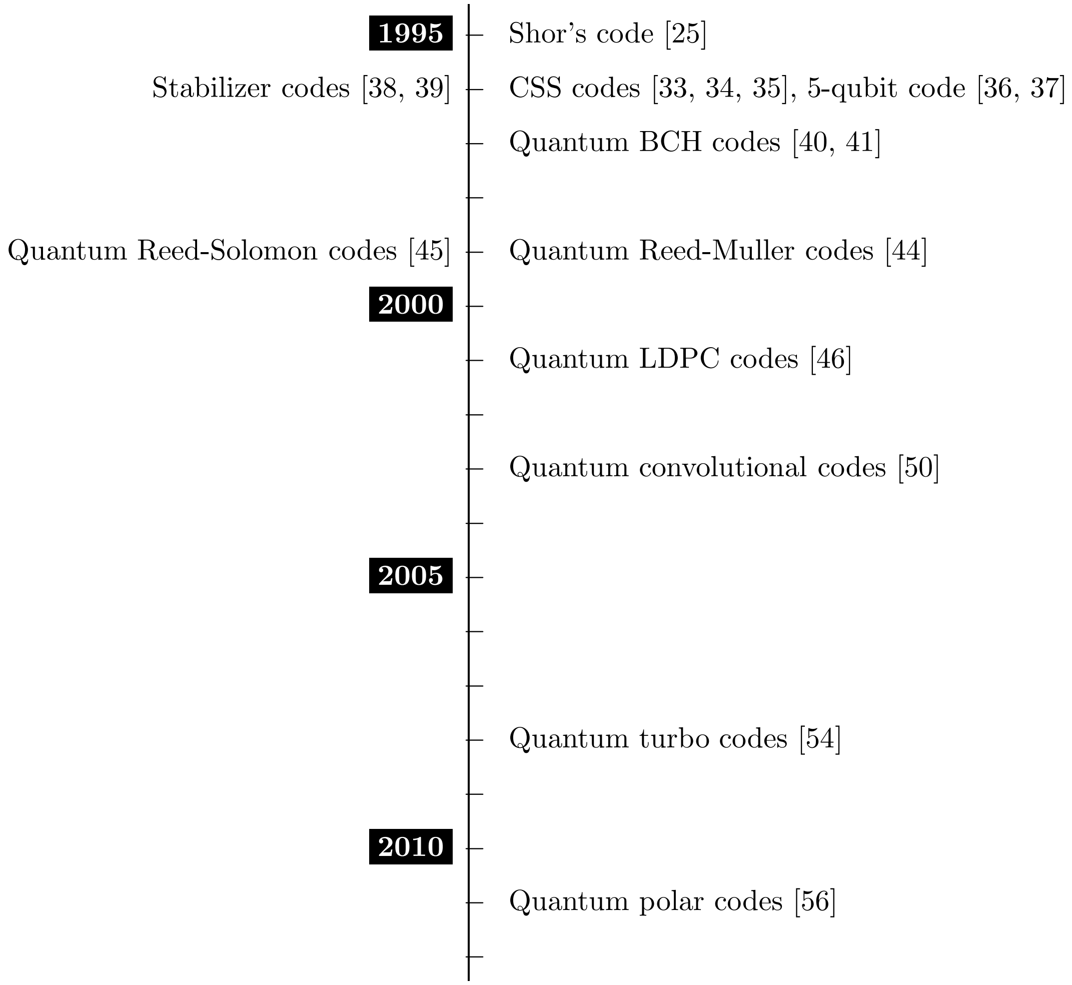

1995
Shor's code [25]
Stabilizer codes [38, 39]
CSS codes [33, 34, 35], 5-qubit code [36, 37]
Quantum BCH codes [40, 41]
Quantum Reed-Solomon codes [45]
Quantum Reed-Muller codes [44]
2000
Quantum LDPC codes [46]
Quantum convolutional codes [50]
2005
Quantum turbo codes [54]
2010
Quantum polar codes [56]

Figure 1.4: Major milestones achieved in the history of quantum error correction codes.

designed rate-$(n-2)/n$ QCCs comparable to their classical counterparts, thus providing higher coding efficiencies than the comparable block codes. Forney *et al.* [52, 53] achieved this by invoking arbitrary classical self-orthogonal rate-$1/n$ $\mathbb{F}_4$-linear and $\mathbb{F}_2$-linear convolutional codes for constructing unrestricted and CSS-type QCCs, respectively. Forney *et al.* [52, 53] also conceived a simple decoding algorithm for single-error correcting codes. Both the coding efficiency and the decoding complexity of the aforementioned QCC structures are compared in Table 1.3. Furthermore, in the spirit of finding new constructions for QCCs, Grassl *et al.* [63, 64] constructed QCCs using the classical self-orthogonal product codes, while Aly *et al.* explored various algebraic constructions in [65] and [66]. Particularly, the QCCs of [65] were derived from classical BCH codes, while the QCCs of [66] were constructed from the classical Reed-Solomon and Reed-Muller codes. Recently, Pelchat and Poulin made a major contribution to the decoding of QCCs by proposing degenerate Viterbi decoding [67], which runs the Maximum *A Posteriori* (MAP) algorithm [68] over the equivalent classes of degenerate errors, thereby improving the attainable performance. The major contributions to the development of QCCs are summarized in Table 1.4.

Author(s)	Coding Efficiency	Decoding Complexity
Ollivier and Tillich [50, 51]	Low	Moderate
Almeida and Palazzo [62]	Moderate	Moderate
Forney *et al.* [52, 53]	High	Low

Table 1.3: Comparison of the Quantum Convolutional Code (QCC) structures.

Year	Author(s)	Contribution
1998	Chau [60]	The first QCCs were developed. Unfortunately, some important encoding/decoding aspects were ignored.
1999	Chau [61]	Classical Viterbi decoding algorithm was generalized to the quantum domain. However, similar to [60], some crucial encoding/decoding aspects were overlooked.
2003	Ollivier and Tillich [50, 51]	Stabilizer-based convolutional codes and their maximum likelihood decoding using the Viterbi algorithm were revisited to overcome the deficiencies of [60, 61]. Failed to provide better performance or decoding complexity than the comparable block codes.
2004	Almeida and Palazzo [62]	Shor-type concatenated QCC was conceived and classical syndrome trellis was invoked for decoding. A high coding efficiency was achieved at the cost of a relatively high encoding complexity.
2005	Forney *et al.* [52, 53]	Unrestricted and CSS-type QCCs were derived from arbitrary classical self-orthogonal $\mathbb{F}_4$ and $\mathbb{F}_2$ CCCs, respectively, yielding a higher coding efficiency as well as a lower decoding complexity than the comparable block codes.
2005	Grassl and Rotteler [63, 64]	Conceived a new construction for QCCs from the classical self-orthogonal product codes.
2007	Aly *et al.* [65]	Algebraic QCCs dervied from BCH codes.
2008	Aly *et al.* [66]	Algebraic QCCs constructed from Reed-Solomon and Reed-Muller Codes.
2013	Pelchat and Poulin [67]	Degenerate Viterbi decoding was conceived, which runs the MAP algorithm over the equivalent classes of degenerate errors, thereby improving the performance.

Table 1.4: Major contributions to the development of Quantum Convolutional Codes (QCCs).

1.2.2 Quantum Low Density Parity Check Codes

Although convolutional codes provide a somewhat better performance than the comparable block codes, yet they are not powerful enough to yield a capacity approaching performance, when used on their own. Consequently, the desire to operate close to the achievable capacity at an affordable decoding complexity further motivated researchers to design beneficial quantum counterparts of the classical LDPC codes [69], which achieve information rates close to the Shannonian capacity limit with the aid of iterative decoding schemes. Furthermore, the sparseness of the LDPC matrix is of particular interest in the quantum domain, because it requires only a small number of interactions per qubit during the error correction procedure, thus facilitating fault-tolerant decoding. Moreover, this sparse nature also makes QLDPC codes highly degenerate.

Postol [46] conceived the first example of a non-dual-containing CSS-based QLDPC code from a finite geometry based classical LDPC in 2001. Later, Mackay *et al.* [47] proposed various code structures (e.g. bicycle codes and unicycle codes) for constructing QLDPC codes from the family of classical dual-containing LDPC codes. Additionally, Mackay *et al.* also proposed the class of Cayley graph-based dual-containing codes in [70], which were further investigated by Couvreur *et al.* in [71, 72]. Aly *et al.* contributed to these developments by constructing dual-containing QLDPC codes from finite geometries in [73], while Djordjevic exploited the Balanced Incomplete Block Designs (BIBDs) in [74], albeit neither of these provided any gain over Mackay's bicycle codes. Lou *et al.* [75, 76] invoked the non-dual-containing CSS structure by using both the generator and the PCM of classical Low Density Generator Matrix (LDGM) based codes. Hagiwara *et al.* [77] conceived Quasi-Cyclic (QC) QLDPC codes, whereby the constituent PCMs of non-dual-containing CSS-type QLDPCs were constructed from a pair of QC-LDPC codes found using algebraic combinatorics. Hagiwara's design of [77] was extended to non-binary QLDPC codes in [78, 79], which operate closer to the Hashing limit than MacKay's bicycle codes. The concept of QC-QLDPC codes was further extended to the class of spatially-coupled QC codes in [80]. While all the aforementioned QLDPC constructions were CSS-based, Camara *et al.* [49] were the first authors to conceive non-CSS QLDPC codes. Later, Tan *et al.* [81] proposed several systematic constructions for non-CSS QLDPC codes, four of which were based on classical binary QC-LDPC codes, while one was derived from classical binary LDPC-convolutional codes. Since most of the above-listed QLDPC constructions exhibit an upper bounded minimum distance, topological QLDPCs[2] were derived from Kitaev's construction in [82, 83, 84]. Amidst these activities, which focused on the construction of QLDPC codes, Poulin *et al.* were the first scientists to address the decoding issues of QLDPC codes [85], which were further improved in [86]. The major contributions made in the context of QLDPC codes are summarized in Table 1.6, while the most promising QLDPC construction methods are compared in Table 1.5[3].

[2] Topological code structures are beyond the scope of this thesis.

[3] All QLDPC codes must have short cycles in the quaternary formalism, which will be discussed in Chapter 7. The second column only indicates 'short cycles' in the binary formalism.

Code Construction	**Short Cycles**	**Minimum Distance**	**Delay**	**Decoding Complexity**
Bicycle codes [47]	Yes	Upper Bounded	Standard	Standard
Cayley-graph based codes [70, 71, 72]	Yes	Increases with the code length	Standard	Increases with the code length
LDGM-based codes [75, 76]	Yes	Upper Bounded	Standard	High
Non-binary quasi-cyclic codes [78, 79]	No	Upper Bounded	Standard	High
Spatially-coupled quasi-cyclic codes [80]	No	Upper Bounded	High	High

Table 1.5: Comparison of the Quantum Low Density Parity Check (QLDPC) code structures.

1.2.3 Quantum Turbo Codes

Pursuing further the direction of iterative code structures, Poulin *et al.* conceived QTCs in [54, 55], based on the interleaved serial concatenation of QCCs. Unlike QLDPC codes, QTCs offer a complete freedom in choosing the code parameters, such as the frame length, coding rate, constraint length and interleaver type. Moreover, their decoding is not impaired by the presence of length-4 cycles associated with the symplectic criterion. Furthermore, in contrast to QLDPC codes, the iterative decoding invoked for QTCs takes into account the inherent degeneracy associated with quantum codes. However, it was found in [54, 55, 87] that the constituent QCCs cannot be simultaneously both recursive and noncatastrophic. Since the recursive nature of the inner code is essential for ensuring an unbounded minimum distance, whereas the noncatastrophic nature is a necessary condition to be satisfied for achieving decoding convergence to a vanishingly low error rate, the QTCs designed in [54, 55] had a bounded minimum distance. The QBER performance curves of the QTCs conceived in [54, 55] also failed to match the classical turbo codes. This issue was dealt with in [88], where the quantum turbo decoding algorithm of [55] was improved by iteratively exchanging the *extrinsic* rather than the *a posteriori* information. The major contributions made in the domain of QTCs are summarized in Table 1.6.

		Year	**Author(s)**	**Code Type**	**Contribution**
QLDPC	Code Construction	2001	Postol [46]	Non-dual	The first example of QLDPC code constructed from a finite geometry based classical code. A generalized formalism for constructing QLDPC codes from the corresponding classical codes was not developed.
		2004	Mackay *et al.* [47]	Dual	Various code structures, e.g. bicycle codes and unicycle codes, were conceived for constructing QLDPC codes from classical dual-containing LDPC codes. Performance impairment due to the presence of unavoidable length-4 cycles was first pointed out in this work. Minimum distance of the resulting codes was upper bounded by the row weight.
		2005	Lou *et al.* [75, 76]	Non-dual	The generator and PCM of classical LDGM codes were exploited for constructing CSS codes. An increased decoding complexity was imposed and the codes had an upper bounded minimum distance.
		2007	Mackay [70]	Dual	Cayley graph-based QLDPC codes were proposed, which had numerous length-4 cycles.
			Camara *et al.* [49]	Non-CSS	QLDPC codes derived from classical self-orthogonal quaternary LDPC codes were conceived, which failed to outperform MacKay's bicycle codes.
			Hagiwara *et al.* [77]	Non-dual	Quasi-cyclic QLDPC codes were constructed using a pair of quasi-cyclic LDPC codes, which were found using algebraic combinatorics. The resultant codes had at least a girth of 6, but they failed to outperform MacKay's constructions given in [47].
		2008	Aly *et al.* [73]	Dual	QLDPC codes were constructed from finite geometries, which failed to outperform Mackay's bicycle codes.
			Djordjevic [74]	Dual	BIBDs were exploited to design QLDPC codes, which failed to outperform Mackay's bicycle codes.

Table 1.6: (Continued on the next page)

		Year	Author(s)	Code Type	Contribution
		2010	Tan *et al.* [81]	Non-CSS	Several systematic constructions for non-CSS QLDPC codes were proposed, four of which were based on classical binary quasi-cyclic LDPC codes, while one was derived from classical binary LDPC-convolutional codes. These code designs failed to outperform Mackay's bicycle codes.
		2011	Couvreur *et al.* [71, 72]	Dual	Cayley graph-based QLDPC codes of [70] were further investigated. The lower bound on the minimum distance of the resulting QLDPC was logarithmic in the code length, but this was achieved at the cost of an increased decoding complexity.
			Kasai [78, 79]	Non-dual	Quasi-cyclic QLDPC codes of [77] were extended to non-binary constructions, which outperformed Mackay's bicycle codes at the cost of an increased decoding complexity. Performance was still not at par with the classical LDPC codes and minimum distance was upper bounded.
			Hagiwara *et al.* [80]	Non-dual	Spatially-coupled QC-QLDPC codes were developed, which outperformed the 'non-coupled' design of [77] at the cost of a small coding rate loss. Performance was similar to that of [78, 79], but larger block lengths were required.
	Decoding	2008	Poulin *et al.* [85]		Heuristic methods were developed to alleviate the performance degradation caused by unavoidable length-4 cycles and symmetric degeneracy error.
		2012	Wang *et al.* [86]		Feedback mechanism was introduced in the context of the heuristic methods of [85] to further improve the performance.
QTC	Code Construction	2008	Poulin *et al.* [54, 55]	Non-CSS	QTCs were conceived based on the interleaved serial concatenation of QCCs. QTCs are free from the decoding issue associated with the length-4 cycles and they offer a wider range of code parameters. Degenerate iterative decoding algorithm was also proposed. Unfortunately, QTCs have an upper bounded minimum distance.

Table 1.6: (Continued on the next page)

		Year	Author(s)	Code Type	Contribution
	Decoding	2014	Wilde *et al.* [88]		The iterative decoding algorithm of [54, 55] failed to yield performance similar to the classical turbo codes. The decoding algorithm was improved by iteratively exchanging the *extrinsic* rather than the *a posteriori* information.

Table 1.6: Major contributions to the development of iterative quantum codes, where the code types 'dual-containing CSS' and 'non-dual-containing CSS' are abbreviated as 'dual' and 'non-dual', respectively.

1.2.4 Entanglement-Assisted Quantum Codes

Some of the well-known classical codes cannot be imported into the quantum domain by invoking the aforementioned stabilizer-based code constructions because the stabilizer codes have to satisfy the stringent symplectic product criterion. This limitation was overcome in [89, 90, 91, 92] with the notion of EA quantum codes, which exploit pre-shared entanglement between the transmitter and receiver. Later, this concept was extended to numerous other code structures, e.g. EA-QLDPC code [93], EA-QCC [94], EA-QTC [95, 88] and EA-polar codes [96]. In [95, 88], it was also found that EA-QCCs may be simultaneously both recursive as well as non-catastrophic. Therefore, the issue of bounded minimum distance of QTCs was resolved with the notion of entanglement. Furthermore, EA-QLDPC codes are free from length-4 cycles in the binary formalism, which in turn results in an impressive performance similar to that of the corresponding classical LDPC codes. Hence, the concept of the entanglement-assisted regime resulted in a major breakthrough in terms of constructing quantum codes, whose behaviour is similar to that of the corresponding classical codes. The major milestones achieved in the history of entanglement-assisted quantum error correction codes are chronologically arranged in Figure 1.5.

1.3 Outline of the Thesis

We next describe the structure of the thesis, which is also summarized in Figure 1.6.

- **Chapter 2: Preliminaries of Quantum Information**

  In Chapter 2, we provide the portrayal of introduction to quantum information theory. We commence our discussions with qubits in Section 2.2, which is extended to an N-qubit quantum system in Section 2.3. In Sections 2.4 and 2.5, we lay out the fundamental concepts of the no-cloning theorem and of entanglement, respectively. We then proceed with a discussion on

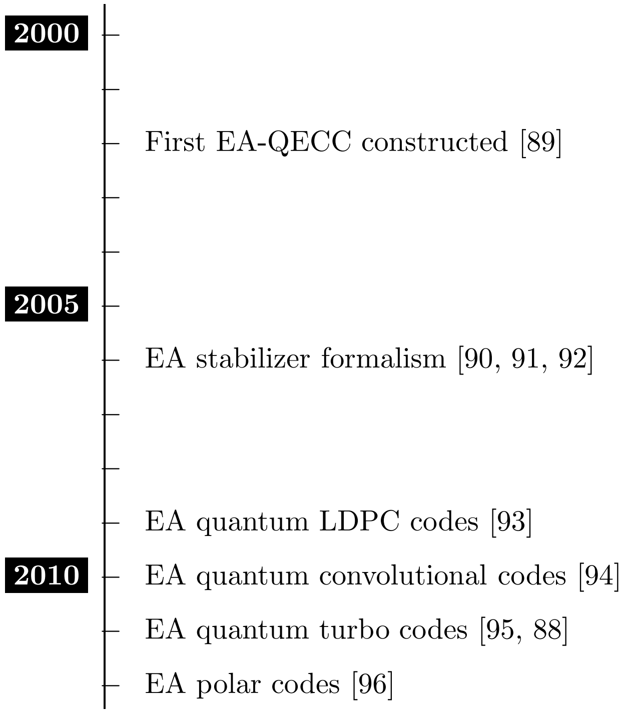

2000
First EA-QECC constructed [89]
2005
EA stabilizer formalism [90, 91, 92]
EA quantum LDPC codes [93]
2010
EA quantum convolutional codes [94]
EA quantum turbo codes [95, 88]
EA polar codes [96]

Figure 1.5: Major milestones achieved in the history of entanglement-assisted quantum error correction codes.

the quantum-domain unitary operators in Section 2.6, while the Pauli group is introduced in Section 2.7. Finally, we highlight the various quantum channel models in Section 2.8.

- **Chapter 3: Near-Capacity Code Designs for Entanglement-Assisted Classical Communication**

  In Chapter 3, we invoke EXtrinsic Information Transfer (EXIT) chart aided near-capacity classical code designs conceived for reliable transmission of classical bits over the quantum communication channel of Figure 1.3. More specifically, we focus our attention on the entanglement-assisted transmission of classical information over quantum channels[4], which is achieved with the aid of the SuperDense (SD) coding protocol. We commence by reviewing the SD protocol in Section 3.2, which is in essence the 'Bit → Qubit Mapper' of Figure 1.3. We next characterize the associated capacity in Section 3.3. In Section 3.4, we conceive a bit-based scheme, which exploits classical channel coding by serially concatenating a classical Irregular Convolutional Code (IRCC) and a classical Unity Rate Code (URC) with a quantum-based SD encoder, hence refer to it as an IRCC-URC-SD system. We present our EXIT-chart aided near-capacity design criterion in Section 3.5, where the IRCC is optimized for achieving a near-capacity performance. Our

[4] A quantum channel can be used for modeling imperfections in quantum hardware, namely, faults resulting from quantum decoherence and quantum gates. Furthermore, a quantum channel can also model quantum-state flips imposed by the transmission medium, including free-space wireless channels and optical fiber links, when qubits are transmitted across these media.

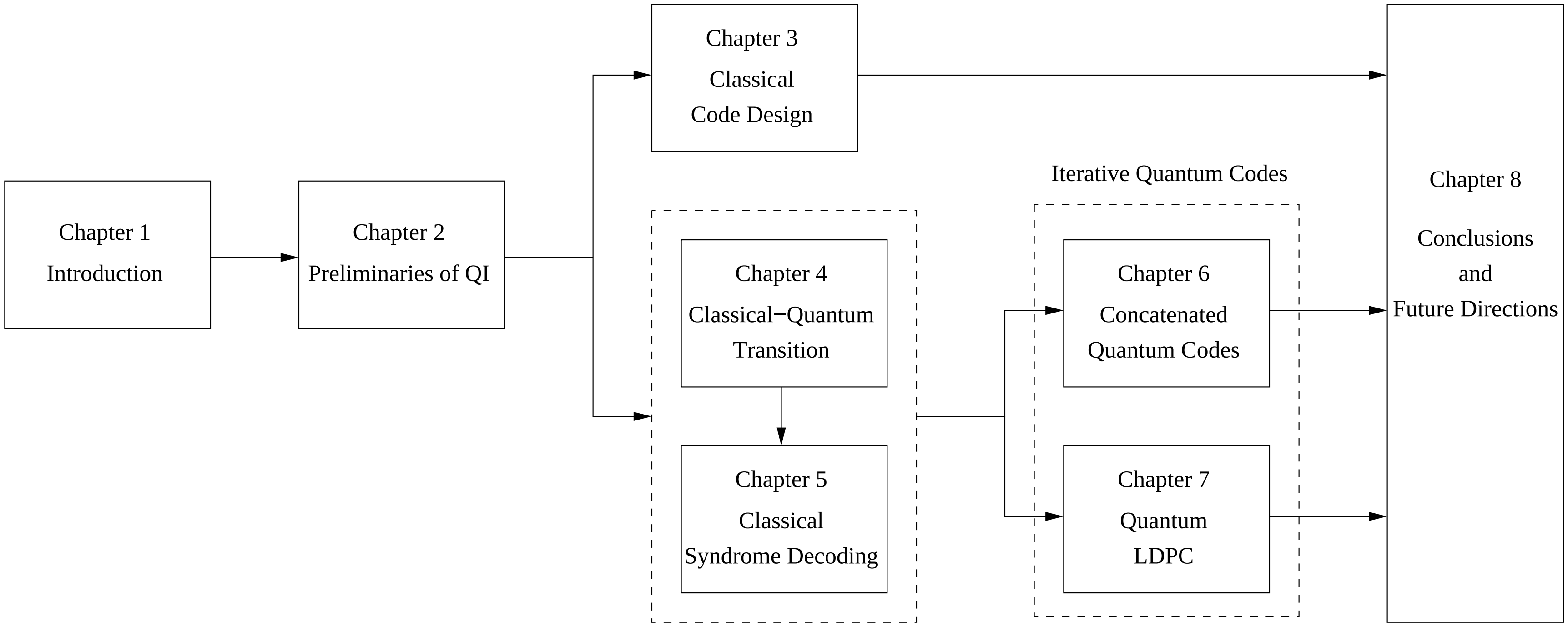


Figure 1.6: Structure of the thesis.

bit-based code structure of Section 3.4 incurs a capacity loss due to the symbol-to-bit conversion. To overcome this capacity loss, we propose a symbol-based code design in Section 3.7, which employs a single-component Convolutional Code (CC) and a symbol interleaver in contrast to the IRCC and bit interleaver of Section 3.7.

- **Chapter 4: From Classical to Quantum Error Correction**

  Since the code designs of Chapter 3 rely on classical-domain redundancy, they are only suitable for the reliable transmission of classical information over a quantum channel. For more general quantum communication systems, which may transmit both classical as well as quantum information, and for quantum computation systems, it is vital to invoke QECCs, which exploit the redundancy in the quantum domain. In this spirit, in Chapters 6 and 7, we design QECCs, for which the foundation is developed in Chapters 4 and 5. More specifically, in Chapter 4, we detail the quantum to classical isomorphism, which facilitates the construction of quantum codes from the known classical codes. In Section 4.2, we review the classical linear block codes. We next discuss the QSCs in Section 4.3, which are derived from the classical linear block codes of Section 4.2. In particular, we lay out the underlying quantum to classical isomorphism, which forms the basis for importing arbitrary classical codes into the quantum domain. We extend our discussions to the construction of QCCs from the CCCs in Section 4.4, while Section 4.5 presents EA-QSCs, which facilitate the design of quantum codes from arbitrary classical codes without imposing any stringent requirements.

- **Chapter 5: Classical Syndrome Decoding**

  In Chapter 4, the stabilizer codes are characterized by an equivalent classical PCM. Therefore, they are decoded using the classical PCM-based syndrome decoding. In this spirit, we discuss the classical syndrome decoding techniques operating over a classical channel in Chapter 5. We commence our discussion with the conceptually simplest LUT-based syndrome decoding in Section 5.2, while Section 5.3 details the construction of the syndrome-based error trellis for linear block codes and convolutional codes. Finally, in Section 5.4, we details the Block Syndrome Decoding (BSD) approach designed for reducing the decoding complexity. In particular, we conceive a syndrome-based block decoding approach for the classical Turbo Trellis Coded Modulation (TTCM) scheme.

- **Chapter 6: EXIT-Chart Aided Hashing Bound Approaching Concatenated Quantum Codes**

  In Chapter 4, we presented the methodology of constructing QECCs from the known classical codes, followed by the associated classical syndrome decoding approach in Chapter 5. Pursuing further the design of QECCs, in Chapter 6, we construct Hashing bound approaching QECCs, based on the foundation laid down in Chapters 4 and 5. The related discourse begins by laying out the design objectives in Section 6.2. Section 6.3 then details the circuit based representation of QCCs, which facilitates the degenerate iterative decoding of concatenated quantum codes. We next present our system model and the associated degenerate iterative decoding in Section 6.4. Finally, in Section 6.5, we extend the application of classical nonbinary EXIT charts to the circuit-based syndrome decoder of QTCs for approaching the Hashing bound[5]. For the sake of further facilitating the Hashing bound approaching code design, we propose the general structure of Quantum IRregular Convolutional Code (QIRCC) in Section 6.7, which constitutes the outer component of a concatenated quantum code.

- **Chapter 7: Quantum Low Density Parity Check Codes**

  Pursuing further the design of iterative code structures, we focus our efforts on QLDPC codes in Chapter 7, which may be constructed from the classical binary as well as quaternary codes. In this context, Section 7.2 reviews the various QLDPC construction methods, while the QLDPC decoding methods and the associated challenges are discussed in Section 7.3. In Section 7.4, we propose a formalism for constructing high-rate row-circulant QC-QLDPC codes from arbitrary row-circulant classical LDPC matrices. In Section 7.6, we conceive a modified non-binary decoding algorithm for homogeneous CSS-type QLDPC codes, for the sake of alleviating the problems imposed by unavoidable length-4 cycles. Finally, Section 7.7 details the reweighted BP algorithm, which is known to alleviate the structural flaw of short cycles in classical LDPC codes.

- **Chapter 8: Conclusions and Future Directions**

  Chapter 8 summarizes the main results of this thesis and outlines a range of promising future research directions.

1.4 Novel Contributions of the Thesis

The novel contributions of this thesis are summarized below:

- A near-capacity code design is conceived for entanglement-assisted classical communication over the quantum depolarizing channel. The proposed system relies on efficient EXIT-chart aided near-capacity classical code designs conceived for approaching the entanglement-assisted classical capacity of a quantum depolarizing channel. It incorporates an IRCC, a URC and a soft-decision aided SD, which is hence referred to as an IRCC-URC-SD arrangement [5].
- Our previously proposed IRCC-URC-SD design is bit-based, thereby incurring a capacity loss due to symbol-to-bit conversion. To circumvent this capacity loss, an alternative iterative code design is proposed, which is referred to as a symbol-based CC-URC-SD. This symbol-based concatenated code design incorporates a single CC as the outer component, while the URC and SD schemes constitute the amalgamated symbol-based inner code. The resultant system is optimized for approaching the capacity by invoking non-binary EXIT charts [7].

[5] The Hashing bound sets the lower limit on the achievable capacity.

- The iterative decoder of of a classical TTCM exchanges extrinsic information between the constituent TCM decoders, which imposes a high computational complexity at the receiver. Therefore we conceive the syndrome-based block decoding of TTCM, which is capable of reducing the decoding complexity by disabling the decoder, when the syndrome becomes zero [6].

- We adapt the conventional nonbinary EXIT charts for concatenated quantum codes by exploiting the intrinsic quantum-to-classical isomorphism. The EXIT chart analysis not only allows us to dispense with the time-consuming Monte Carlo simulations but also facilitates the design of near-capacity codes without resorting to the analysis of their distance spectra. We further analyze the behaviour of both an unassisted (non-recursive) and of an entanglement-assisted (recursive) inner convolutional code using EXIT charts for demonstrating the benefits of recursive code structures in terms of achieving an unbounded minimum distance. Finally, we optimize the constituent components of the concatenated structure using the EXIT charts conceived for approaching the Hashing bound [4].

- A generically applicable structure is conceived for QIRCC, which can be dynamically adapted to match any given inner code, for achieving a Hashing-bound-approaching performance. More specifically, we construct a 10-subcode QIRCC for demonstrating a Hashing-bound-approaching performance [3].

- Since the high-rate classical QC-LDPC codes are known to operate efficiently both at short and at moderate lengths, a new class of high-rate row-circulant QC-QLDPC codes is proposed. The conceived family of QC-QLDPC codes can be constructed from arbitrary row-circulant classical QC-LDPC matrices using the simple transpose and column permutation operations [2].

- We have conceived a modified non-binary decoding algorithm for homogeneous CSS-type QLDPC codes for alleviating the structural flaws of unavoidable length-4 cycles, which are inherently associated with these QLDPC matrices [1].

- We amalgamate the Uniformly-Reweighted Belief Propagation (URW-BP) method with QLDPC decoding for further alleviating the issue of short cycles [1].

Chapter 2

Preliminaries of Quantum Information

2.1 Introduction

This chapter provides a preliminary introduction to quantum information theory. It aims to develop a basic understanding of the laws of quantum mechanics and the associated important terminologies used in quantum information.

This chapter is organized as follows. We commence with a discussion on quantum bits in Section 2.2, which is then extended to an N-qubit quantum system in Section 2.3. We next detail the widely known no-cloning theorem and the concept of entanglement in Sections 2.4 and 2.5, respectively. This is followed by a discussion on quantum unitary transformations in Section 2.6. Finally, we define the Pauli group in Section 2.7, while the different quantum channel models are detailed in Section 2.8. In Section 2.9, we conclude this chapter.

2.2 Quantum Bits

The elementary unit of information in classical computers is a binary digit (or bit), which can either assume a value of 0 or 1. The analogous term in quantum information theory is a qubit (quantum bit), which has the unique characteristic that it can also exist in a linear combination of the states 0 and 1 (often called superposition). The resulting superimposed state of a qubit is represented as [18]:

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle, \tag{2.1}$$

where $|\ \rangle$ is called Dirac notation or Ket [97], which is a standard notation for states in quantum physics, while α and β are complex numbers with $\alpha^2 + \beta^2 = 1$. More specifically, a qubit is a vector in the two-dimensional Hilbert space $\mathbb{C}^2$ with $|0\rangle$ and $|1\rangle$ (known as computational basis states) constituting the orthogonal basis for the vector space, which have the amplitudes α and β, respectively.

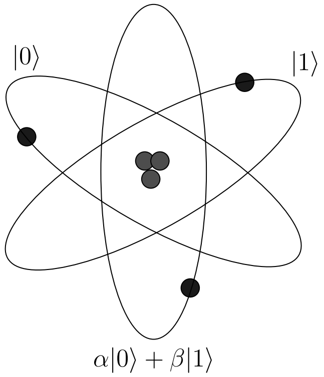


Figure 2.1: A qubit realized by an orbiting electron - ground state denotes the state $|0\rangle$, while the excited state represents the orthogonal basis state $|1\rangle$. A qubit may also be found in a superposition of the two basis states, which is denotes as $\alpha|0\rangle + \beta|1\rangle$.

Consequently, the qubit shown in Eq. (2.1) can also be represented in vector notation as follows [18]:

$$|\psi\rangle = \begin{pmatrix}\alpha \\ \beta\end{pmatrix} = \alpha \begin{pmatrix}1 \\ 0\end{pmatrix} + \beta \begin{pmatrix}0 \\ 1\end{pmatrix} \tag{2.2}$$

where the pure states $|0\rangle$ and $|1\rangle$ are given by:

$$|0\rangle = \begin{pmatrix}1 \\ 0\end{pmatrix} \quad , \qquad |1\rangle = \begin{pmatrix}0 \\ 1\end{pmatrix} . \tag{2.3}$$

However, it is not possible to observe the superimposed state of a qubit. More specifically, a qubit exists in a continuum of states between $|0\rangle$ and $|1\rangle$ until it is ‘measured’ or ‘observed’. Upon ‘measurement’ it collapses to the state $|0\rangle$ with a probability of $|\alpha|^2$ and $|1\rangle$ with a probability of $|\beta|^2$.

The basis states of a 2-dimensional quantum space may be realized in different ways, e.g. two different photon polarizations, different alignments of a nuclear spin, the charge/current/energy of a Josephson junction or two different energy levels of an orbiting electron [18, 98]. The latter has been illustrated in Figure 2.1, where an electron can exist either in the ‘ground’ or the ‘excited’ state, which corresponds to the basis states $|0\rangle$ and $|1\rangle$, respectively. The electron can be moved from the ground state to the excited state by exposing the atom to an appropriate amount of light (or energy) for an appropriate period of time. If the exposure time is reduced, the electron may be moved to an arbitrary superposition of the two basis energy levels, i.e. $\alpha|0\rangle + \beta|1\rangle$, as illustrated in Figure 2.1. However, the weird laws of quantum mechanics do not allow us to see or observe this superimposed state. Upon measurement, the electron will be found in the ground state with a probability of $|\alpha|^2$ and in the excited state with a probability of $|\beta|^2$ [18].

The two-dimensional complex vector space of a qubit can also be visualized in 3D as a unique point on the surface of a unit-radius sphere, which is known as a Bloch sphere [18]. More explicitly, since $\alpha^2 + \beta^2 = 1$, Eq. (2.1) can also be written as [18]:

$$|\psi\rangle = e^{i\gamma}(\cos\frac{\theta}{2}|0\rangle + e^{i\varphi}\sin\frac{\theta}{2}|1\rangle), \tag{2.4}$$

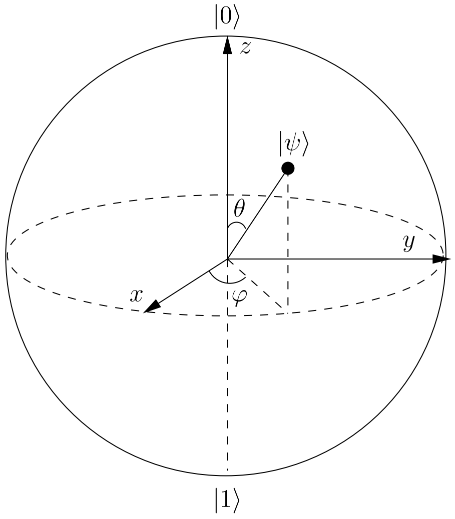


Figure 2.2: 3D representation of a qubit ψ on Bloch sphere, which is parametrized by the variables θ and φ. The computational basis states $|0\rangle$ and $|1\rangle$ correspond to the North and South poles, respectively, while an arbitrary quantum state may lie anywhere on the surface of the sphere.

where θ,γ and φ are real numbers. In Eq. (2.4), the arbitrary phase γ has no observable effect, i.e. $|\psi\rangle$ and $e^{i\gamma}|\psi\rangle$ yield the same output upon measurement. Consequently, the arbitrary phase γ may be ignored, which reduces Eq. (2.4) to:

$$|\psi\rangle = \cos\frac{\theta}{2}|0\rangle + e^{i\varphi}\sin\frac{\theta}{2}|1\rangle. \tag{2.5}$$

Consequently, a qubit is defined as a point on the Bloch sphere using the two variables θ and φ, as shown in Figure 2.2. The computational basis states $|0\rangle$ and $|1\rangle$ correspond to the instances where θ is 0 and π, respectively, in Figure 2.2. A qubit having an arbitrary state, which is characterized by Eq. (2.5), may lie anywhere on the surface of the sphere.

2.3 N-Qubit Composite System

A single qubit is essentially a vector in the 2-dimensional Hilbert space. Consequently, an N-qubit composite system, which consists of N qubits, has a 2^N-dimensional Hilbert space, which is the tensor product of the Hilbert space of the individual qubits. For example, a 2-qubit composite system, having the constituent qubits $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$ and $|\psi'\rangle = \alpha'|0\rangle + \beta'|1\rangle$, may be formulated as follows:

$$\begin{aligned}|\psi\rangle \otimes |\psi'\rangle &= (\alpha|0\rangle + \beta|1\rangle) \otimes (\alpha'|0\rangle + \beta'|1\rangle)\\ &= \alpha\alpha'|00\rangle + \alpha\beta'|01\rangle + \beta\alpha'|10\rangle + \beta\beta'|11\rangle,\end{aligned} \tag{2.6}$$

where $\otimes$ denotes the tensor product and the state $|ij\rangle$ is the tensor product of 1st and 2nd qubits having states $|i\rangle$ and $|j\rangle$, respectively, i.e. $|i\rangle \otimes |j\rangle$. Consequently, the state of an N-qubit system may be represented by a unit-length vector, which is the tensor product of N two-dimensional Hilbert spaces, i.e. $(\mathbb{C}^2)^{\otimes N}$, whose basis vectors are given by all the tensor products of the form $|x_1\rangle \otimes \cdots \otimes |x_N\rangle$, where $x_i \in \{0, 1\}$. The resulting N-qubit superimposed state may be generalized as:

$$\alpha_0|00\ldots 0\rangle + \alpha_1|00\ldots 1\rangle + \cdots + \alpha_{2^N-1}|11\ldots 1\rangle, \tag{2.7}$$

where $\alpha_i \in \mathbb{C}$ and $\sum\limits_{i=0}^{2^N-1} |\alpha_i|^2 = 1$.

Eq. (2.7) gives quantum systems the inherent property of parallelism. For examples, a 2-bit classical computer can store either of the 4 possible bit patterns, i.e. 00, 01, 10, 11. By contrast, a 2-qubit quantum computer can exist in a superposition of all the 4 states simultaneously, as depicted in Eq. (2.6). This in turn facilitates a quantum computer to process all the states concurrently, making it 4 times more powerful, as illustrated in Figure 1.1. Similarly, an N-qubit quantum computer can make 2^N computations at the same time. Hence, the parallelism increases exponentially with a linear increase in the size of the system. Unfortunately, we cannot observe or measure all the 2^N states because the N-qubit quantum state of Eq. (2.7) collapses to one of the basis states upon measurement. More explicitly, the quantum register collapses to the ith basis state with a probability of $|\alpha_i|^2$ under the idealized assumption of having perfect measurement (or gates). However, quantum parallelism may be manipulated to determine a certain desired property of a function, hence substantially reducing the computational overhead for certain complex problems[18].

2.4 No-Cloning Theorem

The concept of cloning (or copying) forms the basis of classical error correction codes. However, quantum information does not allow the cloning of qubits [99]. This is a direct consequence of the linearity of transformations.

Let us assume that U is a copying operation, which copies the arbitrary states $|\psi\rangle$ and $|\phi\rangle$ as follows:

$$U|\psi\rangle = |\psi\rangle \otimes |\psi\rangle, \qquad U|\phi\rangle = |\phi\rangle \otimes |\phi\rangle. \tag{2.8}$$

Furthermore, let $|\psi'\rangle$ be a linear combination of $|\psi\rangle$ and $|\phi\rangle$ so that we have:

$$|\psi'\rangle = \lambda_1|\psi\rangle + \lambda_2|\phi\rangle, \tag{2.9}$$

which is copied by the operator U is as follows:

$$\begin{aligned} U|\psi'\rangle &= |\psi'\rangle \otimes |\psi'\rangle \\ &= (\lambda_1|\psi\rangle + \lambda_2|\phi\rangle) \otimes (\lambda_1|\psi\rangle + \lambda_2|\phi\rangle) \\ &= \lambda_1^2|\psi\rangle \otimes |\psi\rangle + \lambda_1\lambda_2|\psi\rangle \otimes |\phi\rangle + \lambda_1\lambda_2|\phi\rangle \otimes |\psi\rangle + \lambda_2^2|\phi\rangle \otimes |\phi\rangle. \end{aligned} \tag{2.10}$$

Since the copying operation U must be linear, we also have:

$$U|\psi'\rangle = U\left(\lambda_1|\psi\rangle + \lambda_2|\phi\rangle\right) = \lambda_1 U|\psi\rangle + \lambda_2 U|\phi\rangle. \tag{2.11}$$

Substituting Eq. (2.8) into Eq. (2.11) gives:

$$U|\psi'\rangle = \lambda_1|\psi\rangle \otimes |\psi\rangle + \lambda_2|\phi\rangle \otimes |\phi\rangle. \tag{2.12}$$

We may notice here that Eq. (2.10) and Eq. (2.12) are not equal. The equality holds only when either λ_1 or λ_2 is zero, i.e. we have $|\psi'\rangle = |\psi\rangle$ or $|\psi'\rangle = |\phi\rangle$. Hence, it is not possible to clone an arbitrary quantum state.

2.5 Entanglement

The 2-qubit composite system depicted in Eq. (2.6) is a tensor product of the constituent qubits $|\psi\rangle$ and $|\psi'\rangle$. By contrast, two qubits are said to be 'entangled' if they cannot be decomposed into the tensor product of the constituent qubits. Let us consider the state:

$$|\psi\rangle = \alpha|00\rangle + \beta|11\rangle, \tag{2.13}$$

where both α and β are non-zero. It is not possible to decompose it into two individual qubits because we have:

$$\begin{aligned} \alpha|00\rangle + \beta|11\rangle &\neq (\alpha_1|0\rangle + \beta_1|1\rangle)(\alpha_2|0\rangle + \beta_2|1\rangle) \\ &= \alpha_1\alpha_2|00\rangle + \alpha_1\beta_2|01\rangle + \beta_1\alpha_2|10\rangle + \beta_1\beta_2|11\rangle, \end{aligned} \tag{2.14}$$

for any choice of non-zero α_i and β_i subject to normalization. Consequently, a peculiar link exists between the two qubits such that measuring one qubit also collapses the other, despite their spatial separation. More specifically, if we measure the first qubit of $|\psi\rangle$ seen in Eq. (2.13), we may obtain a $|0\rangle$ with a probability of $|\alpha|^2$ and a $|1\rangle$ with a probability of $|\beta|^2$. If the first qubit is found to be $|0\rangle$, then the measurement of the second qubit will definitely be $|0\rangle$. Similarly, if the first qubit is $|1\rangle$, then the second qubit will also collapse to $|1\rangle$. This mysterious correlation between the two qubits, which doesn't exist in the classical world, is called entanglement. It was termed 'spooky action at a distance' by Einstein [100].

Entanglement finds many applications in quantum information theory. Generally, the 2-qubit based entanglement protocols rely on the following four orthonormal states [18]:

$$\begin{aligned} &\frac{1}{\sqrt{2}}(|00\rangle + |11\rangle) \ , \quad \frac{1}{\sqrt{2}}(|00\rangle - |11\rangle) \ , \\ &\frac{1}{\sqrt{2}}(|01\rangle + |10\rangle) \ , \quad \frac{1}{\sqrt{2}}(|01\rangle - |10\rangle) \ , \end{aligned} \tag{2.15}$$

which are known as Bell states, named after John S. Bell, also referred to as the Einstein-Podolsky-Rosen (EPR) pairs. Two of the widely known applications of entanglement are superdense coding [28] and teleportation [101].

2.6 Unitary Operators

A linear operator, whose inverse is its adjoint (hermitian conjugate), is known as a unitary operator (a quantum gate) [18]. More explicitly, a unitary transformation U acts linearly on the superimposed quantum state, i.e. we have:

$$U(\alpha|0\rangle + \beta|1\rangle) = \alpha U(|0\rangle) + \beta U(|1\rangle), \tag{2.16}$$

and,

$$UU^{\dagger} = \mathbf{I}, \tag{2.17}$$

where $U^{\dagger}$ is the adjoint of U, while $\mathbf{I}$ denotes the identity matrix. More explicitly, a unitary operator preserves the inner product, hence ensuring that the sum of probabilities of all possible states is equal to 1.

Some of the basic unitary operators are discussed below:.

- **Pauli Matrices (I, X, Y, Z-gates):** Pauli matrices (also called Pauli operators) are single-qubit quantum gates defined as [102]:

$$\mathbf{I} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \quad \mathbf{X} = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \quad \mathbf{Y} = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}, \quad \mathbf{Z} = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}. \tag{2.18}$$

The identity matrix $\mathbf{I}$ of Eq. (2.18) is a simple repeat gate. For an input state $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$, the output of an $\mathbf{I}$-gate can be computed as:

$$\begin{aligned} |\psi'\rangle = \mathbf{I}|\psi\rangle &= \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} \alpha \\ \beta \end{pmatrix} \\ &= \begin{pmatrix} \alpha \\ \beta \end{pmatrix} \equiv \alpha|0\rangle + \beta|1\rangle \equiv |\psi\rangle. \end{aligned} \tag{2.19}$$

The Pauli-$\mathbf{X}$ of Eq. (2.18) is analogous to the classical *NOT* gate, which operates as follows:

$$\begin{aligned} |\psi'\rangle = \mathbf{X}|\psi\rangle &= \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} \alpha \\ \beta \end{pmatrix} \\ &= \begin{pmatrix} \beta \\ \alpha \end{pmatrix} \equiv \beta|0\rangle + \alpha|1\rangle. \end{aligned} \tag{2.20}$$

Hence, Pauli-$\mathbf{X}$ swaps the probabilities of the computational basis states $|0\rangle$ onto $|1\rangle$. In terms of Bloch sphere of Figure 2.2, the operation of Eq. (2.20) can be visualized as a 180° rotation about x-axis. Similarly, Pauli-$\mathbf{Y}$ of Eq. (2.18) corresponds to a 180° rotation about y-axis. It swaps the amplitudes of the two superimposed states and introduces a phase shift of π between the resulting superimposed states, which can be mathematically represented as:

$$\begin{aligned}|\psi'\rangle = \mathbf{Y}|\psi\rangle &= \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix} \begin{pmatrix} \alpha \\ \beta \end{pmatrix} \\ &= \begin{pmatrix} -i\beta \\ i\alpha \end{pmatrix} = -i(\beta|0\rangle - \alpha|1\rangle) \equiv e^{-i\pi/2}(\beta|0\rangle + e^{i\pi}\alpha|1\rangle). \end{aligned} \tag{2.21}$$

The Pauli-**Z** of Eq. (2.18) rotates the state about z-axis by 180°, i.e. shifts the angle φ of Figure 2.2 by π. Thus, it introduces a phase shift of π between the two superimposed states, i.e we have:

$$\begin{aligned}|\psi'\rangle = \mathbf{Z}|\psi\rangle &= \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \begin{pmatrix} \alpha \\ \beta \end{pmatrix} \\ &= \begin{pmatrix} \alpha \\ -\beta \end{pmatrix} = \alpha|0\rangle - \beta|1\rangle) \equiv \alpha|0\rangle + e^{i\pi}\beta|1\rangle. \end{aligned} \tag{2.22}$$

Hence, **X**-gate is equivalent to bit flip, **Z**-gate to phase flip and **Y**-gate is a phase flip followed by a bit flip. Therefore, the three Pauli matrices are related as follows:

$$\mathbf{Y} = i\mathbf{XZ}, \tag{2.23}$$

where $i = \sqrt{-1}$.

- **Hadamard Gate:** Hadamard gate is a single-qubit gate, which has no classical counterpart. It maps a pure state $|0\rangle$ or $|1\rangle$ onto a superposition of the computational basis vectors as shown below [18]:

$$|0\rangle \rightarrow \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle), \qquad |1\rangle \rightarrow \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle). \tag{2.24}$$

The resulting states of Eq. (2.24) are also known as Hadamard basis states. In matrix notation, the operation of Eq. (2.24) can be modeled as:

$$\mathbf{H} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ -1 & 1 \end{pmatrix}. \tag{2.25}$$

Furthermore, the Hadamard operation can be visualized as a 90° rotation about y-axis on a Bloch sphere of Figure 2.2, followed by a 180° rotation about x-axis.

- **Phase Gate:** Phase gate **S** is a $\pi/2$-phase-shift or equivalently i-phase-shift gate, which shifts the angle φ of Figure 2.2 by $\pi/2$, which can be mathematically encapsulated as follows [18]:

$$\mathbf{S} = \begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix}. \tag{2.26}$$

More explicitly, **S** changes the phase of $|1\rangle$ by $\pi/2$, while leaving $|0\rangle$ intact. This may be visualized as a 90° rotation about the z-axis. Hence, the phase gate **S** is related to the **Z**-gate as follows:

$$\mathbf{S} = \sqrt{\mathbf{Z}}. \tag{2.27}$$

- **Controlled-NOT Gate:** Controlled-NOT (CNOT) gate is a multi-qubit gate, which is analogous to a classical XOR gate. It takes two inputs, i.e. a control qubit and a target qubit. When the control qubit is in state $|1\rangle$, the target undergoes a NOT operation. Otherwise, it is left unchanged. The operation of a CNOT gate can be modeled as [102]:

$$\text{CNOT}(|a, x\rangle) \equiv |a, a \oplus x\rangle. \tag{2.28}$$

The corresponding matrix is given by:

$$\text{CNOT} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}. \tag{2.29}$$

2.7 The Pauli Group

The Pauli matrices **I**, **X**, **Y** and **Z** anti-commute with each other, as shown below:

$$\begin{aligned} \mathbf{XY} = i\mathbf{Z},\ \mathbf{YX} = -i\mathbf{Z} \rightarrow \mathbf{XY} = -\mathbf{YX} \\ \mathbf{YZ} = i\mathbf{X},\ \mathbf{ZY} = -i\mathbf{X} \rightarrow \mathbf{YZ} = -\mathbf{ZY} \\ \mathbf{ZX} = i\mathbf{Y},\ \mathbf{XZ} = -i\mathbf{Y} \rightarrow \mathbf{ZX} = -\mathbf{XZ} \end{aligned} \tag{2.30}$$

Therefore, the set of Pauli matrices constitute the non-Abelian group of Table 2.1, which is obtained by multiplying the left-most column with the top row. For example, the second row of Table 2.1 is computed by multiplying Pauli-**X** with the set of Pauli operators, which constitute the top row of Table 2.1, i.e. we have:

$$\mathbf{XI} = \mathbf{X}, \quad \mathbf{XX} = \mathbf{I}, \quad \mathbf{XY} = i\mathbf{Z}, \quad \mathbf{XZ} = -i\mathbf{Y}. \tag{2.31}$$

The resultant non-Abelian group of Table 2.1 is termed as a single qubit 'Pauli group', which is denoted by $\mathcal{G}_1$. Hence, the Pauli group $\mathcal{G}_1$ consists of all the Pauli matrices (**I**, **X**, **Y** and **Z**) together with the multiplicative factors ± 1 and $\pm i$, i.e. we have:

$$\mathcal{G}_1 \equiv \{\pm\mathbf{I}, \pm i\mathbf{I}, \pm\mathbf{X}, \pm i\mathbf{X}, \pm\mathbf{Y}, \pm i\mathbf{Y}, \pm\mathbf{Z}, \pm i\mathbf{Z}\}. \tag{2.32}$$

The general N-qubit Pauli group $\mathcal{G}_N$ is an N-fold tensor product of $\mathcal{G}_1$, i.e. we have $\mathcal{G}_N = \mathcal{G}_1^{\otimes N}$. More specifically, consider an N-qubit system, where $\mathcal{P}_i$ denotes the Pauli operator acting on the ith qubit, which may be mathematically modeled as follows [98]:

$$\mathcal{P}_i \triangleq \mathbf{I}_1 \otimes \cdots \otimes \mathbf{I}_{i-1} \otimes \mathcal{P}_i \otimes \mathbf{I}_{i+1} \otimes \cdots \otimes \mathbf{I}_N. \tag{2.33}$$

Then $\mathcal{G}_N$ is given by:

$$\mathcal{G}_N = \mathcal{G}_1^{\otimes n} = \epsilon \mathcal{P}_1 \otimes \mathcal{P}_1 \otimes \cdots \otimes \mathcal{P}_N, \tag{2.34}$$

where $\epsilon \in \{\pm 1, \pm i\}$ and $\mathcal{P}_i \in \{\mathbf{I}, \mathbf{X}, \mathbf{Y}, \mathbf{Z}\}$.

	I	**X**	**Y**	**Z**
I	$\mathbf{I}$	$\mathbf{X}$	$\mathbf{Y}$	$\mathbf{Z}$
X	$\mathbf{X}$	$\mathbf{I}$	$i\mathbf{Z}$	$-i\mathbf{Y}$
Y	$\mathbf{Y}$	$-i\mathbf{Z}$	$\mathbf{I}$	$i\mathbf{X}$
Z	$\mathbf{Z}$	$i\mathbf{Y}$	$-i\mathbf{X}$	$\mathbf{I}$

Table 2.1: Single qubit Pauli group $\mathcal{G}_1$ [98].

2.8 Quantum Channels

Recall from Section 1.1 that quantum decoherence is a major impediment to the practical realization of quantum computation and communication systems. Decoherence may be viewed as the unwanted entanglement of the qubit with the environment, which perturbs its coherent quantum state. Let us consider the decoherence process for the basis states $|0\rangle$ and $|1\rangle$, which can be encapsulated as [25]:

$$\begin{aligned} |e_0\rangle|0\rangle &\rightarrow |a_0\rangle|0\rangle + |a_1\rangle|1\rangle, \\ |e_0\rangle|1\rangle &\rightarrow |a_2\rangle|0\rangle + |a_3\rangle|1\rangle, \end{aligned} \tag{2.35}$$

where $|e_0\rangle$ is the state of the environment before interaction, while $|a_i\rangle$ denotes the ith post-decoherence state of the environment (not necessarily orthogonal or normalized), which ensures that the overall evolution of Eq. (2.35) is unitary. Consequently, a qubit in the state $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$ decoheres as:

$$|e_0\rangle|\psi\rangle \rightarrow \alpha\left(|a_0\rangle|0\rangle + |a_1\rangle|1\rangle\right) + \beta\left(|a_2\rangle|0\rangle + |a_3\rangle|1\rangle\right). \tag{2.36}$$

Eq. (2.36) can be rearranged as:

$$\begin{aligned} |e_0\rangle|\psi\rangle \rightarrow & \frac{1}{2}\left(|a_0\rangle + |a_3\rangle\right)\left(\alpha|0\rangle + \beta|1\rangle\right) + \frac{1}{2}\left(|a_0\rangle - |a_3\rangle\right)\left(\alpha|0\rangle - \beta|1\rangle\right) + \\ & \frac{1}{2}\left(|a_1\rangle + |a_2\rangle\right)\left(\alpha|1\rangle + \beta|0\rangle\right) + \frac{1}{2}\left(|a_1\rangle - |a_2\rangle\right)\left(\alpha|1\rangle - \beta|0\rangle\right), \end{aligned} \tag{2.37}$$

which is equivalent to:

$$\begin{aligned} |e_0\rangle|\psi\rangle \rightarrow & \frac{1}{2}\left(|a_0\rangle + |a_3\rangle\right)\mathbf{I}|\psi\rangle + \frac{1}{2}\left(|a_0\rangle - |a_3\rangle\right)\mathbf{Z}|\psi\rangle + \\ & \frac{1}{2}\left(|a_1\rangle + |a_2\rangle\right)\mathbf{X}|\psi\rangle + \frac{-i}{2}\left(|a_1\rangle - |a_2\rangle\right)\mathbf{Y}|\psi\rangle. \end{aligned} \tag{2.38}$$

Hence, as we may observe in Eq. (2.37), the state ψ is mapped onto a linear combination of the original state (Pauli-**I** operation), phase flipped state (Pauli-**Z** operation), bit flipped state (Pauli-**X** operation) and both phase and bit flipped state (Pauli-**Y** operation). In the process of the quantum error correction, the superimposed state of Eq. (2.37) collapses to one of these four possibilities upon measurement. Therefore, the overall decoherence process can be visualized as inflicting bit errors or phase errors or possibly both errors on the qubit, which was also depicted in Figure 1.2. Alternatively, we may intuitively argue that since any arbitrary unitary operator can be expressed as a linear combination of the Pauli-**I**, Pauli-**Z**, Pauli-**X** and Pauli-**Y** operators, decoherence can also be described in terms of these Pauli operators. Hence, quantum channel models are defined on the basis of the set of Pauli operators.

A quantum channel can be used for modeling imperfections in quantum hardware, namely, faults resulting from quantum decoherence and quantum gates. Furthermore, a quantum channel can also model quantum-state flips imposed by the transmission medium, including free-space wireless channels and optical fiber links, when qubits are transmitted across these media. Some of the commonly used quantum channel models are discussed below [18]:

- **Bit-Flip Channel:** Analogous to a classical binary symmetric channel, a bit-flip channel characterized by the probability p maps the basis state $|0\rangle \rightarrow |1\rangle$ and $|1\rangle \rightarrow |0\rangle$ with a probability of p. The associated set of operators are defined as:

$$\mathbf{E}_0 = \sqrt{1-p}\,\mathbf{I} = \sqrt{1-p}\begin{pmatrix}1 & 0\\ 0 & 1\end{pmatrix}, \quad \mathbf{E}_1 = \sqrt{p}\,\mathbf{X} = \sqrt{p}\begin{pmatrix}0 & 1\\ 1 & 0\end{pmatrix}, \tag{2.39}$$

  where $\mathbf{E}_j$ is the jth operator, which maps a given channel input onto the corresponding output.

- **Phase-Flip Channel:** A phase-flip channel characterized by the probability p inflicts a Pauli-**Z** error on the transmitted qubit with a probability of p, which can be encapsulated as:

$$\mathbf{E}_0 = \sqrt{1-p}\,\mathbf{I} = \sqrt{1-p}\begin{pmatrix}1 & 0\\ 0 & 1\end{pmatrix}, \quad \mathbf{E}_1 = \sqrt{p}\,\mathbf{Z} = \sqrt{p}\begin{pmatrix}1 & 0\\ 0 & -1\end{pmatrix}. \tag{2.40}$$

- **Bit-Phase-Flip Channel:** A bit-phase-flip channel characterized by the probability p inflicts a Pauli-**Y** error on the transmitted qubit with a probability of p, which can be defined as:

$$\mathbf{E}_0 = \sqrt{1-p}\,\mathbf{I} = \sqrt{1-p}\begin{pmatrix}1 & 0\\ 0 & 1\end{pmatrix}, \quad \mathbf{E}_1 = \sqrt{p}\,\mathbf{Y} = \sqrt{p}\begin{pmatrix}0 & -i\\ i & 0\end{pmatrix}. \tag{2.41}$$

- **Depolarizing Channel:** A depolarizing channel characterized by the probability p inflicts a bit-error (Pauli-**X**) or a phase-error (Pauli-**Z**) or both bit and phase errors (Pauli-**Y**) with a probability of $p/3$ each, which can be expressed as:

$$\begin{aligned} \mathbf{E}_0 &= \sqrt{1-p}\,\mathbf{I} = \sqrt{1-p}\begin{pmatrix}1 & 0\\ 0 & 1\end{pmatrix}, & \mathbf{E}_1 &= \sqrt{\frac{p}{3}}\,\mathbf{X} = \sqrt{\frac{p}{3}}\begin{pmatrix}0 & 1\\ 1 & 0\end{pmatrix}, \\ \mathbf{E}_2 &= \sqrt{\frac{p}{3}}\,\mathbf{Z} = \sqrt{\frac{p}{3}}\begin{pmatrix}1 & 0\\ 0 & -1\end{pmatrix}, & \mathbf{E}_3 &= \sqrt{\frac{p}{3}}\,\mathbf{Y} = \sqrt{\frac{p}{3}}\begin{pmatrix}0 & -i\\ i & 0\end{pmatrix}. \end{aligned} \tag{2.42}$$

  The depolarizing channel of Eq. (2.42) may be referred to as a symmetric channel, since the three types of errors occur with equal probabilities. By contrast, if the Pauli-**X**, Pauli-**Z** and Pauli-**Y** errors occur with different probabilities, the channel is termed as being asymmetric [103, 104].

- **Amplitude Damping Channel:** Amplitude damping channel models the loss of energy from a quantum system. An amplitude damping channel characterized by the damping probability γ, or more specifically the probability of losing a photon, is modeled as:

$$\mathbf{E}_0 = \begin{pmatrix} 1 & 0 \\ 0 & \sqrt{1-\gamma} \end{pmatrix}, \quad \mathbf{E}_1 = \begin{pmatrix} 0 & \sqrt{\gamma} \\ 0 & 0 \end{pmatrix}. \tag{2.43}$$

  According to Eq. (2.43), the operator $\mathbf{E}_1$ changes the state $|1\rangle$ to $|0\rangle$ depicting that energy is lost to the environment, while the operator $\mathbf{E}_0$ reduces the amplitude of the state $|1\rangle$ because energy is dissipated, which makes it less likely to encounter the state $|1\rangle$.

- **Phase Damping Channel:** Phase damping characterizes the loss of quantum information without the loss of energy. It may include for example the scattering of photons, or perturbation of electronic states caused by the stray electrical charges. Phase damping channel can be described as follows:

$$\mathbf{E}_0 = \begin{pmatrix} 1 & 0 \\ 0 & \sqrt{1-\lambda} \end{pmatrix}, \quad \mathbf{E}_1 = \begin{pmatrix} 0 & 0 \\ 0 & \sqrt{\lambda} \end{pmatrix}, \tag{2.44}$$

  where λ is the probability of scattering of a photon (without loss of energy). Similar to the amplitude damping channel, a $\mathbf{E}_0$ reduces the amplitude of state $|1\rangle$. On the other hand, the operator $\mathbf{E}_1$ destroys the state $|0\rangle$, while reduces the amplitude of state $|1\rangle$.

In this treatise, we will only consider the widely used symmetric depolarizing channel model of Eq. (2.42) [33, 85, 47, 105].

2.9 Summary and Conclusions

This chapter provides a brief introduction to quantum information. We commence the discussion in Section 2.2 with the introduction of qubits, which are analogous to the classical bits but have the additional capability of existing in a superposition of the two classical states, namely 0 and 1. A qubit is conventionally represented as a linear combination of the computational basis states $|0\rangle$ and $|1\rangle$, i.e. $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$, which collapses to the classical states 0 or 1 upon measurement. More specifically, $|\alpha|^2$ and $|\beta|^2$ denote the probability of finding the qubit in the states $|0\rangle$ and $|1\rangle$, respectively, upon measurement. This property has in turn made quantum parallelism a reality, enabling N-qubit quantum computers to concurrently process 2^N computations, which was discussed in Section 2.3. In Section 2.4, we detailed the no-cloning theorem, which states that the laws of quantum mechanics do not permit the cloning of an arbitrary quantum states. We next presented the concept of entanglement in Section 2.5, which is in essence a mysterious correlation that exists despite the spatial separation between the entangled qubits. More specifically, entangled qubits cannot be decomposed into the tensor product of the individual qubits. We next discussed the quantum unitary transformations in Section 2.6, which are also summarized in Table 2.2. In Section 2.7, we introduced the notion of a Pauli group. A single qubit Pauli group is a closed multiplicative group of the Pauli matrices, while an N-qubit Pauli group is a tensor product of N single qubit Pauli groups. Finally, we laid down various quantum channel models in Section 2.8, which are summarized in Table 2.3. In the rest of the thesis, we will only focus on the depolarizing channel.

Gate	**Symbol**	**Operation**	**Matrix**
Pauli-**I**	**I**	Identity operation.	$\begin{pmatrix}1 & 0\\ 0 & 1\end{pmatrix}$
Pauli-**X**	**X**	Bit flip.	$\begin{pmatrix}0 & 1\\ 1 & 0\end{pmatrix}$
Pauli-**Z**	**Z**	Phase flip.	$\begin{pmatrix}1 & 0\\ 0 & -1\end{pmatrix}$
Pauli-**Y**	**Y**	Bit and phase flip.	$\begin{pmatrix}0 & -i\\ i & 0\end{pmatrix}$
Hadamard	**H**	Maps a pure state $\|0\rangle$ or $\|1\rangle$ onto a superposition of the basis states.	$\frac{1}{\sqrt{2}}\begin{pmatrix}1 & 1\\ -1 & 1\end{pmatrix}$
Phase	**S**	Changes the phase of the basis state $\|1\rangle$ by $\pi/2$, while leaving $\|0\rangle$ intact.	$\begin{pmatrix}1 & 0\\ 0 & i\end{pmatrix}$
Controlled-NOT	CNOT	A 2-qubit gate, which flips the target qubit when the control qubit is in the state $\|1\rangle$.	$\begin{pmatrix}1 & 0 & 0 & 0\\ 0 & 1 & 0 & 0\\ 0 & 0 & 0 & 1\\ 0 & 0 & 1 & 0\end{pmatrix}$

Table 2.2: Summary of the quantum unitary operators of Section 2.2.

Channel	**Definition**
Bit-flip	Inflicts a Pauli-**X** error with a probability of p.
Phase-flip	Inflicts a Pauli-**Z** error with a probability of p.
Bit-phase-flip	Inflicts a Pauli-**Y** error with a probability of p.
Depolarizing	Inflicts a Pauli-**X**, **Z** or **Y** error with a probability of $p/3$ each.
Amplitude damping	Models the energy dissipation from a quantum system with a damping probability γ.
Phase damping	Models the loss of quantum information without any energy dissipation, which is characterized by the scattering probability λ.

Table 2.3: Summary of the quantum channel models of Section 2.8.

Chapter 3

Near-Capacity Code Designs for Entanglement-Assisted Classical Communication

3.1 Introduction

Quantum-based communication constitutes an attractive solution for absolute secure transmission [18]. More explicitly, any 'measurement' or 'observation' of the transmitted qubits by the eavesdropper perturbs the associated quantum superposition, hence intimating the parties concerned [18]. In this context, entanglement-assisted transmission of classical information over quantum channels is of particular significance. This idea was conceived by Bennett [28] in his widely-cited 2-qubit SuperDense (2SD) coding protocol, which transmits 2 classical bits per channel use (cbits/use) over a noiseless quantum channel with the aid of a pre-shared maximally entangled qubit. The corresponding Entanglement-Assisted Classical Capacity (EACC) of the so-called quantum depolarizing channel was quantified in [106, 107].

Analogous to Shannon's well-known capacity theorem conceived for classical channels, the EACC quantifies the capacity limit of reliable transmission of classical information over a noisy quantum channel, when an unlimited amount of noiseless entanglement is shared between the transmitter and the receiver. The corresponding 'classical-quantum-classical' conversion based transmission model, whereby classical information is transmitted over a quantum channel with the aid of the SuperDense (SD) coding protocol, is depicted in Figure 3.1. Here, Alice intends to transmit her 2-bit classical message x to Bob using a 2-qubit maximally entangled state $|\psi_x\rangle^{AB}$, where A denotes the information qubit, while B is a pre-shared entangled qubit transmitted over a noiseless channel. More specifically,

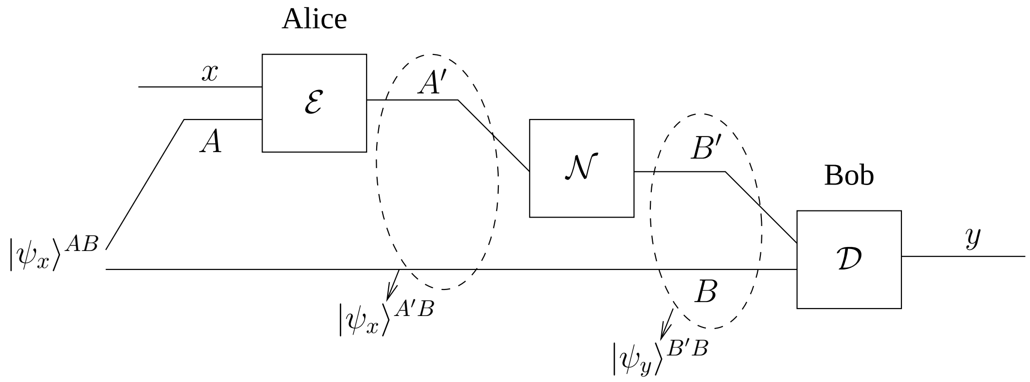


Figure 3.1: Classical-quantum-classical transmission model employing 2-qubit SD.

the pre-shared qubit B is transmitted to the receiver before the actual transmission commences, for example it can be shared during the off-peak hours, when the channel is under-utilized. The classical message x is encoded by the block $\mathcal{E}$ of Figure 3.1 into the corresponding quantum state using the 2SD coding protocol of [28]. The processed qubit A' is passed through a quantum depolarizing channel, which is denoted as $\mathcal{N}^{A' \to B'}$. At the block $\mathcal{D}$ of Figure 3.1, the receiver Bob performs symbol-by-symbol Bell-basis measurement[1] [18, 10] on the received state $|\psi_y\rangle^{B'B}$, yielding the 2-bit classical message y. This transmission model was extended to a distributed network in [108], whereby the 2SD scheme of [28] was generalized to an N-particle system with the aid of an N-qubit entangled state. The resultant protocol facilitates for the receiver to detect messages from $(N-1)$ users with the aid of a single N-qubit entangled quantum state as well as a single joint quantum measurement, albeit this is achieved at the cost of a reduced EACC. Recently, Chiuri *et al.* [109] experimentally determined the achievable EACC of a quantum depolarizing channel, which paves the way for the practical implementation of future quantum-based communication systems. However, reliable transmission is impossible without efficient error correction codes.

Inspired by the near-capacity performance of concatenated classical code designs, in this chapter we design both bit-based as well as symbol-based concatenated classical-quantum code structures with the aid of EXtrinsic Information Transfer (EXIT) charts for the sake of achieving a performance close to the EACC of the quantum depolarizing channel. More explicitly, our novel contributions are as follows [5, 7]:

- We have conceived an SD-based near-capacity design for entanglement-assisted classical communication over a quantum depolarizing channel. Our design, referred to as an IRCC-URC-2SD arrangement, incorporates a classical Irregular Convolutional Code (IRCC) and a Unity Rate Code (URC). We have also introduced a soft-decision aided SD decoder for facilitating iterative decoding.

[1] Bell-basis measurement is a joint measurement on a 2-qubit composite system for the sake of detecting the orthonormal Bell states of Eq. (2.15).

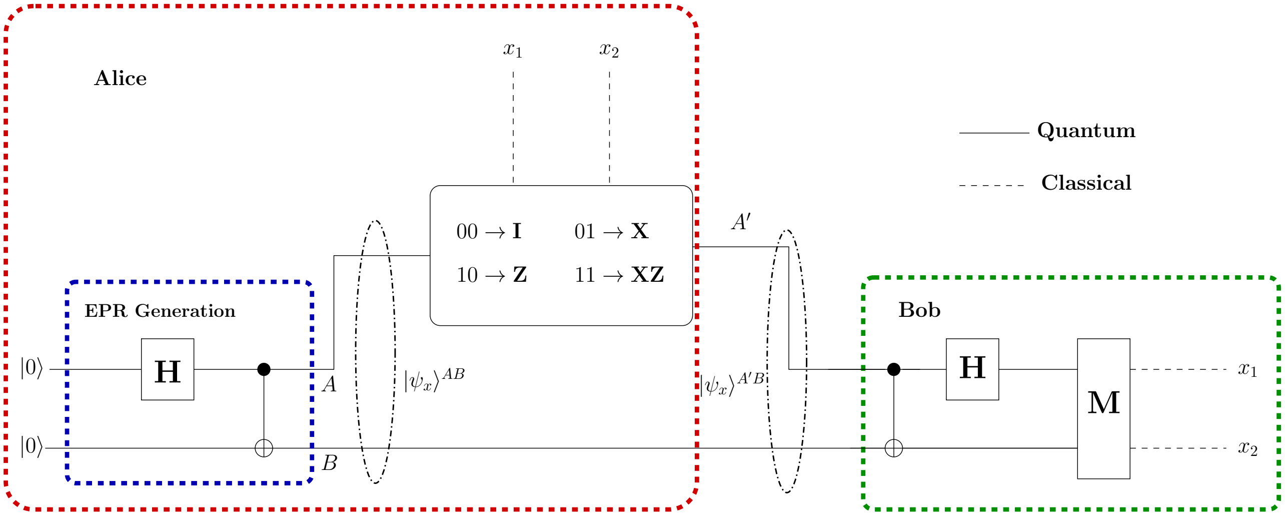


Figure 3.2: The quantum circuit of 2-qubit superdense coding. Alice generates the Einstein-Podolsky-Rosen (EPR) pair $|\psi_x\rangle^{AB}$ using a **H** gate and CNOT gate, respectively. Qubit A is used for encoding the 2-bit message, while the qubit B is pre-shared with Bob. Bob performs Bell-basis measurement on the received state $|\psi_x\rangle^{A'B}$, yielding the original classical message.

- Our IRCC-URC-2SD design is bit-based, thereby incurring an capacity loss due to symbol-to-bit conversion. To circumvent this capacity loss, we have proposed an alternative iterative code design referred to as a symbol-based CC-URC-2SD. Our symbol-based design incorporates a single Convolutional Code (CC) as the outer component, while the URC and 2SD schemes constitute the amalgamated symbol-based inner code.

The rest of the chapter is laid out as follows. We review the SD coding protocol in Section 3.2, while the EACC of N-qubit SD schemes is investigated in Section 3.3. The bit-based system model and our near-capacity design are detailed in Sections 3.4 and 3.5, respectively, while the corresponding simulation results are discussed in Section 3.6. Next we have detailed our symbol-based scheme in Section 3.7 and the associated results are discussed in Section 3.8. Finally, our conclusions are offered in Section 3.9.

3.2 Review of the Superdense Coding Protocol

3.2.1 2-Qubit Superdense Coding

The 2SD protocol [28] invokes the peculiar law of quantum entanglement for transmitting two classical bits using a single qubit. Figure 3.2 shows the quantum circuit of 2SD [28]. Here, Alice intends to transmit two bits of classical information $x = (x_1\ x_2)$ to Bob. Alice initiates the process by generating a maximally entangled Bell state, also referred to as the Einstein-Podolsky-Rosen (EPR) pair [28], which is given by [28]:

$$|\psi_x\rangle^{AB} = \frac{|00\rangle + |11\rangle}{\sqrt{2}}. \tag{3.1}$$

This is achieved by applying the Hadamard gate (**H**) of Eq. (2.24) and the CNOT gate of Eq. (2.28) to two qubits initialized to the state $|0\rangle$, as depicted in the 'EPR Generation' block of Figure 3.2. More explicitly, the Einstein-Podolsky-Rosen pair generation proceeds as follows:

Step 1: Apply the Hadamard gate to the first qubit:

$$|00\rangle \rightarrow \frac{1}{\sqrt{2}}\left(|0\rangle + |1\rangle\right)|0\rangle. \tag{3.2}$$

Step 2: Apply the CNOT gate to the second qubit, which is controlled by the first qubit:

$$\frac{1}{\sqrt{2}}\left(|00\rangle + |10\rangle\right) \rightarrow \frac{1}{\sqrt{2}}\left(|00\rangle + |11\rangle\right) \equiv |\psi_x\rangle^{AB}. \tag{3.3}$$

The resultant qubit A is used for encoding the 2-bit classical message, while the qubit B may be pre-shared with Bob before actual transmission takes place, for example during the instances, when the channel is not busy. During the encoding procedure, Alice performs either the **I**, **X**, **Z** or **XZ** operation of Table 3.1 on her qubit A, depending on her 2-bit message. More specifically, the classical message is embedded in the qubit A as follows:

- For $(x_1\ x_2) = (0\ 0)$, do not apply any operation,
- For $(x_1\ x_2) = (0\ 1)$, apply the **X** gate of Eq. (2.18) to A,

$$\frac{1}{\sqrt{2}}\left(|00\rangle + |11\rangle\right) \rightarrow \frac{1}{\sqrt{2}}\left(|10\rangle + |01\rangle\right). \tag{3.4}$$

- For $(x_1\ x_2) = (0\ 1)$, apply the **Z** gate of Eq. (2.18) to A,

$$\frac{1}{\sqrt{2}}\left(|00\rangle + |11\rangle\right) \rightarrow \frac{1}{\sqrt{2}}\left(|00\rangle - |11\rangle\right). \tag{3.5}$$

- For $(x_1\ x_2) = (0\ 1)$, apply the **Z** gate followed by the **X** gate to A,

$$\frac{1}{\sqrt{2}}\left(|00\rangle + |11\rangle\right) \rightarrow \frac{1}{\sqrt{2}}\left(|10\rangle - |01\rangle\right). \tag{3.6}$$

This classical to quantum mapping is summarized in Table 3.1.

Alice sends the appropriately processed qubit A' over the quantum channel to Bob. Let us assume having a noiseless channel here. Since the four Bell states $\psi_x\rangle^{A'B}$ of Table 3.1 are orthonormal, they are distinguishable at the receiver. Recall that qubit B is pre-shared with Bob. Upon receiving the processed qubit A', Bob performs a collective Bell-basis measurement on the received state $|\psi_x\rangle^{A'B}$, which is carried out as follows:

$(x_1\ x_2)$	A	$\lvert\psi_x\rangle^{A'B}$
0 0	$\mathbf{I}$	$\lvert 00\rangle + \lvert 11\rangle$
0 1	$\mathbf{X}$	$\lvert 10\rangle + \lvert 01\rangle$
1 0	$\mathbf{Z}$	$\lvert 00\rangle - \lvert 11\rangle$
1 1	$\mathbf{XZ}$	$\lvert 10\rangle - \lvert 01\rangle$

Table 3.1: Classical-to-quantum mapping for 2-qubit superdense coding. *(The normalization factor* $\frac{1}{\sqrt{2}}$ *is ignored for simplicity.)*

Step 1: Apply the CNOT gate to qubit B, which is controlled by the qubit A':

$$\begin{aligned}
|\psi_{00}\rangle &= \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle) \rightarrow \frac{1}{\sqrt{2}}(|00\rangle + |10\rangle),\\
|\psi_{01}\rangle &= \frac{1}{\sqrt{2}}(|10\rangle + |01\rangle) \rightarrow \frac{1}{\sqrt{2}}(|11\rangle + |01\rangle),\\
|\psi_{10}\rangle &= \frac{1}{\sqrt{2}}(|00\rangle - |11\rangle) \rightarrow \frac{1}{\sqrt{2}}(|00\rangle - |10\rangle),\\
|\psi_{11}\rangle &= \frac{1}{\sqrt{2}}(|10\rangle - |01\rangle) \rightarrow \frac{1}{\sqrt{2}}(|11\rangle - |01\rangle).
\end{aligned} \tag{3.7}$$

Step 2: Apply the Hadamard gate to the first qubit:

$$\begin{aligned}
\frac{1}{\sqrt{2}}(|00\rangle + |10\rangle) &\rightarrow \frac{1}{\sqrt{2}}\frac{1}{\sqrt{2}}((|0\rangle + |1\rangle)|0\rangle + (|0\rangle - |1\rangle)|0\rangle) \equiv |00\rangle,\\
\frac{1}{\sqrt{2}}(|11\rangle + |01\rangle) &\rightarrow \frac{1}{\sqrt{2}}\frac{1}{\sqrt{2}}((|0\rangle - |1\rangle)|1\rangle) + (|0\rangle + |1\rangle)|1\rangle) \equiv |01\rangle,\\
\frac{1}{\sqrt{2}}(|00\rangle - |10\rangle) &\rightarrow \frac{1}{\sqrt{2}}\frac{1}{\sqrt{2}}((|0\rangle + |1\rangle)|0\rangle - (|0\rangle - |1\rangle)|0\rangle) \equiv |10\rangle,\\
\frac{1}{\sqrt{2}}(|11\rangle - |01\rangle) &\rightarrow \frac{1}{\sqrt{2}}\frac{1}{\sqrt{2}}((|0\rangle - |1\rangle)|1\rangle - (|0\rangle + |1\rangle)|1\rangle) \equiv -|11\rangle.
\end{aligned} \tag{3.8}$$

Step 3: Measure the first qubit:

$$\begin{aligned}
|00\rangle &\rightarrow 00,\\
|01\rangle &\rightarrow 01,\\
|10\rangle &\rightarrow 10,\\
-|11\rangle &\rightarrow 11.
\end{aligned} \tag{3.9}$$

Hence, Alice transmits only a single qubit, namely A', to Bob through the quantum channel for communicating a 2-bit classical message, resulting in a transmission rate of 2 cbits/use. Indeed, the transmission of the pre-shared entangled qubit B also consumes transmission resources, hence the

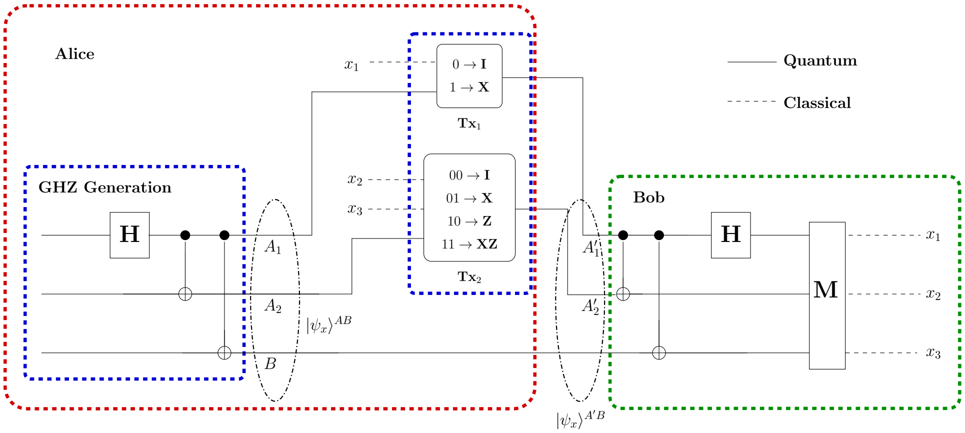


Figure 3.3: The quantum circuit for 3-qubit superdense coding. Alice generates the Greenberger-Horne-Zeilinger state $|\psi_x\rangle^{AB}$ using a **H** gate and CNOT gates respectively. Qubits A_1 and A_2 are used for encoding the 3-bit message, while the qubit B is pre-shared with Bob. Bob measures the received state $|\psi_x\rangle^{A'B}$ in the orthonormal basis, yielding the original classical message.

overall transmission requirements remain the same as in a classical scenario. However, traditionally this is considered to be less of a problem, because the entangled qubit may be shared during off-peak hours, when the network is under-utilized [28]. Alternatively, if both the transmitter and receiver are mobile, sharing may take place if and when they are close to each other [110].

3.2.2 N-Qubit Superdense Coding

N-qubit SuperDense coding (NSD) [108] is a generalization of the 2SD scheme to a multi-qubit channel, which facilitates for the receiver to read messages from multiple users with the aid of a single entangled quantum state as well as using a single joint quantum measurement. Let us consider a system supporting N users sharing an N-qubit Greenberger-Horne-Zeilinger (GHZ) state [111], where each user possesses one qubit. Furthermore, one of the users intends to receive information from the $(N-1)$ other users (the source transmitters in this case). For N qubits, there are 2^N unitary operations, which map the initial N-qubit state onto a unique quantum state. Therefore, the source transmitters mutually decide *a priori* to perform only certain operations on the qubit in their possession. Since there are 2^N operations and $(N-1)$ source transmitters, one transmitter can perform four operations on its qubit, while the remaining $(N-2)$ transmitters can perform two operations. Consequently, the former can transmit 2 cbits/use, while the latter can only transmit 1 cbit/use. The overall rate is therefore $\frac{N}{N-1}$ cbits/use. The receiver makes a collective measurement on the N-qubit state for determining the classical information transmitted by each transmitter.

Let us now consider the 3-qubit system of Figure 3.3, where two source transmitters intend to transmit three classical information bits to a receiver over a quantum channel. Since there are three users, the corresponding 3-qubit Greenberger-Horne-Zeilinger state, which is shared amongst the users,

$(x_1\ x_2\ x_3)$	A_1	A_2	$\lvert\psi_x\rangle^{A'B}$
0 0 0	**I**	**I**	$\lvert 000\rangle + \lvert 111\rangle$
0 0 1	**I**	**X**	$\lvert 010\rangle + \lvert 101\rangle$
0 1 0	**I**	**Z**	$\lvert 000\rangle - \lvert 111\rangle$
0 1 1	**I**	**XZ**	$\lvert 010\rangle - \lvert 101\rangle$
1 0 0	**X**	**I**	$\lvert 100\rangle + \lvert 011\rangle$
1 0 1	**X**	**X**	$\lvert 110\rangle + \lvert 001\rangle$
1 1 0	**X**	**Z**	$\lvert 100\rangle - \lvert 011\rangle$
1 1 1	**X**	**XZ**	$\lvert 110\rangle - \lvert 001\rangle)$

Table 3.2: Classical-to-quantum mapping for 3-qubit superdense coding. *The normalization factor* $\frac{1}{\sqrt{2}}$ *is ignored for simplicity.*

is given by:

$$\lvert\psi_x\rangle^{AB} = \frac{\lvert 000\rangle + \lvert 111\rangle}{\sqrt{2}}. \tag{3.10}$$

Here A constitutes a 2-qubit subsystem having qubits A_1 and A_2. The entangled triplet $\lvert\psi_x\rangle^{AB}$ is prepared by the first transmitter (Tx$_1$) and shared both with the second source transmitter (Tx$_2$) as well as the receiver, i.e. with Bob, before actual communication takes place. For simplicity, we may combine the two source transmitters into a single device, namely into Alice's device, who wishes to transmit 3 classical bits to Bob. More explicitly, one bit is transmitted from Tx$_1$ of Figure 3.3, while two bits are transmitted from Tx$_2$ to Bob. The qubits A_1 and A_2 are used by Tx$_1$ and Tx$_2$, respectively, for mapping three classical bits onto two processed qubits, while the entangled qubit B is pre-shared with Bob, as illustrated in Figure 3.3. Since Alice is in possession of the pair of qubits A_1 and A_2, she can perform the **I**, **X**, **Z** or **XZ** Pauli operations on each of these qubits for generating distinct output qubit states $\lvert\psi_x\rangle^{A'B}$, which are orthonormal and therefore distinguishable at the receiver. This may be achieved by adopting any set of classical-to-quantum mapping rules, which are capable of ensuring that the resultant states $\lvert\psi_x\rangle^{A'B}$ are orthonormal. Table 3.2 enlists one such mapping.

3.3 Entanglement-Assisted Classical Capacity

Under the assumptions discussed in Section 3.2.1, 2SD doubles the capacity of a noiseless quantum channel. Conventionally it is assumed that the pre-sharing of the entangled qubit destined from Alice to Bob takes place over a noiseless channel and only the processed qubit(s) is passed through a noisy quantum channel [106, 112]. The corresponding EACC of 2SD has already been derived in [106, 112] based on its equivalence to a 4-ary symmetric classical channel. In this section, we will generalize it to N-qubit SD by exploiting the well-known equivalent M-ary classical channel model ($M = 2^N$).

Let us recall that the capacity C of a classical channel is equivalent to the maximum value of the conveyed mutual information $I(x, y)$ between the transmitted symbol x and the received symbol y, i.e. we have [113]:

$$\mathrm{C} = \max_{\mathrm{P}(x)} I(x, y) \ = \max_{\mathrm{P}(x)} [\mathrm{H}(y) - \mathrm{H}(y|x)], \tag{3.11}$$

where H is the classical entropy function. Since C is maximized for equiprobable source symbols, the capacity of an M-ary classical channel is given by:

$$\mathrm{C} = \log_2 M - \mathrm{H}(y|x), \tag{3.12}$$

which is further defined as follows [114, 115]:

$$\mathrm{C} = \log_2 M + \mathrm{E}\left[\sum_{m=0}^{M-1} \mathrm{P}(y|x = x^{(m)}) \log_2 \mathrm{P}(y|x = x^{(m)})\right], \tag{3.13}$$

using Eq. (10) and (11) of [114]. Here E[.] is the expectation (or time average) of y and $x^{(m)}$ is the mth hypothetically transmitted classical message for $m \in \{0, 1, \ldots, M-1\}$.

Based on Eq. (3.13), the capacity of NSD coding relying on a single noiseless pre-shared entangled qubit may be readily expressed as:

$$\mathrm{C}_{Nsd} = \frac{N + \sum_{m=0}^{M-1} \mathrm{P}(y|x = x^{(m)}) \log_2 \mathrm{P}(y|x = x^{(m)})}{N-1} \quad \text{cbits/use}, \tag{3.14}$$

where $\mathrm{P}(y|x)$ denotes the transition probabilities of the induced classical channel[2].

Symbol-by-symbol measurements performed at the 2-qubit superdense decoder reduces the transmission model of Figure 3.1 to a 4-ary classical channel. Consequently, the channel transition probabilities of the induced classical channel may be encapsulated as:

$$\mathrm{P}(y|x = x^{(m)}) = \begin{cases} 1-p, & \text{if } \mathtt{E} = 0 \\ p/3, & \text{if } \mathtt{E} \in \{1, 2, 3\}, \end{cases} \tag{3.15}$$

where $m \in \{0, 1, 2, 3\}$. Furthermore, $\mathtt{E}$ is the decimal equivalent of the N-bit classical error e, which is induced by the depolarizing channel. More specifically, the N-bit classical error $e = [e_1, \ldots, e_i, \ldots, e_N]$ relates the i^{th} bit of $x = [x_1, \ldots, x_i, \ldots, x_N]$ to that of $y = [y_1, \ldots, y_i, \ldots, y_N]$ as follows:

$$y_i = x_i \oplus e_i \quad \text{or} \quad e_i = y_i \oplus x_i. \tag{3.16}$$

Substituting Eq. (3.15) in Eq. (3.14) yields the entanglement-assisted classical capacity of 2SD over a quantum depolarizing channel, i.e we have:

$$\mathrm{C}_{2sd} = 2 + (1-p)\log_2(1-p) + p\log_2(p/3), \tag{3.17}$$

[2] Due to the time-invariant nature of $\mathrm{P}(y|x)$, the average information is the same as the instantaneous value. The expectation operation of Eq. (3.13) can be therefore ignored.

which gives a maximum capacity of 2 cbits/use for the noiseless scenario.

Similarly, symbol-by-symbol measurements performed at the 3-qubit superdense decoder reduces the overall transmission to an 8-ary classical channel. However, unlike the 2SD scheme, now 2 qubits are transmitted over the noisy quantum channel. All the possible quantum channel errors along with the corresponding probabilities of occurrence and the resultant classical error patterns e are listed in Table 3.3. The resulting corrupted state $|\psi_y\rangle^{B'B}$ for the transmitted state $\frac{1}{\sqrt{2}}(|000\rangle + |111\rangle)$ is also tabulated in Table 3.3. It must be noted that different quantum errors may result in the same e, for example the third and sixth rows of Table 3.3 have the same classical error pattern. Consequently, we combined the probabilities corresponding to the same error patterns arriving at the following channel transition probabilities for the 3SD scheme:

$$\mathrm{P}(y|x = x^{(m)}) = \begin{cases} (1-p)^2 + p^2/9, & \text{if } \mathtt{E} \in \{0\} \\ (1-p)(p/3) + p^2/9, & \text{if } \mathtt{E} \in \{2,3,6,7\} \\ 2(1-p)(p/3), & \text{if } \mathtt{E} \in \{4\} \\ 2p^2/9, & \text{if } \mathtt{E} \in \{1,5\}, \end{cases} \tag{3.18}$$

where we have $m \in \{0, 1, \ldots, 7\}$. Eq. (3.18) may be substituted in Eq. (3.14) to compute the EACC of the 3SD scheme, when communicating over a quantum depolarizing channel. More explicitly, we have:

$$\mathrm{C}_{3sd} = \frac{3}{2} + \frac{1}{2}\sum_{m=0}^{7} \mathrm{P}(y|x = x^{(m)}) \log_2 \mathrm{P}(y|x = x^{(m)}). \tag{3.19}$$

Hence, the 3SD scheme has a maximum capacity of 1.5 cbits/use, when the channel is noiseless.

3.4 Bit-Based Code Structure

In this section we will present the architecture of our proposed classical-quantum communication system, which is designed for approaching the EACC of the NSD code with the aid of EXIT charts [116, 68, 117]. Figure 3.4 shows the general schematic of the proposed system, which employs a classical IRCC [118, 119] for achieving the near-capacity performance. Furthermore, a classical symbol-based recursive URC having a generator polynomial of $G(D) = \frac{1}{1+D}$ [68] is used as a precoder for reaching the $(1, 1)$ point of perfect decoding convergence in the EXIT chart [120]. We amalgamate our conceived soft-decision SD with the symbol-based URC, which hence constitutes an amalgamated inner component, while the bit-based IRCC is our outer component.

At the transmitter, the system is fed with classical bits $\{u_1\}$, which are encoded by an IRCC encoder. The IRCC-encoded bits $\{v_1\}$ of Figure 3.4 are then interleaved (π), yielding the permuted bit stream $\{u_2\}$, which is converted to symbols[3] and fed to the URC encoder of Figure 3.4. Classical to quantum domain conversion then takes place at the SD encoder, which maps the classical symbols x onto the orthogonal quantum states $|\psi_x\rangle^{A'B}$ using the entangled state $|\psi_x\rangle^{AB}$, as discussed in

[3] Bit-to-symbol convertor is assumed to be inside the URC Encoder block of Figure 3.4.

Error on A_1	**Error on A_2**	$\|\psi_y\rangle^{B'B}$	**Error** $(e/\mathtt{E})$	**Error Probability**
I	**I**	$\|000\rangle + \|111\rangle$	000/0	$(1-p)(1-p)$
X	**I**	$\|100\rangle + \|011\rangle$	011/3	$(p/3)(1-p)$
Z	**I**	$\|000\rangle - \|111\rangle$	100/4	$(p/3)(1-p)$
Y	**I**	$\|100\rangle - \|011\rangle$	111/7	$(p/3)(1-p)$
I	**X**	$\|010\rangle + \|101\rangle$	010/2	$(1-p)(p/3)$
I	**Z**	$\|000\rangle - \|111\rangle$	100/4	$(1-p)(p/3)$
I	**Y**	$\|010\rangle - \|101\rangle$	110/6	$(1-p)(p/3)$
X	**X**	$\|110\rangle + \|001\rangle$	001/1	$(p/3)(p/3)$
Z	**X**	$\|010\rangle - \|101\rangle$	110/6	$(p/3)(p/3)$
Y	**X**	$\|110\rangle - \|001\rangle$	101/5	$(p/3)(p/3)$
X	**Z**	$\|100\rangle - \|011\rangle$	111/7	$(p/3)(p/3)$
Z	**Z**	$\|000\rangle + \|111\rangle$	000/0	$(p/3)(p/3)$
Y	**Z**	$\|100\rangle + \|011\rangle$	011/3	$(p/3)(p/3)$
X	**Y**	$\|110\rangle - \|001\rangle$	101/5	$(p/3)(p/3)$
Z	**Y**	$\|010\rangle + \|101\rangle$	010/2	$(p/3)(p/3)$
Y	**Y**	$\|110\rangle + \|001\rangle$	001/1	$(p/3)(p/3)$

Table 3.3: List of all the possible quantum errors when the first and second qubit, A_1 and A_2 respectively, of

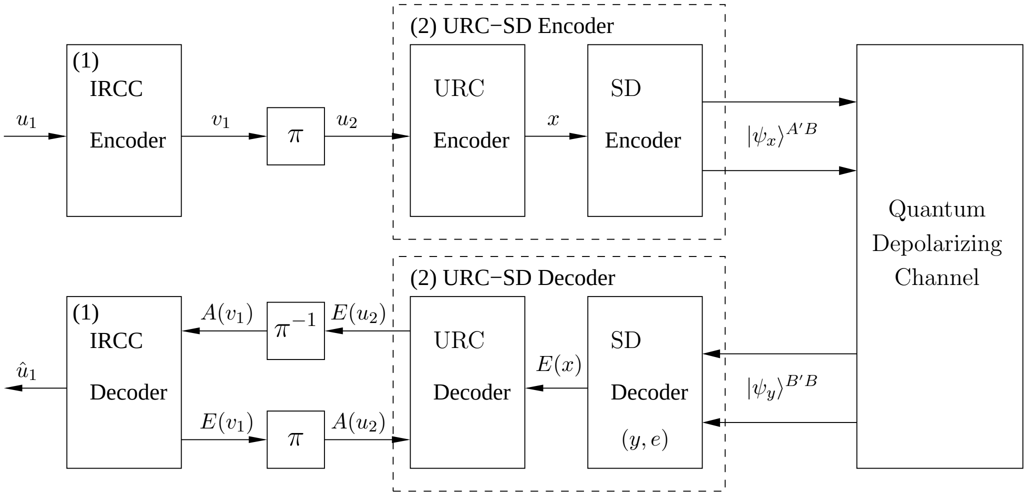


Figure 3.4: Schematic of the proposed IRCC-URC-SD classical-quantum communication system.

Section 3.1. Hence, the SD encoder has a function similar to that of the classical Phase-Shift Keying (PSK) or Quadrature Amplitude Modulation (QAM) bit-to-symbol mapper, which maps several classical bits onto a complex-valued phasor for communication using the classical electromagnetic waves. The qubits of the resultant quantum state are then serially transmitted over the quantum depolarizing channel[4].

At the receiver, iterative decoding is invoked for exchanging extrinsic information between the inner (URC-SD) and outer (IRCC) decoders. Here the notations $A(b)$ and $E(b)$ refer to the *a priori* and *extrinsic* probabilities of b, where we have $b \in \{v_1, u_2, x\}$, which are exploited for achieving decoding convergence to a vanishingly low Bit Error Rate (BER). The SD decoder converts the received orthogonal states $|\psi_y\rangle^{B'B}$ to classical symbols y by performing a joint measurement in the orthonormal basis. It must be highlighted here that a conventional SD decoder yields the hard-decision outputs. Instead, here we conceive a soft-decision SD decoder, which computes the corresponding extrinsic probability $E(x)$ for the transmitted classical symbol x, as follows:

$$E(x) \approx \mathrm{P}(y|x), \tag{3.20}$$

where $\mathrm{P}(y|x)$ is given by Eq. (3.15) and (3.18) for the 2-qubit and 3-qubit schemes, respectively. The soft output $E(x)$ is then fed into the Maximum A-Posteriori (MAP) decoder of URC, which engages in iterative decoding with the IRCC decoder.

[4] As illustrated earlier in Figure 3.1, the processed qubit(s) A' is transmitted over the noisy quantum channel, while B is shared between Alice and Bob over a noiseless channel.

3.5 Near-Capacity Design

3.5.1 EXIT Charts

EXIT charts [116, 68, 117] are capable of visualizing the convergence behaviour of iterative decoding schemes by exploiting the input/output relations of the constituent decoders in terms of their average Mutual Information (MI) transfer characteristics. In the context of our proposed model of Figure 3.4, the EXIT chart visualizes the exchange of the following four MI terms:

1. average *a priori* MI between u_2 and $A(u_2)$: $I_{A(u_2)}$,
2. average *a priori* MI between v_1 and $A(v_1)$: $I_{A(v_1)}$,
3. average *extrinsic* MI between u_2 and $E(u_2)$: $I_{E(u_2)}$, and
4. average *extrinsic* MI between v_1 and $E(v_1)$: $I_{E(v_1)}$.

Here, $I_{A(u_2)}$ and $I_{E(u_2)}$ constitute the EXIT curve of the inner decoder, while $I_{A(v_1)}$ and $I_{E(v_1)}$ yield the EXIT curve of the outer decoder. For the sake of constructing the inner and outer EXIT curves, the Log Likelihood Ratios (LLRs)[5] related to the *a priori* probabilities of $A(u_2)$ and $A(v_1)$ respectively, are modeled using a Gaussian distribution, having a mean of $\sigma_A^2/2$ and a variance of σ_A^2, for a range of $I_{A(u_2)}, I_{A(v_1)} \in [0, 1]$. The corresponding average *extrinsic* MI can be formulated as [114, 115]:

$$I_{E(u_2)} = \log_2 M + \mathrm{E}\left[\sum_{m=0}^{M-1} E(u_2^{(m)}) \log_2(E(u_2^{(m)}))\right], \tag{3.21}$$

and

$$I_{E(v_1)} = \log_2 M + \mathrm{E}\left[\sum_{m=0}^{M-1} E(v_1^{(m)}) \log_2(E(v_1^{(m)}))\right]. \tag{3.22}$$

Furthermore, since we are employing symbol-to-bit conversion at the URC decoder, we incorporate binary EXIT charts in our design. This in turn implies that in Eq. (3.21) and (3.22) we have $M = 2$ and $m \in \{0, 1\}$. The resultant inner EXIT function T_{u_2} is given by:

$$I_{E(u_2)} = T_{u_2}[I_{A(u_2)}, p], \tag{3.23}$$

while the outer EXIT function T_{v_1} is as follows:

$$I_{E(v_1)} = T_{v_1}[I_{A(v_1)}]. \tag{3.24}$$

[5] The LLR $L(x)$ of a bit x is the log of the ratio of the probabilities of the bit taking the two possible values of 0 and 1, which is given by[68]:

$$L(x) = \ln\left(\frac{\mathrm{P}(x=1)}{\mathrm{P}(x=0)}\right).$$

More explicitly, unlike T_{v_1}, T_{u_2} is a function of the depolarizing probability p, since the inner decoder is fed by the channel. Finally, the MI transfer characteristics of both the decoders encapsulated by Eq. (3.23) and (3.24) are plotted in the same graph, with the x and y axes of the outer decoder swapped. The resultant EXIT chart is capable of visualizing the exchange of extrinsic MI as a staircase-shaped decoding trajectory, as the iterations proceed. Examples of EXIT charts will be given in Section 3.6.

3.5.2 Near-Capacity IRCC-URC-SD Design

We have exploited the area property of EXIT charts [121] for designing a near-capacity classical error correction code for our classical-quantum communication system of Figure 3.4. According to this property, the area under the normalized EXIT curve of the inner decoder is approximately equal to the attainable channel capacity [121], provided that the channel's input symbols are equiprobable. Since our system model of Figure 3.4 transmits classical information over a quantum depolarizing channel, the attainable channel capacity of the system is the entanglement-assisted classical capacity given in Eq. (3.14). However, as mentioned in Section 3.4, symbol-to-bit conversion takes place at the output of the URC decoder. This incurs a capacity loss [122]. More explicitly, the corresponding bit-based capacity of 2SD may be computed by marginalizing the symbol-based channel transition probabilities $\mathrm{P}(y|x)$ of Eq. (3.15) to the bit-based probabilities $\mathrm{P}(y_i|x_i)$ for $i \in \{1, 2\}$, assuming that the constituent bits are independent. More specifically, we get:

$$\begin{aligned}\mathrm{P}(y_i = x_i|x_i) &= 1 - \frac{2}{3}p,\\ \mathrm{P}(y_i \neq x_i|x_i) &= \frac{2}{3}p.\end{aligned} \tag{3.25}$$

Based on this marginalized perspective, the resultant 4-ary classical channel may be viewed as a pair of independent Binary Symmetric Channels (BSCs) having a crossover probability of $2p/3$. The capacity of each BSC is given by:

$$\mathrm{C}_{\mathrm{BSC}}^{i} = 1 + (1 - \frac{2}{3}p)\log_2(1 - \frac{2}{3}p) + \frac{2}{3}p\log_2(\frac{2}{3}p), \tag{3.26}$$

for $i \in \{1, 2\}$. Since 2 classical bits are transmitted per channel use in the 2SD scheme, the symbol-based capacity of Eq. (3.17) is reduced to the sum of the capacity of these two BSCs, which is equivalent to:

$$\begin{aligned}\mathrm{C}_{2sd}^{\mathrm{bit}} &= \sum_{i=1}^{2} \mathrm{C}_{\mathrm{BSC}}^{i}\\ &= 2 \cdot \left[1 + (1 - \frac{2}{3}p)\log_2(1 - \frac{2}{3}p) + \frac{2}{3}p\log_2(\frac{2}{3}p)\right],\end{aligned} \tag{3.27}$$

where $\mathrm{C}_{\mathrm{BSC}}^{i}$ is the capacity of the ith BSC given in Eq. (3.26). More explicitly, the generalized formula of the bit-based capacity of an NSD scheme relying on a single noiseless pre-shared entangled qubit is given by:

$$\mathrm{C}_{Nsd}^{\mathrm{bit}} = \frac{1}{N-1} \cdot \sum_{i=1}^{N} \mathrm{C}_{\mathrm{BSC}}^{i}, \tag{3.28}$$

where $N = 2$ for the 2SD scheme.

Similarly, for computing the bit-based EACC of 3SD, the induced 8-ary channel may be viewed as three independent BSCs. For the sake of computing the channel transition probabilities associated with each of the three BSCs, we normalize the symbol-based conditional probability of Eq. (3.18) for each of the three constituent bits as follows:

$$\begin{aligned}
\mathrm{P}(y_1 = x_1|x_1) &= \sum_{e_1=0} \mathrm{P}(y = x + e|x = x^{(m)}) = 1 - \frac{4}{3}p + \frac{8}{9}p^2, \\
\mathrm{P}(y_1 \neq x_1|x_1) &= \sum_{e_1=1} \mathrm{P}(y = x + e|x = x^{(m)}) = \frac{4}{3}p - \frac{8}{9}p^2, \\
\mathrm{P}(y_2 = x_2|x_2) &= \sum_{e_2=0} \mathrm{P}(y = x + e|x = x^{(m)}) = 1 - \frac{4}{3}p + \frac{8}{9}p^2, \\
\mathrm{P}(y_2 \neq x_2|x_2) &= \sum_{e_2=1} \mathrm{P}(y = x + e|x = x^{(m)}) = \frac{4}{3}p - \frac{8}{9}p^2, \\
\mathrm{P}(y_3 = x_3|x_3) &= \sum_{e_3=0} \mathrm{P}(y = x + e|x = x^{(m)}) = 1 - \frac{2}{3}p, \\
\mathrm{P}(y_3 \neq x_3|x_3) &= \sum_{e_3=1} \mathrm{P}(y = x + e|x = x^{(m)}) = \frac{2}{3}p.
\end{aligned} \tag{3.29}$$

Using Eq. (3.13) as well as (3.29) and exploiting the fact that 3 classical bits are transmitted per 2 channel uses in 3SD, the symbol-based capacity of Eq. (3.19) is reduced to:

$$\begin{aligned}
\mathrm{C}_{3sd}^{\mathrm{bit}} &= \frac{1}{2} \cdot \sum_{i=1}^{3} \mathrm{C}_{\mathrm{BSC}}^{i} \\
&= \frac{1}{2} \cdot [3 - 2 \times \mathrm{H}_2(\frac{4}{3}p - \frac{8}{9}p^2) - \mathrm{H}_2(\frac{2}{3}p)],
\end{aligned} \tag{3.30}$$

where $\mathrm{C}_{\mathrm{BSC}}^{i}$ is the capacity of the ith BSC, while $\mathrm{H}_2(z)$ is the binary entropy function, which is given by,

$$\mathrm{H}_2(z) = -z \log_2(z) - (1 - z) \log_2(1 - z). \tag{3.31}$$

The capacity loss for both the 2SD and 3SD schemes is quantified in Figure 3.5, which compares their bit-based and symbol-based capacities. Nevertheless, it must be pointed out that by virtue of being a unity rate code, the URC does not impose any capacity loss, as verified in Figure 3.5. The capacity of our inner decoder (URC-SD) is approximately equal to the attainable bit-based entanglement-assisted classical capacity for both 2-qubit and 3-qubit superdense codes. The URC is only invoked for transforming the horizontal EXIT curve of the SD decoder to a slanted one for the sake of improving the scheme's decoding convergence, as detailed in the next section.

Furthermore, the area under the normalized EXIT curve of the outer decoder is equivalent to $(1 - R_o)$, where R_o is its coding rate [121]. Therefore, our near-capacity design aims for creating a

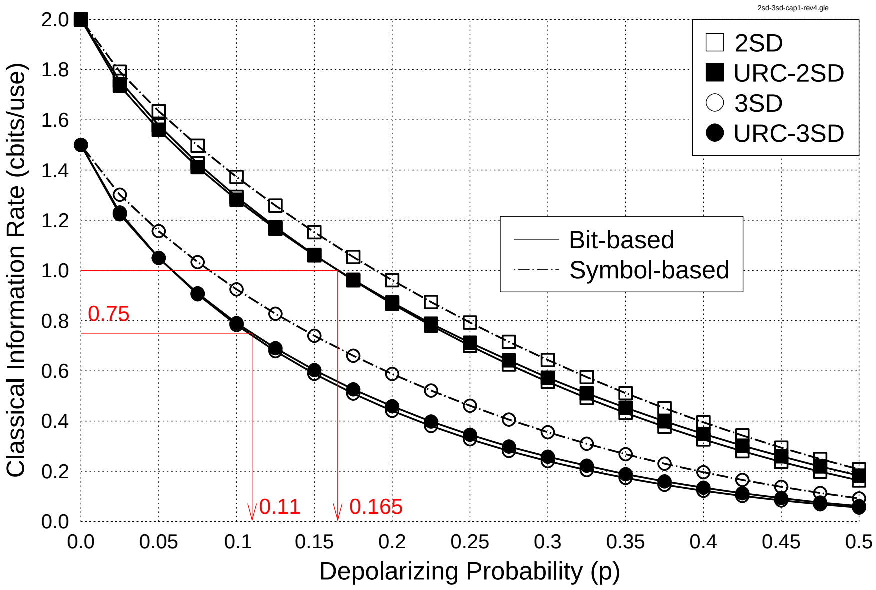


Figure 3.5: Classical information rate (cbits/use) versus quantum depolarizing probability for 2-qubit and 3-qubit superdense codes with and without URC. Symbol-to-bit conversion incurs a capacity loss for 2SD as well as 3SD.

narrow, but marginally open tunnel between the EXIT curves of the inner and outer decoders at the highest possible depolarizing probability, which corresponds to the lowest possible SNR for a classical channel. A feasible design option could be to create the EXIT curves of all the possible convolutional codes to find the optimal code $\mathcal{C}$, which gives the best match, i.e. whose EXIT curve yields a marginally open tunnel with the inner decoder's EXIT curve of URC-SD. To circumvent this tedious task, we have invoked the IRCC of [119], whereby a family of subcodes $\mathcal{C}_l$, $l \in \{1, 2, \ldots, L\}$, is used for constructing the target code $\mathcal{C}$. Due to its inherent flexibility, the resultant IRCC provides a better match than any single code. Furthermore, for the sake of reducing the encoding and decoding complexity, the family of subcodes $\mathcal{C}_l$ is constructed by selecting an r_i-rate convolutional code $\mathcal{C}_i$ as the mother code and obtaining the remaining $(L-1)$ subcodes $\mathcal{C}_l$ by puncturing the mother code for rate $r_l > r_1$ and by adding more generators and subsequently puncturing for $r_l < r_1$. The l^{th} subcode has a coding rate of r_l and it encodes a specifically designed fraction, ϱ_l, of the original information bits to $\varrho_l N_c$ encoded bits. Here, N_c is the total length of the coded frame. More specifically, for an L-subcode IRCC, ϱ_l is the l^{th} IRCC weighting coefficient satisfying the following constraints [118, 119]:

$$\sum_{l=1}^{L} \varrho_l = 1 \ , \ R_o = \sum_{l=1}^{L} \varrho_l r_l \ , \ \varrho_l \in [0,1], \forall l \ , \tag{3.32}$$

which can be conveniently represented in the following matrix form:

$$\begin{aligned} \begin{bmatrix} 1 & 1 & \ldots & 1 \\ r_1 & r_2 & \ldots & r_L \end{bmatrix} \begin{bmatrix} \varrho_1 & \varrho_2 \ldots & \varrho_L \end{bmatrix}^T &= \begin{bmatrix} 1 \\ R_o \end{bmatrix} \\ \mathbf{C} \ \boldsymbol{\varrho} &= \mathbf{d} \ . \end{aligned} \tag{3.33}$$

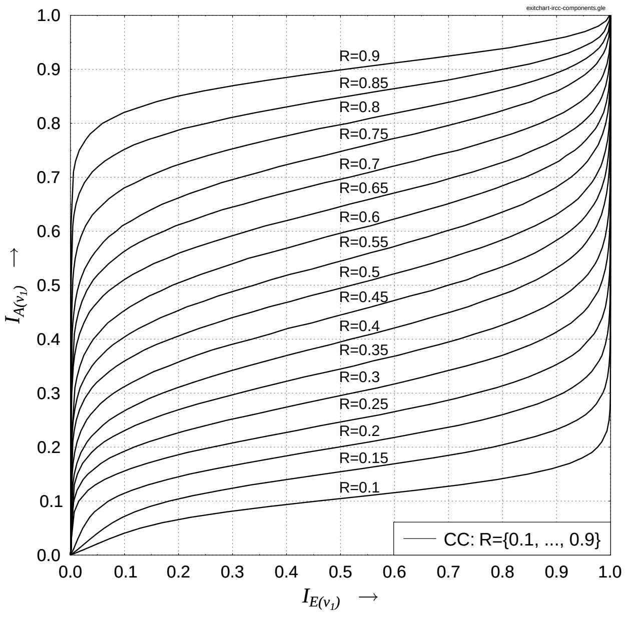


Figure 3.6: Normalized outer EXIT curves (inverted) of the 17 IRCC subcodes.

In our design, we have employed an IRCC relying on a set of 17 memory-4 convolutional subcodes having 17 different coding rates between 0 and 1, which was found in [119]. These 17 subcodes are derived such that it covers the complete range of coding rates from 0.1 to 0.9 with a rate-increment of 0.05, i.e. having rates of $r_l \in \{0.1, 0.15, 0.2, \ldots, 0.85, 0.9\}$. Figure 3.6 shows the inverted outer EXIT curves for each of the constituent subcode of the IRCC scheme.

In physically tangible terms, the input bit stream is divided into 17 fractions corresponding to the 17 different-rate subcodes and the specific optimum fractions to be encoded by these codes are found by dynamic programming. More specifically, the EXIT curves of the 17 subcodes, given in Figure 3.6, are superimposed onto each other after weighting by the appropriate fraction-based weighting coefficients, which are determined by minimizing the area of the open EXIT-tunnel. To elaborate a little further, the transfer function of the IRCC is given by:

$$I_{E(v_1)} = T_{v_1}\left[I_{A(v_1)}\right] = \sum_{l=1}^{L} \varrho_l \; T_{v_1,l}\left[I_{A(v_1)}\right] \;, \tag{3.34}$$

where $T_{v_1,l}\left[I_{A(v_1)}\right] = I_{E(v_1),l}$ is the transfer function of the l^{th} subcode. We employed the curve matching algorithm of [118, 119] for optimizing the weighting coefficients of the IRCC subcodes by ensuring that a narrow, yet open tunnel exists between the EXIT curves of the outer and inner decoder at the highest possible depolarizing probability; thus, guaranteeing that the system has a near-capacity performance.

SD scheme	2-qubit
IRCC coding rate	1/2
IRCC active subcodes	$\varrho_4 = 0.0177,\ \varrho_5 = 0.0145,\ \varrho_7 = 0.6455,\ \varrho_{12} = 0.1797,$ $\varrho_{13} = 0.0580,\ \varrho_{16} = 0.0105,\ \varrho_{17} = 0.0742$
Interleaver length	$30,000$ bits
Overall system rate	1 cbits/use

Table 3.4: Simulation parameters of the IRCC-URC-SD scheme of Figure 3.4.

3.6 Results and Discussions I

Based on the near-capacity design of Section 3.5, we have designed an SD-based near-capacity code for entanglement-assisted classical communication over the quantum depolarizing channel. We next evaluate the performance of our 2-qubit and 3-qubit designs in Section 3.6.1 and 3.6.2, respectively.

3.6.1 Performance of IRCC-URC-2SD

Since we intend to design a system having a rate of 1 cbit/use, we have assumed a constant overall coding rate of 0.5 for the IRCC. Figure 3.7 shows the normalized EXIT curves for 2SD at a depolarizing probability of 0.15 and using an interleaver length of $30,000$ bits. As expected, the EXIT curve of the 2SD decoder is a horizontal straight line. Hence, our URC is used as a precoder, to transform this horizontal EXIT curve into a slanted curve which terminates at the $(1, 1)$ point of the EXIT chart; thus, facilitating a possible convergence to an infinitesimally low BER. More specifically, the area under the EXIT curve remains the same, yet reaches the $(1, 1)$ point. Furthermore, using the curve matching algorithm of [118, 119], the IRCC weight vector of Eq. (3.33) was optimized to get a narrow open tunnel as evident in Figure 3.7. The corresponding simulation parameters are summarized in Table 3.4, where only seven subcodes are activated. The tunnel of Figure 3.7 is narrow, but wide enough for successful convergence, as visualized using the decoding trajectories. If the depolarizing probability is increased beyond $p = 0.15$, the EXIT curves of the inner and outer decoder would crossover, hence closing the tunnel. Thus, the system has a convergence threshold of $p = 0.15$. In other words, it can tolerate depolarizing probabilities upto $p = 0.15$, and yet achieve an infinitesimally low BER. However, this would require a high number of iterations between the IRCC and URC-2SD, hence imposing a high complexity.

The coding rate of the designed IRCC-URC-2SD system is 1 cbit/use, since a 1/2-rate IRCC is used. From the bit-based capacity curve of Figure 3.5, it can be found that the associated noise limit is $p^* = 0.165$. By contrast, the convergence threshold of our system is $p = 0.15$. Thus, it operates within

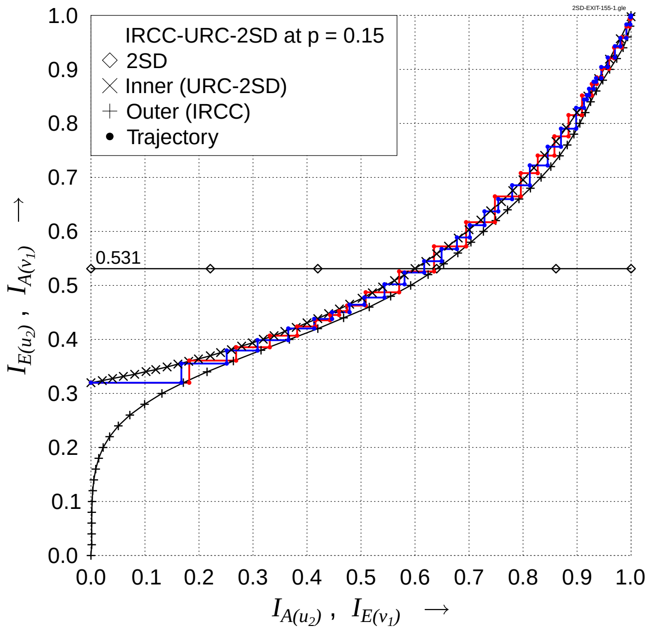


Figure 3.7: Normalized EXIT curves of the IRCC-URC-2SD system of Figure 3.4 at a depolarizing probability of 0.15 using the simulation parameters of Table 3.4.

$[10 \times \log_{10}(\frac{0.165}{0.15})] = 0.4$ dB of the noise limit[6]. Alternatively, this discrepancy may also be quantified in terms of the difference in the area under the inner and outer EXIT curves, which corresponds to the normalized capacity loss. The area under the normalized EXIT curve of our URC-2SD scheme is 0.531, whereas that under the IRCC is 0.5. Thus, the capacity of our IRCC-URC-2SD scheme is only $[0.031 \times 2] = 0.062$ cbits/use away from the capacity, when $p = 0.15$.

We have further evaluated the BER performance of our IRCC-URC-2SD scheme in Figure 3.8 for the simulation parameters of Table 3.4. As it can be observed in Figure 3.8, the performance improves upon increasing the number of iterations. More specifically, the 2-qubit system starts to converge to a lower BER, as the number of iterations increases at a depolarizing probability of $p = 0.15$, which matches the convergence thresholds predicted using EXIT charts. More explicitly, since the EXIT chart tunnel closes beyond the depolarizing probability threshold of $p = 0.15$, the system fails to converge, if the depolarizing probability is increased further. Hence, the performance does not improve upon increasing the number of iterations if the depolarizing probability exceeds the threshold. By contrast, when the depolarizing probability is below the threshold, the BER improves at each successive iteration. Here, the trade-off between the complexity imposed and the performance attained comes into play. It should also be noted that the performance improves with diminishing returns at a higher number of iterations. For example, doubling the number of iterations from $\mathtt{I} = 8$ to

[6] The difference in dB between two channel depolarizing probabilities p_1 and p_2 is calculated as follows [95, 88]: $\left(10 \times \log_{10} \frac{p_1}{p_2}\right)$.

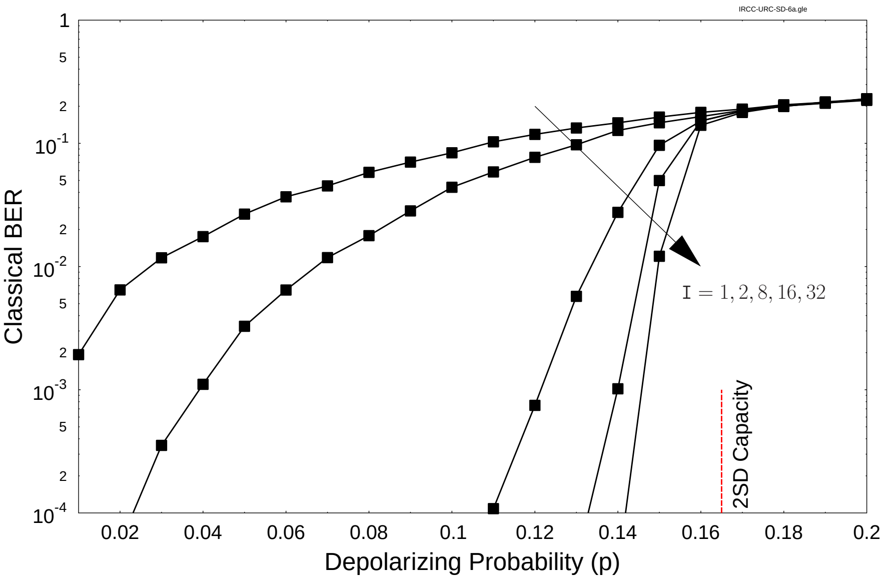


Figure 3.8: Achievable BER performance of the IRCC-URC-2SD scheme of Figure 3.4 upon increasing the number of iterations, i.e. $\mathtt{I} = \{1, 2, 8, 16, 32\}$. The simulation parameters are summarized in Table 3.4. The dashed-line at $p^* = 0.165$ marks the noise limit for a classical information rate of 1 cbit/use, which is obtained from the bit-based EACC curve of 2SD given in Figure 3.5.

$\mathtt{I} = 16$ for IRCC-URC-2SD increases the tolerable depolarizing probability by 0.0225, corresponding to a BER of 10^{-4}. A further increase to $\mathtt{I} = 32$ iterations only improves p by around 0.01 at a BER of 10^{-4}. We further demonstrate this in Figure 3.9, where we quantify the distance from the capacity, i.e. from the noise limit of $p^* = 0.165$, in terms of dB at a BER of 10^{-4} upon increasing the number of iterations. We may observe in Figure 3.9 that we approach the achievable noise limit with diminishing returns, as the number of iterations is increased.

To elaborate further on the significance of using an IRCC rather than a conventional 1/2-rate CC, we have also conceived a corresponding setup, whereby the IRCC of Figure 3.4 is replaced by a memory-4 1/2-rate CC in the proposed IRCC-URC-2SD system. This is synonymous to employing an IRCC, which has only the 9th subcode active. Figure 3.10 shows the resultant EXIT curves for $p = 0.15$ and $p = 0.125$. It can be observed in Figure 3.10 that for $p = 0.15$, which is the convergence threshold of our IRCC-URC-2SD design, the inner and outer EXIT curves of the CC-URC-2SD scheme exhibit a cross-over. Thus, implying that the CC-URC-2SD configuration fails to converge at $p = 0.15$. An open tunnel emerges only when p is decreased to 0.125. Consequently, the convergence threshold of CC-URC-2SD is $p = 0.125$, which is lower than that of our near-capacity design of Figure 3.7. It must also be pointed out here that the area between the inner and outer EXIT curves at the convergence threshold is wider than Figure 3.7. The wider the gap, the higher the capacity loss. Therefore, using a regular CC, rather than an IRCC, yields a poor match between the inner and outer decoders' EXIT curves.

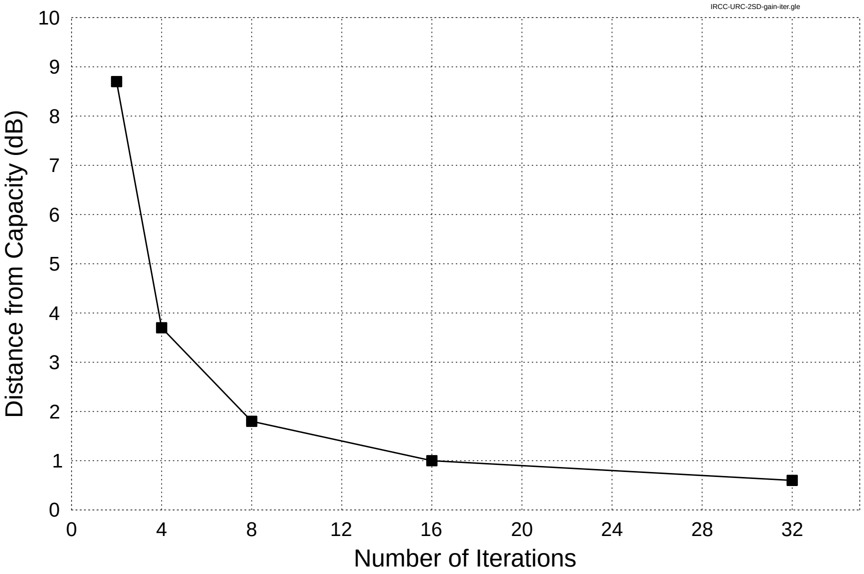


Figure 3.9: Performance of the IRCC-URC-2SD scheme of Figure 3.8, at a BER of 10^{-4}, which is quantified in terms of the distance from the bit-based EACC of 2SD, i.e. $p^* = 0.165$, as the number of iterations is increased.

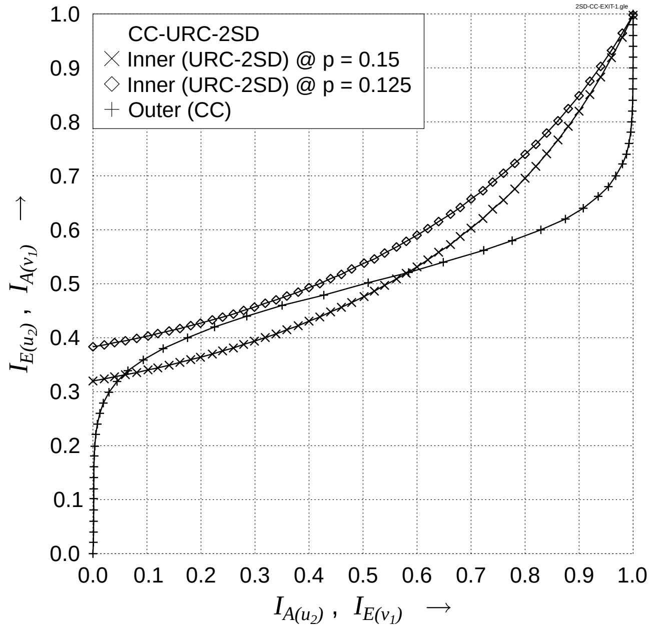


Figure 3.10: Normalized EXIT curves of the CC-URC-2SD system using the simulation parameters of Table 3.4, but only the 9th subcode of IRCC is active.

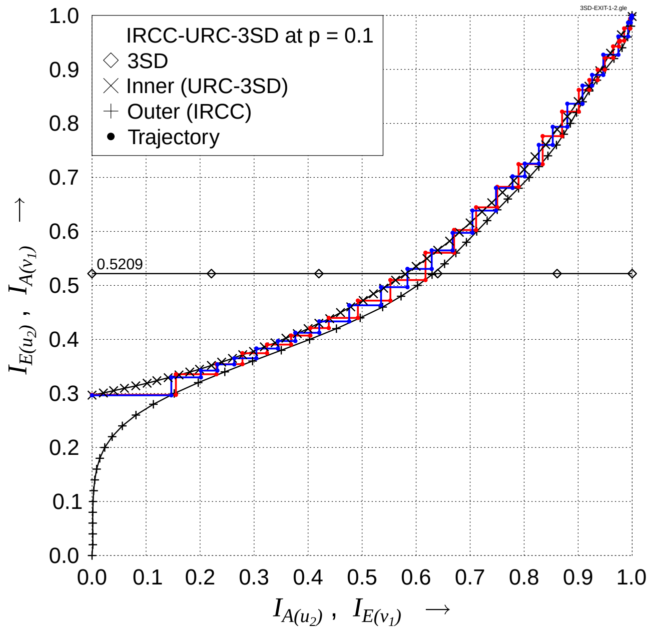


Figure 3.11: Normalized EXIT curves of the IRCC-URC-3SD system at a depolarizing probability of 0.1 using the simulation parameters of Table 3.5.

3.6.2 Performance of IRCC-URC-3SD

As another example, we designed an IRCC-URC-3SD scheme having a classical transmission rate of 0.75 cbits/use. We have therefore assume a coding rate of 0.5 for the IRCC. Figure 3.11 shows the EXIT curves for our 3-qubit SD at a depolarizing probability of 0.1. The optimized IRCC weights are given in Table 3.5, where only five subcodes are activated. For $p \leq 0.1$, the system successfully converges and the decoding trajectory terminates at the $(1, 1)$ point of the EXIT chart. Since our 3SD transmits 1.5 cbits/use and we have used 1/2-rate IRCC, the effective throughput of the designed system is 0.75 cbits/use. The corresponding depolarizing probability according to the bit-based capacity curve of Figure 3.5 is $p^* = 0.11$. Thus, in terms of the depolarizing probability, our designed system operates within $[10 \times \log_{10}(\frac{0.11}{0.10})] = 0.4$ dB of the capacity. Furthermore, the area under the normalized EXIT curve of the inner decoder is 0.5209. The deviation from the capacity curve is therefore $[0.0209 \times 1.5] \approx 0.031$ cbits/use.

We have further evaluated the corresponding BER performance in Figure 3.12 for the simulation parameters of Table 3.5. Analogous to our IRCC-URC-2SD scheme in Figure 3.8, the performance in Figure 3.12 improves upon increasing the number of iterations. Particularly, the system converges for $p \leq 0.1$, which conforms to our EXIT chart predictions of Figure 3.11. Furthermore, for $p \leq 0.1$, the performance tends to approach the noise limit of $p^* = 0.11$. This is also demonstrated in Figure 3.13, which plots the distance from the capacity (dB) at a BER of 10^{-4} as a function of the number of iterations. As observed previously for our 2SD scheme, the performance converges towards the noise

SD scheme	3-qubit
IRCC coding rate	1/2
IRCC active subcodes	$\varrho_6 = 0.2641,\ \varrho_7 = 0.4062,\ \varrho_{12} = 0.1068,$ $\varrho_{13} = 0.1247,\ \varrho_{17} = 0.0982$
Interleaver length	$30,000$ bits
Overall system rate	0.75 cbits/use

Table 3.5: Simulation parameters of the IRCC-URC-SD scheme of Figure 3.4.

limit, as the number of iterations increases, but this happens with diminishing returns at higher number of iterations.

We have further benchmarked the performance of our IRCC-URC-SD system of Figure 3.4 against the classical Turbo Code (TC) in Figure 3.14 for both 2SD as well as 3SD designs. This was achieved by replacing the IRCC-URC unit of Figure 3.4 with TC[7]. We have used a memory-3 1/2-rate TC for our comparison, since it invokes 16 states in each iteration, which is the same as the number of states invoked per iteration in our design[8]. The uncoded BER curves of our 2SD and 3SD schemes are also plotted in Figure 3.14. Furthermore, we have used a sufficiently high number of iterations, i.e. $\mathtt{I} = 32$, for our designed system to ensure that the system reaches the top right corner of the EXIT chart at a depolarizing probability that is close to the noise limit. More specifically, as observed in Figure 3.9 and Figure 3.13 for the 2SD and 3SD schemes, respectively, doubling the number of iterations from 16 to 32 improves the performance only slightly. Therefore, we can safely assume that if the increasing the number of iterations may not improve the performance appreciably. By contrast, $\mathtt{I} = 16$ iterations were used for TC since it did not yield any appreciable performance improvement, when the number of iterations was increased beyond $\mathtt{I} = 8$, as evidenced in Figure 3.14. Our proposed IRCC-URC-SD system is capable of performing closer to the capacity, hence, outperforming the turbo code for both 2SD and 3SD. The corresponding distances from the capacity expressed in terms of dB at a BER of 10^{-4} are tabulated in Table 3.6, where the noise limits for 2SD and 3SD are $p^* = 0.165$ and $p^*0.11$, respectively.

[7] Symbol-to-bit conversion takes place at the output of SD decoder. Consequently, the symbol-based probabilities of Eq. (3.15) and (3.18) are converted to bit-based LLRs, assuming that the bits constituting the symbol are independent.

[8] Since a memory-3 turbo code has two components with 2^3 states, total number of states per iteration are $2 \times 2^3 = 16$. Similarly, a memory-4 IRCC invokes $2^4 = 16$ states per iteration.

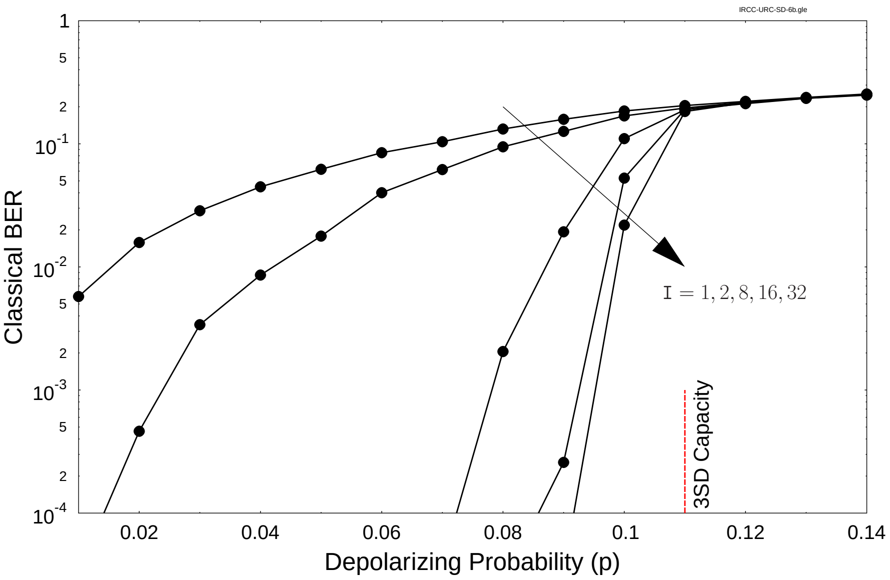


Figure 3.12: Achievable BER performance of the IRCC-URC-3SD scheme of Figure 3.4 with increasing number of iterations, i.e. $\mathtt{I} = \{1, 2, 8, 16, 32\}$. The simulation parameters are summarized in Table 3.5. The dashed-line at $p^* = 0.11$ marks the noise limit for a classical information rate of 0.75 cbit/use, which is obtained from the bit-based EACC curve of 3SD given in Figure 3.5.

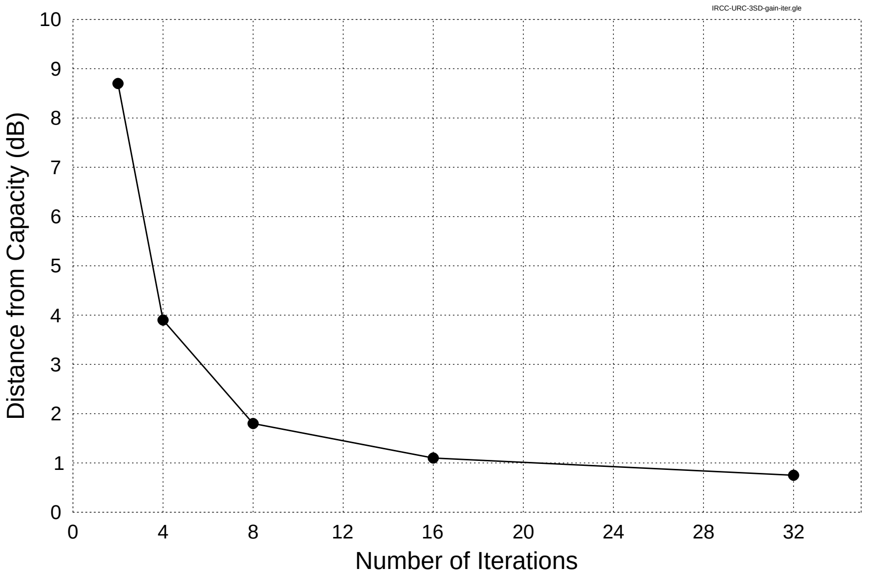


Figure 3.13: Performance of the IRCC-URC-3SD scheme of Figure 3.12 at a BER of 10^{-4}, which is quantified in terms of the distance from the bit-based EACC of 3SD, i.e. $p^* = 0.11$, as the number of iterations is increased.

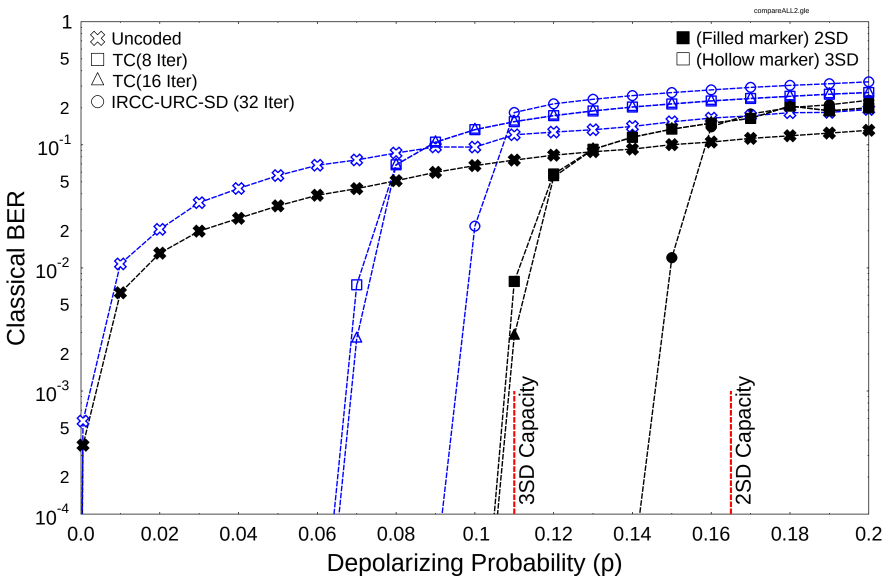


Figure 3.14: Comparison of the achievable BER performance of our IRCC-URC-SD scheme with TC-SD having a memory-3 1/2-rate TC as the outer component. 2SD and 3SD schemes are plotted with filled and hollow markers, respectively, and their simulation parameters are summarized in Table 3.4 and Table 3.5, respectively. Uncoded BER curves for both 2SD as well as 3SD are also plotted for comparison. Results are summarized in Table 3.6.

3.7 Symbol-Based Code Structure

Since the design of Figure 3.4 uses a bit interleaver and hence bit-based iterative decoding, symbol-to-bit conversion is invoked before the related soft-information is fed from the inner decoder (URC-SD) to the outer decoder (IRCC). This in turn incurs a capacity loss. As gleaned from Figure 3.5, for a 2SD scheme having a classical information rate of 1 cbit/use, a bit-based system ensures reliable transmission for $p \leq 0.165$, while a symbol-based system would increase the noise limit to $p^* = 0.1875$. Therefore, a bit-based error correction scheme incurs a capacity loss of around 0.6 dB as compared to its symbol-based counterpart. This capacity loss was previously identified in [122] for classical discrete-memoryless channels and a modified binary LDPC code was proposed to circumvent this

	TC-SD	**IRCC-URC-SD**
2SD	1.9 dB	0.6 dB
3SD	2.2 dB	0.75 dB

Table 3.6: Distance of the TC-SD and IRCC-URC-SD schemes from the capacity at a BER of 10^{-4} using the performance curves of Figure 3.14.

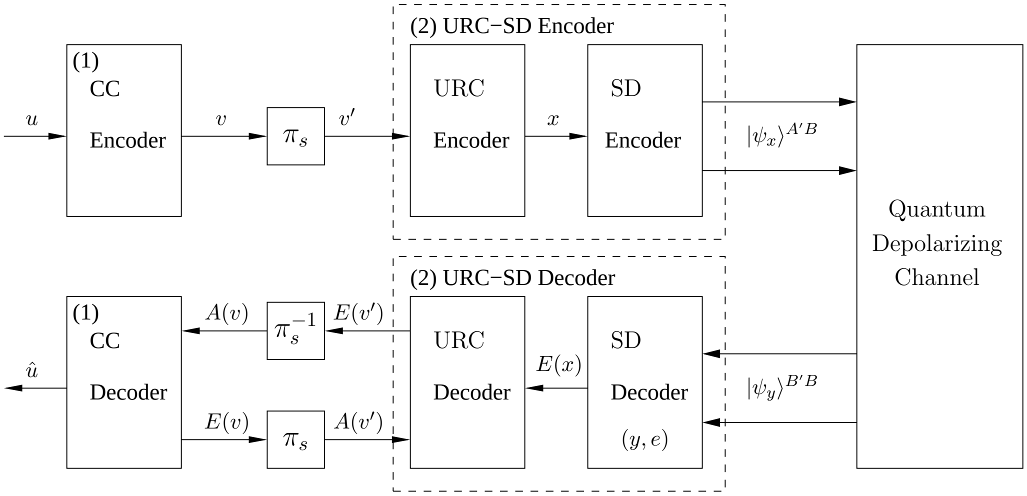


Figure 3.15: Schematic of the proposed symbol-based CC-URC-SD classical-quantum communication system.

issue. Similarly, to circumvent the quantum channel capacity loss, we have conceived an iterative code design for symbol-based CC-URC-2SD, which incorporates a single CC as the outer component, while the URC and 2SD schemes constitute the amalgamated inner code.

Figure 3.15 shows the proposed system model. At the transmitter, the system is fed with classical bits $\{u\}$, which are encoded by a 1/2-rate CC. The encoded 4-ary coded symbols $v = (v_1\ v_2)$ are then interleaved by a symbol interleaver (π_s), yielding the permuted symbol stream v', which is fed to the symbol-based recursive URC having a generator polynomial of $G(D) = \frac{1}{1+D}$ [68]. Similar to the bit-based design of Figure 3.4, classical to quantum domain conversion then takes place at the SD encoder of Figure 3.15 and the encoded qubits $|\psi_x\rangle^{A'B}$ are serially transmitted over the quantum depolarizing channel. The receiver of Figure 3.15 is also same as that of Figure 3.4 with the bit interleaver replaced by a symbol interleaver.

Since our proposed model of Figure 3.15 relies on symbol-based iterative decoding, we invoke non-binary EXIT charts of [114, 68, 117] for the sake of achieving a near-capacity performance.

3.8 Results and Discussions II

Using the non-binary EXIT-charts, we have optimized our iterative code structure of Figure 3.15 to design a system with a coding rate of 1 cbit/use. According to the symbol-based capacity curve of Figure 3.5, the corresponding noise limit for our system is $p^* = 0.1875$.

Design Objective I: *For the sake of comparing the symbol-based scheme of Figure 3.15 with the bit-based scheme of Figure 3.4, which uses a* 1/2*-rate memory-*4 *IRCC, find the optimal* 1/2*-rate memory-*4 *convolutional code, which gives the best match with URC-2SD in the CC-URC-2SD*

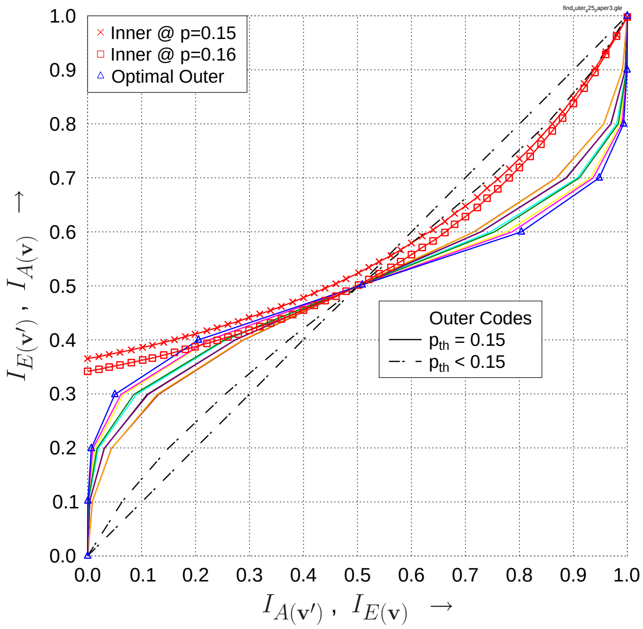


Figure 3.16: Normalized EXIT curves of the CC-URC-2SD system. Various 1/2-rate memory-4 convolutional codes were used as outer components. The optimal outer code has generator polynomials $(g_1, g_2) = (31, 36)_8$.

configuration, when symbol-based iterative decoding is invoked.

For the sake of achieving this objective, we created the EXIT curves of all the possible 1/2-rate memory-4 convolutional codes by evaluating all legitimate generator polynomials to find the optimal code $\mathcal{C}$, which yields a marginally open tunnel at the highest possible channel depolarizing probability. The EXIT characteristics of some of these $(2, 1, 4)$ CCs are plotted in Figure 3.16 along with the inner decoder EXIT curve of the URC-2SD scheme at $p = 0.15$ and $p = 0.16$. As gleaned from the figure, all outer decoder EXIT curves plotted in 'solid' lines exhibit a convergence threshold of $p_{\text{th}} = 0.15$, i.e. a marginally open tunnel exists for $p = 0.15$. If the depolarizing probability is increased beyond 0.15, the inner and outer decoder EXIT curves will crossover, thereby closing the tunnel. By contrast, the pair of outer decoder EXIT curves plotted in 'dashed' lines have $p_{\text{th}} < 0.15$. Hence, our desired optimal code $\mathcal{C}$ is one of those associated with $p_{\text{th}} = 0.15$. It may be further observed in Figure 3.16 that the EXIT curve labeled as 'Optimal Outer', whose octally represented generator polynomials are $(g_1, g_2) = (31, 36)_8$, converges faster than the others[9]. Therefore, we have selected it as our optimal outer component.

The BER performance of the optimal CC of Figure 3.16 is recorded in Figure 3.17 using the simulation parameters of Figure 3.7. As it can be observed, the turbo-cliff formulation starts around

[9] The optimal outer code yields the widest area between the inner and outer EXIT curves after the $(0.5, 0.5)$-point. This signifies that fewer decoding iterations are required.

SD scheme	2-qubit
Interleaver length	$30,000$ bits
Overall system rate	1 cbits/use
Convolutional Code	
Coding rate	1/2
Memory	4
(g_1, g_2)	$(31, 36)_8$

Table 3.7: Simulation parameters of the CC-URC-2SD scheme of Figure 3.15.

$p = 0.15$, which matches the convergence threshold predicted using EXIT charts. More specifically, at $p \leq 0.15$, the system converges to a low BER as the number of iterations increases, while for $p \geq 0.16$, the performance fails to improve upon increasing the number of iterations. This is because, as shown in Figure 3.16, the EXIT chart tunnel closes at $p = 0.16$. Thus, the system fails to converge to a low BER for $p \geq 0.16$. It may also be observed that the performance only moderately improves with diminishing returns at higher number of iterations. Furthermore, since doubling the number of iterations from $\mathtt{I} = 10$ to $\mathtt{I} = 20$ only improves the performance slightly at a BER of 10^{-4}, we may conclude that $\mathtt{I} = 20$ iterations are sufficient to approach the $(1, 1)$-point of near-perfect convergence. This is also demonstrated in Figure 3.18, where the distance from the capacity in dBs is plotted at a BER of 10^{-4} for the increasing number of iterations. We may observe in Figure 3.18 that there is only a negligible improvement in performance, when the number of iterations is increased from 15 to 20.

We further compare our symbol-based CC-URC-2SD to the bit-based IRCC-URC-2SD of Figure 3.4 in Figure 3.19, where the uncoded BER of 2SD is also plotted. It may be observed that both systems have the same convergence threshold of $p = 0.15$, which is within $[10 \times \log_{10}(\frac{0.15}{0.1875})] = 1$ dB of the achievable noise limit. Since an IRCC has a higher encoding and decoding structural complexity than a single-component CC, we can achieve the same convergence threshold at a lower encoding/decoding structural complexity using the symbol-based scheme. Furthermore, the CC-URC-2SD system exhibits an improved BER performance compared to the IRCC-URC-2SD scheme, as shown in Figure 3.19. After $\mathtt{I} = 2$ iterations, the IRCC-URC-2SD arrangement yields a BER of 10^{-4} at $p = 0.0225$, while the CC-URC-SD scheme has a BER of 10^{-4} at $p = 0.0525$. Therefore, CC-URC-2SD outperforms the IRCC-URC-2SD arrangement by $[10 \times \log_{10}(\frac{0.0225}{0.0525})] = 3.7$ dB. Moreover, as demonstrated in Figure 3.8, the IRCC-URC-2SD scheme achieves perfect convergence after about $\mathtt{I} = 32$ iterations, while only $\mathtt{I} = 20$ iterations are sufficient for the symbol-based CC-URC-2SD. We further benchmark the performance against the achievable symbol-based capacity of $p^* = 0.1875$. At a BER of 10^{-4} and after a sufficiently high number of iterations ($\mathtt{I} = 20$ for CC-URC-2SD and $\mathtt{I} = 32$ for IRCC-URC-2SD), the CC-URC-2SD scheme operates within $[10 \times \log_{10}(\frac{0.149}{0.1875})] = 1$ dB of the capacity, while

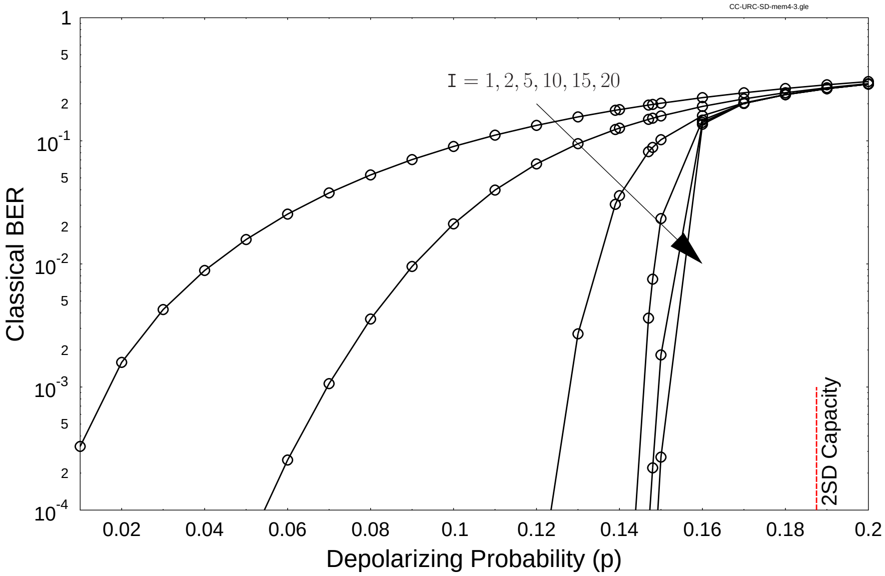


Figure 3.17: Achievable BER performance of the CC-URC-2SD scheme of Figure 3.15 with increasing number of iterations, i.e. $\mathtt{I} = \{1, 2, 5, 10, 15, 20\}$. Simulation parameters are summarized in Table 3.7. The dashed-line at $p^* = 0.1875$ marks the noise limit for a classical information rate of 1 cbit/use, which is obtained from the symbol-based EACC curve of 2SD given in Figure 3.5.

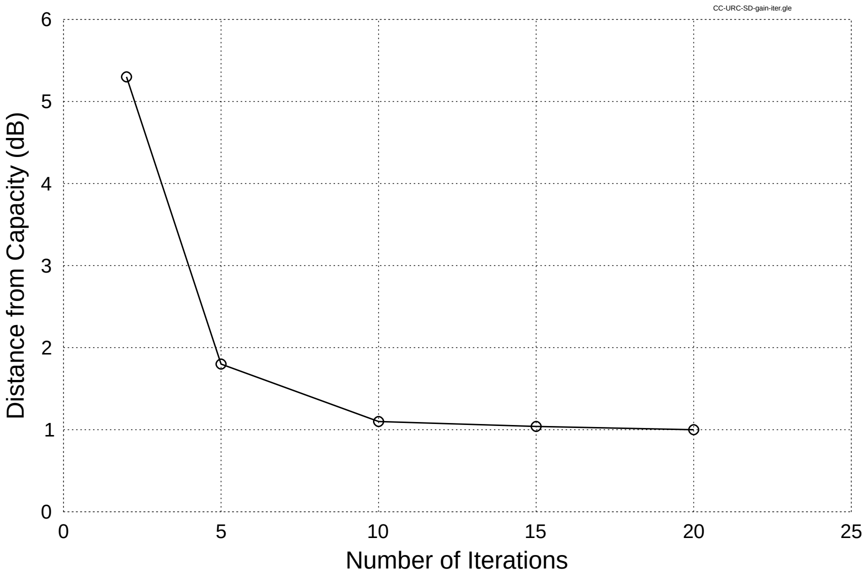


Figure 3.18: Performance of the CC-URC-2SD scheme of Figure 3.17, at a BER of 10^{-4}, which is quantified in terms of the distance from the symbol-based EACC of 2SD, i.e. $p^* = 0.1875$, as the number of iterations is increased.

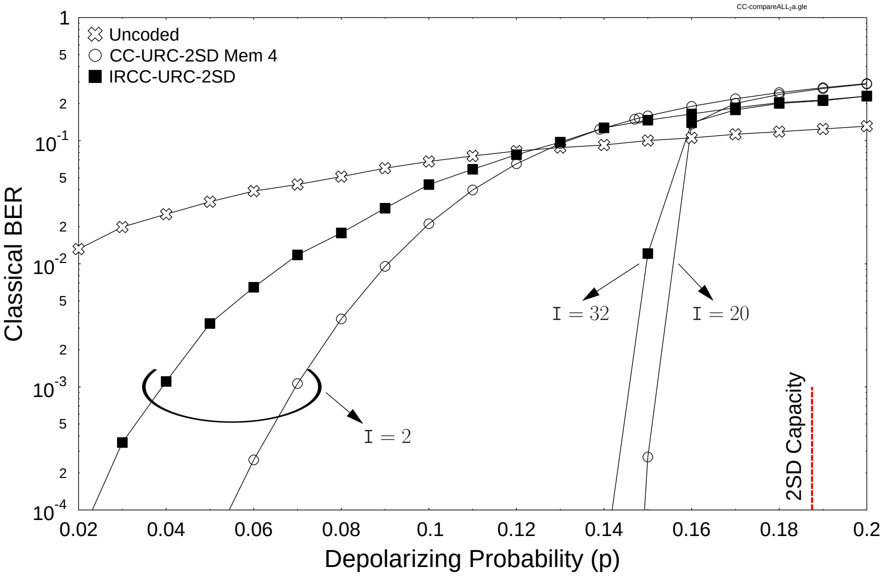


Figure 3.19: Comparison of the achievable BER performance of the bit-based IRCC-URC-2SD of Figure 3.4 and the symbol-based CC-URC-2SD design of Figure 3.15 using the simulation parameters of Table 3.4 and Table 3.7, respectively.

the IRCC-URC-2SD regime exhibits a deviation of $[10 \times \log_{10}(\frac{0.142}{0.1875})] = 1.2$ dB from the capacity. Thus, the performance of both systems becomes comparable, once perfect convergence is achieved. However, the IRCC-URC-2SD scheme requires 60% more iterations than the symbol-based CC-URC-SD arrangement.

Design Objective II: *Find the optimal* $1/2$*-rate memory-*2 *and memory-*3 *convolutional codes, which exhibit the best EXIT-curve shape match with URC-2SD in the CC-URC-2SD configuration, when symbol-based iterative decoding is invoked.*

Again, for the sake of finding the optimal memory-2 and memory-3 outer components, we created the EXIT curves for all the possible codes, as we previously did in Figure 3.16. It was found that the $\text{CC}(2,1,2)$ having the generators $(g_1, g_2) = (7,5)_8$ and the $\text{CC}(2,1,3)$ with generators $(g_1, g_2) = (17,15)_8$ yield the best match. The corresponding EXIT curves for the optimized memory-2 and memory-3 CCs are plotted in Figure 3.20 together with the optimal memory-4 CC of Figure 3.16. All codes have the same decoding convergence threshold. The corresponding BER performance is compared in Figure 3.21 after both 2 and 20 iterations. The CC associated with a higher constraint length exhibits a lower BER before perfect convergence is achieved, e.g. after 2 iterations as shown in Figure 3.21. Furthermore, after 20 iterations, all codes have a similar performance at a BER of 10^{-4}. Codes having a lower constraint length have the additional benefit of a lower decoding complexity, since fewer states are invoked per iteration. We have further compared the optimized symbol-based CC-URC-2SD designs for varying constraint lengths to the bit-based IRCC-URC-2SD in Table 3.8 by quantifying their performance at a BER of 10^{-4} in terms of the distance (dB) from the noise limit

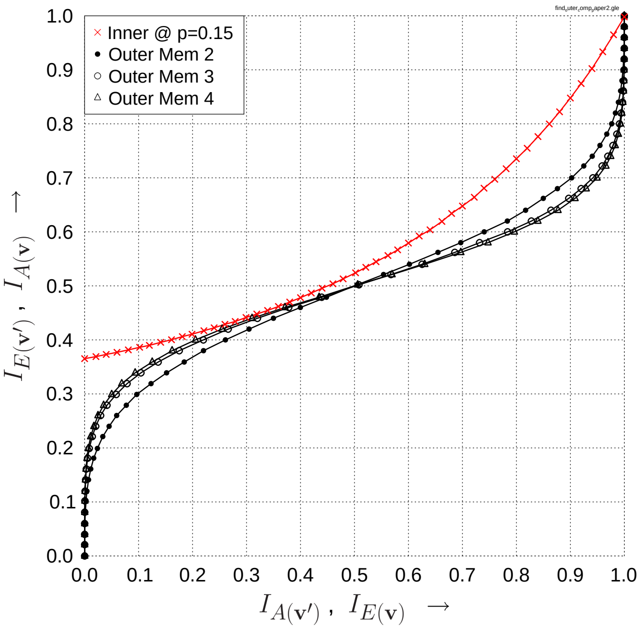


Figure 3.20: Normalized EXIT curves of the CC-URC-2SD system optimized for varying constraint lengths. Optimal outer components are plotted here: CC$(2,1,2)$ with $(g_1, g_2) = (7,5)_8$, CC$(2,1,3)$ with $(g_1, g_2) = (17,15)_8$ and CC$(2,1,4)$ with $(g_1, g_2) = (31,36)_8$.

of $p^* = 0.1875$. All the three symbol-based configurations outperform the bit-based IRCC-URC-SD scheme.

3.9 Summary and Conclusions

In this chapter, we have conceived both bit-based as well as symbol-based concatenated classical-quantum code structures for entanglement-assisted classical communication over a quantum depolarizing channel. We commenced with a review of the SD protocol in Section 3.2, which facilitates the transmission of N classical bits by sending only $(N-1)$ qubits over the noisy quantum channel, while one qubit is pre-shared with the receiver. This results in a transmission rate of 2 cbits/use for the 2SD protocol of Section 3.2.1 and a rate of $\frac{N}{N-1}$ cbits/use for the general NSD scheme of Section 3.2.2. We then derived the EACC for the general NSD transmission in Section 3.3, which was also customized for the 2SD and 3SD schemes. In Section 3.4, we presented our proposed bit-based IRCC-URC-SD system of Figure 3.4, which relies on channel coding operating in the classical domain by serially concatenating an IRCC and a URC aided SD encoder. Furthermore, we have introduced a soft-decision aided superdense decoder facilitating iterative decoding. More specifically, the URC and SD constitute the amalgamated inner component, while the IRCC constitutes the outer component. Therefore, iterative decoding is invoked for exchanging extrinsic information between the inner (URC-SD) and outer

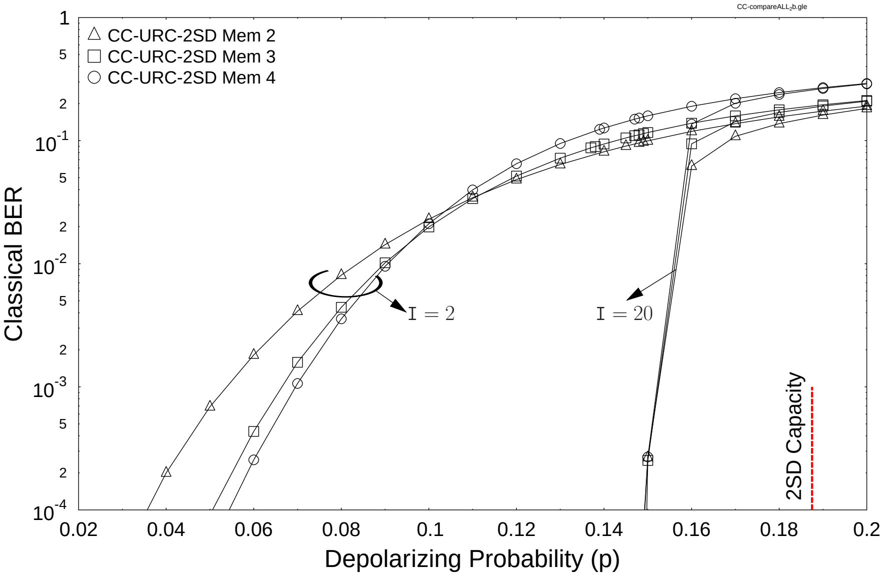


Figure 3.21: Comparison of the achievable BER performance of the symbol-based CC-URC-2SD design of Figure 3.15 optimized for varying constraint lengths. Optimal CCs are: CC(2, 1, 2) with $(g_1, g_2) = (7, 5)_8$, CC(2, 1, 3) with $(g_1, g_2) = (17, 15)_8$ and CC(2, 1, 4) with $(g_1, g_2) = (31, 36)_8$. Other simulation parameters are same as Table 3.7.

Outer Code	$\mathtt{I} = 2$	$\mathtt{I} \to \infty$
CC(2, 1, 2)	7.2 dB	1 dB
CC(2, 1, 3)	5.7 dB	1 dB
CC(2, 1, 4)	5.5 dB	1 dB
IRCC	9.2 dB	1.2 dB

Table 3.8: Comparison of the symbol-based CC-URC-2SD schemes having the optimized CC(2, 1, 2), CC(2, 1, 3) and CC(2, 1, 4) of Figure 3.21 to the bit-based IRCC-URC-2SD design of Figure 3.19 quantified in terms of the distance from the capacity (noise limit $p^* = 0.1875$) at a BER of 10^{-4}. Here '∞' denotes 'sufficiently high' number of iterations for ensuring near-perfect convergence, which is assumed to be $\mathtt{I} = 20$ for CC-URC-2SD and $\mathtt{I} = 32$ for IRCC-URC-2SD.

(IRCC) decoders. Furthermore, we presented our EXIT-chart aided near-capacity design criterion in Section 3.5 and demonstrated how the IRCC weighting coefficients have to be optimized for ensuring that a marginally open tunnel exits between the inner and outer decoders' EXIT curves at the highest possible depolarizing probability.

We then evaluated the performance of our bit-based IRCC-URC-SD design for 2SD and 3SD in Section 3.6.1 and 3.6.2, respectively, which was benchmarked against the bit-based EACC given in Figure 3.5. It was demonstrated that our BER performance curves of Figure 3.8 and Figure 3.12 conform to the EXIT chart predictions of Figure 3.7 and Figure 3.11, respectively. Furthermore, the proposed system of Figure 3.4 operates within 0.4 dB of the achievable noise limit for both 2SD as well as 3SD schemes. More specifically, our design exhibits a deviation of only 0.062 and 0.031 cbits/use from the corresponding 2-qubit and 3-qubit capacity limits, respectively. We also benchmarked our system against the classical convolutional and turbo codes in Figure 3.10 and Figure 3.14, respectively. It was shown in Figure 3.14 that the TC-2SD scheme operates within 1.9 dB of the capacity at a BER of 10^{-4}, while the performance of our bit-based IRCC-URC-2SD is only 0.6 dB from the capacity. Similarly, the TC-3SD scheme is 2.5 dB from the capacity at a BER of 10^{-4} in contrast to the bit-based IRCC-URC-3SD, which operates within 0.75 dB.

Our bit-based code structure of Figure 3.4 incurs a capacity loss due to the symbol-to-bit conversion, as quantified in Figure 3.5. To overcome this capacity loss, we conceived a symbol-based code design in Section 3.7, which employs a single-component CC and a symbol interleaver in contrast to the IRCC and bit interleaver of Figure 3.4. We optimized our symbol-based CC-URC-2SD design with the aid of non-binary EXIT charts in Figure 3.16. Our simulation results of Section 3.8 demonstrated that the symbol-based CC-URC-2SD provides a significant BER performance improvement, despite its lower encoding/decoding complexity than that of the bit-based IRCC-URC-2SD. Quantitatively, after 2 iterations, our proposed symbol-based CC-URC-2SD design incorporating a memory-4 CC outperformed the bit-based IRCC-URC-2SD scheme by 3.7 dB at a BER of 10^{-4}, as evidenced in Figure 3.19. Furthermore, the bit-based IRCC-URC-2SD arrangement required around 60% more iterations than the symbol-based CC-URC-2SD for achieving perfect decoding convergence. We also demonstrated in Figure 3.21 that the decoding complexity can be further reduced by using memory-2 and memory-3 CCs, which rely on only 4 and 8 states, respectively, per iteration. It was found in Figure 3.21 that even the memory-2 and memory-3 designs outperform the bit-based IRCC-URC-2SD. Finally, the performances of our bit-based IRCC-URC-SD ($\mathtt{I} = 32$) as well as the symbol-based CC-URC-SD ($\mathtt{I} = 20$) at a BER of 10^{-4} are summarized in Figure 3.22, along with the TC-SD ($\mathtt{I} = 16$) benchmark. To dispense with the exhaustive search, it may be helpful to conceive a symbol-based IRCC, whose weighting coefficients can be dynamically adapted to provide the best EXIT-curve match with that of a given inner code, as also discussed in Section 8.2.

To conclude, in this chapter, we exploited classical redundancy for the reliable transmission of classical information over a quantum channel. Consequently, this design approach is only appropriate when the information to be transmitted is classical. For more general quantum communication systems, which may transmit classical as well as quantum information, and for quantum computation

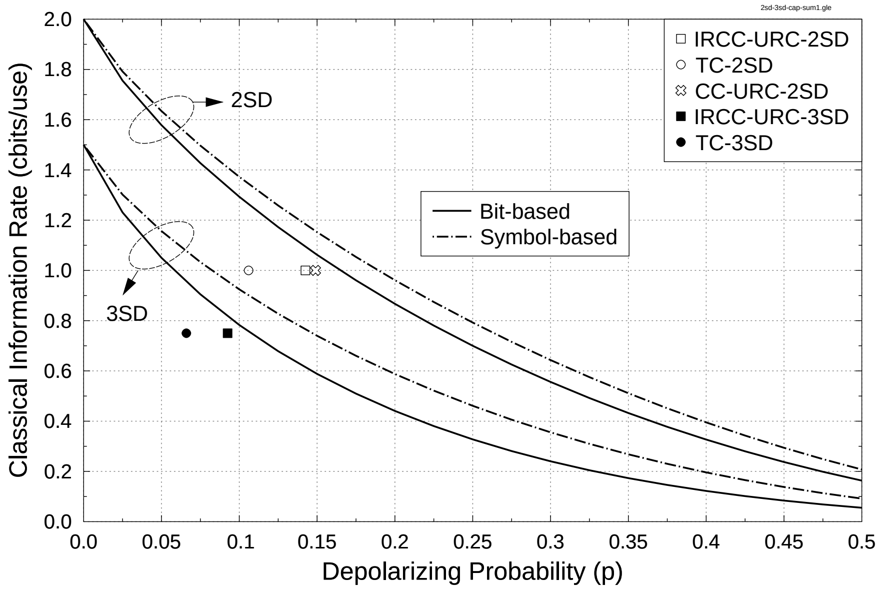


Figure 3.22: Classical information rate (cbits/use) versus the quantum depolarizing probability for bit-based and symbol-based 2SD as well as 3SD schemes. The performances of our designed bit-based IRCC-URC-SD ($\mathtt{I} = 32$) and symbol-based CC-URC-SD ($\mathtt{I} = 20$), along with the TC-SD ($\mathtt{I} = 16$) benchmark, are compared at a BER of 10^{-4}.

systems, it is vital to invoke quantum error correction codes, hence exploiting the redundancy in the quantum domain. In this spirit, we detail the design principles for constructing quantum codes from the known classical codes in the next chapter.

Chapter 4

From Classical to Quantum Error Correction

4.1 Introduction

In Chapter 3, we invoked near-capacity classical code designs for classical transmission over a quantum communication channel. Our classical-quantum code design, which amalgamates the classical codes with a quantum superdense code, is only suitable when the transmitted information is classical. For reliable quantum information transmission as well as for quantum computing systems, we have to resort to the family of Quantum Error Correction Codes (QECCs), which exploit redundancy in the quantum domain in contrast to the classical redundancy of Chapter 3. Meritorious families of QECCs can be derived from the known classical codes by exploiting the underlying quantum-to-classical isomorphism, while also taking into account the peculiar laws of quantum mechanics. This transition from the classical to the quantum domain has to address the following challenges [18]:

- **No-Cloning Theorem:** Most classical codes are based on the transmission of multiple replicas of the same bit, e.g. in a simple rate-1/3 repetition code each information bit is transmitted thrice. This is not possible in the quantum domain according to the no-cloning theorem [99], discussed in Section 2.4, which states that an arbitrary unknown quantum state cannot be copied/cloned.
- **Continuous Nature of Quantum Errors:** In contrast to the classical errors, which are discrete with bit-flip being the only type of error, recall from Section 2.8 that a qubit may experience both a bit-flip as well as a phase-flip or in fact both. These impairments have a continuous nature and the erroneous qubit may lie anywhere on the surface of the Bloch sphere of Figure 2.2.
- **Qubits Collapse upon Measurement:** 'Measurement' of the received bits is a vital step representing a hard-decision operation in the field of classical error correction, but this is not feasible in the quantum domain, since qubits collapse to classical bits upon measurement.

In a nutshell, a classical (n, k) binary code is designed to protect discrete-valued message sequences of length k by encoding them into one of the 2^k discrete codewords of length n. By contrast, since a quantum state of k qubits is specified by 2^k continuous-valued complex coefficients, quantum error correction aims for encoding a k-qubit state into an n-qubit state, so that all the 2^k complex coefficients can be perfectly restored [47]. For example, let $k = 2$, then the 2-qubit information word $|\psi\rangle$ is given by:

$$|\psi\rangle = \alpha_0|00\rangle + \alpha_1|01\rangle + \alpha_2|10\rangle + \alpha_3|11\rangle. \tag{4.1}$$

Consequently, the error correction algorithm would aim for correctly preserving all the four coefficients, i.e. α_0, α_1, α_2 and α_3. It is interesting to note here that although the coefficients α_0, α_1, α_2 and α_3 are continuous in nature, yet the entire continuum of errors can be corrected, if we can correct a discrete set of errors, i.e. bit (Pauli-**X**), phase (Pauli-**Z**) as well as both (Pauli-**Y**) errors inflicted on either or both qubits [18]. This is because the act of measurement collapses the entire continuum of errors to a discrete set. More explicitly, for $|\psi\rangle$ of Eq. (4.1), the discrete error set is as follows:

$$\begin{aligned}\{&\mathbf{IX}, \mathbf{IZ}, \mathbf{IY}, \mathbf{XI}, \mathbf{XX}, \mathbf{XZ}, \mathbf{XY}, \mathbf{ZI}, \mathbf{ZX}, \mathbf{ZZ},\\ &\mathbf{ZY}, \mathbf{YI}, \mathbf{YX}, \mathbf{YZ}, \mathbf{YY}\}.\end{aligned} \tag{4.2}$$

However, the errors **X**, **Y** and **Z** may occur with varying frequencies. In this discourse, we will focus on the specific design of codes conceived for mitigating the deleterious effects of the quantum depolarizing channel, which has been extensively investigated in the context of QECCs [47, 54, 56].

Most of the quantum codes developed to date owe their existence to the theory of stabilizer codes, which allows us to import any arbitrary classical binary as well as quaternary code to the quantum domain. In this chapter, we will delve deeper into the stabilizer formalism, giving insights into the construction of quantum codes from the known classical codes. Since the stabilizer codes owe their existence to the classical linear block codes, we commence with a review of the classical linear block codes in Section 4.2. Stabilizer codes are then introduced in Section 4.3, with a particular emphasis on the general stabilizer formalism, on the underlying quantum-to-classical isomorphism and on the classification of quantum errors, which are discussed in Sections 4.3.1, 4.3.2 and 4.3.3, respectively. We next detail the construction of quantum convolutional codes in Section 4.4, while the entanglement-assisted quantum codes are discussed in Section 4.5. Finally, the highlights of the chapter are summarized in Section 4.6.

4.2 Review of Classical Linear Block Codes

The stabilizer formalism derives its existence from the theory of classical linear block codes. A classical linear block code $C(n, k)$ maps k-bit information blocks onto n-bit codewords. For small values of k and n, this can be readily achieved using a look-up table, which maps the input information blocks onto the encoded message blocks. However, for large values of k and n, the process may be simplified using a $k \times n$ generator matrix G as follows:

$$\overline{x} = xG, \tag{4.3}$$

where x and $\overline{x}$ are row vectors for information and encoded messages, respectively. Furthermore, G may be decomposed as:

$$G = (I_k|P), \tag{4.4}$$

where I_k is a $(k \times k)$-element identity matrix and P is a $k \times (n-k)$-element matrix. This in turn implies that the first k bits of the encoded message are information bits, followed by $(n-k)$ parity bits.

At the decoder, syndrome decoding is invoked, which determines the position of the channel-induced error using the observed syndromes rather than directly acting on the received codewords. More precisely, each generator matrix is associated with an $(n-k) \times n$-element Parity Check Matrix (PCM) H which is given by:

$$H = \left(P^T|I_{n-k}\right), \tag{4.5}$$

and is defined such that $\overline{x}$ is a valid codeword only if,

$$\overline{x}H^T = 0. \tag{4.6}$$

For a received vector $y = \overline{x} + e$, where e is the error incurred during transmission, the error syndrome of length $(n-k)$ is computed as:

$$s = yH^T = (\overline{x} + e)H^T = \overline{x}H^T + eH^T = eH^T, \tag{4.7}$$

which is then used for identifying the erroneous bit.

Let us consider a simple 3-bit repetition code, which makes three copies of the intended information bit. More precisely, $k = 1$ and $n = 3$. It is specified by the following generator matrix:

$$G = \begin{pmatrix} 1 & 1 & 1 \end{pmatrix}, \tag{4.8}$$

which yields two possible codewords [111] and [000]. At the receiver, a decision may be made on the basis of the majority voting by reading the received bits. For example, if $y = [011]$ is received, then we may conclude that the transmitted bit was 1. Alternatively, we may invoke the PCM-based syndrome decoding. According to Eq. (4.5), the corresponding PCM is given by:

$$H = \begin{pmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \end{pmatrix}. \tag{4.9}$$

It can be worked out that $yH^T = 0$ only for the two valid codewords [111] and [000]. For all other received codewords, at least one of the two syndrome elements is set to 1, e.g. when the first bit is corrupted, i.e. $y = [011]$ or $[100]$, $s = [11]$. The Look-Up Table (LUT) for the PCM of Eq. (4.9) is given in Table 4.1, which enlists all the 1-bit errors that may be identified using this syndrome decoding procedure.

Syndrome (s)	**Index of Error**
[11]	1
[10]	2
[01]	3

Table 4.1: Look-up table for the PCM of Eq. (4.9), which enlists the single-bit errors along with the corresponding syndromes.

This process of error correction using the generator and parity check matrices is usually preferred due to its compact nature. Generally, $C(n,k)$ code, which encodes a k-bit information message into an n-bit codeword, would require 2^k n-bit codewords. Thus, it would require a total of $n2^k$ bits to completely specify the code space. By contrast, the aforementioned approach only requires kn bits of the generator matrix. Hence, memory resources are saved exponentially, while encoding and decoding operations are efficiently implemented. These attractive features of classical block linear codes and the associated PCM-based syndrome decoding have led to the development of quantum stabilizer codes.

4.3 Quantum Stabilizer Codes

4.3.1 Stabilizer Formalism

Let us recall from Section 2.2 that qubits collapse to classical bits upon measurement [18]. This prevents us from directly applying the classical error correction techniques for reliable quantum transmission. Inspired by the PCM-based syndrome decoding of classical codes, Gottesman [38, 39] introduced the notion of stabilizer formalism, which facilitates the design of quantum codes from the classical ones. Analogous to Shor's pioneering 9-qubit code [25], stabilizer formalism circumvents the measurement issue by observing the error syndromes without reading the actual quantum information. More specifically, Quantum Stabilizer Codes (QSCs) invoke the syndrome decoding approach of classical linear block codes for estimating the errors incurred during transmission.

Figure 4.1 shows the general schematic of a quantum communication system relying on a QSC for reliable transmission. An $[n,k]$ QSC[1] encodes the information qubits $|\psi\rangle$ into the coded sequence $|\overline{\psi}\rangle$ with the aid of $(n-k)$ auxiliary (also called ancilla) qubits, which are initialized to the state $|0\rangle$. The noisy sequence $|\hat{\psi}\rangle = \mathcal{P}|\overline{\psi}\rangle$, where $\mathcal{P}$ is the n-qubit channel error, is received at the receiver (RX), which engages in a 3-step process for the sake of recovering the intended transmitted information. More explicitly, RX computes the syndrome of the received sequence $|\hat{\psi}\rangle$ and uses it to estimate the channel error $\tilde{\mathcal{P}}$ with the aid of the classical syndrome decoding. The recovery operator $\mathcal{R}$ then

[1] We use round brackets (.) for classical codes, while the square brackets [.] are used for quantum codes.

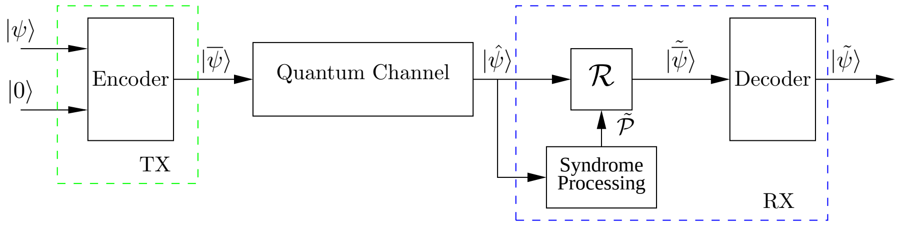


Figure 4.1: System Model: Quantum communication system relying on a quantum stabilizer code.

uses the estimated error $\tilde{\mathcal{P}}$ to restore the transmitted coded stream. Finally, the decoder, or more specifically the inverse encoder, processes the recovered coded sequence $|\tilde{\overline{\psi}}\rangle$, yielding the estimated transmitted information qubits $|\tilde{\psi}\rangle$.

An $[n, k]$ QSC, constructed over a code space $\mathcal{C}$, which maps the information word (logical qubits) $|\psi\rangle \in \mathbb{C}^{2^k}$ onto the codeword (physical qubits) $|\overline{\psi}\rangle \in \mathbb{C}^{2^n}$, where $\mathbb{C}^d$ denotes the d-dimensional Hilbert space, is defined by a set of $(n-k)$ independent commuting n-tuple Pauli operators g_i, for $1 \leq i \leq (n-k)$. The corresponding stabilizer group $\mathcal{H}$ contains both g_i and all the products of g_i for $1 \leq i \leq (n-k)$ and forms an Abelian subgroup of $\mathcal{G}_n$. A unique feature of these operators is that they do not change the state of valid codewords, while yielding an eigenvalue of -1 for corrupted states.

Let us now elaborate on this definition of the stabilizer code by considering a simple 3-qubit bit-flip repetition code, which is capable of correcting single-qubit bit-flip errors. Since the laws of quantum mechanics do not permit cloning of the information qubit, we cannot encode $|\psi\rangle$ to $(\psi \otimes \psi \otimes \psi)$. Instead, the 3-qubit bit-flip repetition code entangles two auxiliary qubits with the information qubit such that the basis states $|0\rangle$ and $|1\rangle$ are copied thrice in the superposition of basis states of the resulting 3-qubit codeword, i.e. $|0\rangle$ and $|1\rangle$ are mapped as follows:

$$\begin{aligned} |0\rangle &\rightarrow |\overline{0}\rangle \equiv |000\rangle, \\ |1\rangle &\rightarrow |\overline{1}\rangle \equiv |111\rangle. \end{aligned} \tag{4.10}$$

Consequently, the information word $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$ is encoded as:

$$\alpha|0\rangle + \beta|1\rangle \rightarrow \alpha|\overline{0}\rangle + \beta|\overline{1}\rangle \equiv \alpha|000\rangle + \beta|111\rangle, \tag{4.11}$$

using the quantum circuit of Figure 4.2. The resultant codeword $|\overline{\psi}\rangle$ is stabilized by the operators $g_1 = \mathbf{ZZI}$ and $g_2 = \mathbf{ZIZ}$. Here the term ‘stabilize’ implies that the valid codewords are not affected by the generators g_1 and g_2 and yield an eigenvalue of $+1$, as shown below:

$$\begin{aligned} g_1\left[|\overline{\psi}\rangle\right] &= \alpha|000\rangle + \beta|111\rangle \equiv |\overline{\psi}\rangle, \\ g_2\left[|\overline{\psi}\rangle\right] &= \alpha|000\rangle + \beta|111\rangle \equiv |\overline{\psi}\rangle. \end{aligned} \tag{4.12}$$

On the other hand, if a corrupted state $|\hat{\psi}\rangle$ is received, then the stabilizer generators yield an eigenvalue

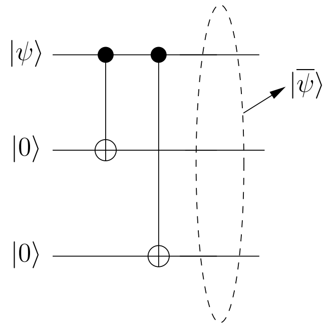


Figure 4.2: Circuit for 3-qubit bit-flip repetition code. The information qubit $|\psi\rangle$ is encoded into $|\overline{\psi}\rangle$ using two auxiliary qubits, which are entangled with $|\psi\rangle$ using CNOT gates.

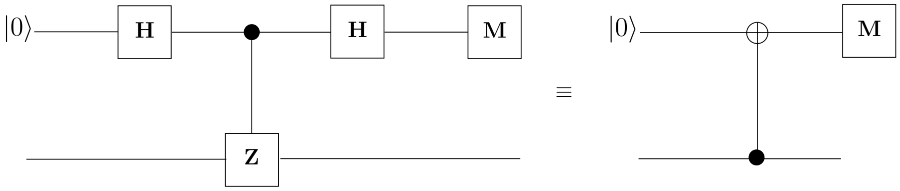


Figure 4.3: Quantum circuit for measuring the **Z** operator acting on the bottom qubit [18]. The top qubit is the auxiliary qubit used for computing syndrome. The circuit on the left is popular, while the one on the right is suitable for implementation.

of -1, e.g. let $|\hat{\psi}\rangle = |100\rangle + \beta|011\rangle$ where $\mathcal{P} = \mathbf{XII}$, then we have:

$$\begin{aligned} g_1\left[|\hat{\psi}\rangle\right] &= -\alpha|100\rangle - \beta|011\rangle \equiv -|\hat{\psi}\rangle, \\ g_2\left[|\hat{\psi}\rangle\right] &= -\alpha|100\rangle - \beta|011\rangle \equiv -|\hat{\psi}\rangle. \end{aligned} \tag{4.13}$$

More specifically, the stabilizer generators have a role similar to the PCM of a classical linear block code. The eigenvalue is -1 if the n-tuple Pauli error $\mathcal{P}$ acting on the transmitted codeword $|\overline{\psi}\rangle$ anti-commutes with the stabilizer g_i (analogous to $yH^T \neq 0$ in Eq. (4.7)) and it is $+1$ if $\mathcal{P}$ commutes with g_i (analogous to $yH^T = 0$ in Eq. (4.7)). Therefore, we have:

$$g_i|\hat{\psi}\rangle = \begin{cases} |\overline{\psi}\rangle, & g_i\mathcal{P} = \mathcal{P}g_i \\ -|\overline{\psi}\rangle, & g_i\mathcal{P} = -\mathcal{P}g_i, \end{cases} \tag{4.14}$$

where $|\hat{\psi}\rangle = \mathcal{P}|\overline{\psi}\rangle$. The resultant ± 1 eigenvalue gives the corresponding error syndrome s, which is 0 for an eigenvalue of $+1$ and 1 for an eigenvalue of -1. This computation of the syndrome using the **Z** operators is realized by employing the quantum circuit of Figure 4.3, where the circuit on the left is popular, while the one on the right is the equivalent circuit suitable for implementation [18]. In both circuits of Figure 4.3, the top qubit is the auxiliary qubit used for computing the syndrome, while the bottom qubit is the coded qubit subjected to the **Z** operator. The resultant syndromes are listed in

$\lvert\hat{\psi}\rangle = \mathcal{P}\lvert\overline{\psi}\rangle$	$g_1\lvert\hat{\psi}\rangle$	$g_2\lvert\hat{\psi}\rangle$	**Syndrome** (s)	**Index of Error**
$\alpha\lvert 100\rangle + \beta\lvert 011\rangle$	-1	-1	[11]	1
$\alpha\lvert 010\rangle + \beta\lvert 101\rangle$	-1	$+1$	[10]	2
$\alpha\lvert 001\rangle + \beta\lvert 110\rangle$	$+1$	-1	[01]	3

Table 4.2: Single-qubit bit-flip errors along with the corresponding eigenvalues for 3-qubit bit-flip repetition code having $g_1 = \mathbf{ZZI}$ and $g_2 = \mathbf{ZIZ}$.

Table 4.2 along with the corresponding single-qubit bit-flip errors, eigenvalues and the location of the single-bit errors, which may be identified using the syndrome decoding approach.

A 3-qubit phase-flip repetition code may be constructed using a similar approach. This is because phase errors in the Hadamard basis $\{\lvert+\rangle, \lvert-\rangle\}$ are similar to the bit errors in the computational basis $\{\lvert 0\rangle, \lvert 1\rangle\}$. More explicitly, the states $\lvert+\rangle$ and $\lvert-\rangle$ are defined as:

$$\begin{aligned} \lvert+\rangle &\equiv \mathbf{H}\lvert 0\rangle = \frac{\lvert 0\rangle + \lvert 1\rangle}{\sqrt{2}}, \\ \lvert-\rangle &\equiv \mathbf{H}\lvert 1\rangle = \frac{\lvert 0\rangle - \lvert 1\rangle}{\sqrt{2}}, \end{aligned} \tag{4.15}$$

where $\mathbf{H}$ is a single-qubit Hadamard gate, which is given by [18]:

$$\mathbf{H} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}. \tag{4.16}$$

Therefore, Pauli-$\mathbf{Z}$ acting on the states $\lvert+\rangle$ and $\lvert-\rangle$ yields:

$$\begin{aligned} \mathbf{Z}\lvert+\rangle &= \lvert-\rangle, \\ \mathbf{Z}\lvert-\rangle &= \lvert+\rangle, \end{aligned} \tag{4.17}$$

which is similar to the operation of Pauli-$\mathbf{X}$ on the states $\lvert 0\rangle$ and $\lvert 1\rangle$, i.e. we have:

$$\begin{aligned} \mathbf{X}\lvert 0\rangle &= \lvert 1\rangle, \\ \mathbf{X}\lvert 1\rangle &= \lvert 0\rangle. \end{aligned} \tag{4.18}$$

Consequently, analogous to Eq. (4.10), a 3-qubit phase-flip repetition code encodes $\lvert 0\rangle$ and $\lvert 1\rangle$ in the Hadamard basis as follows:

$$\begin{aligned} \lvert 0\rangle &\to \lvert\overline{0}\rangle \equiv \lvert + + +\rangle, \\ \lvert 1\rangle &\to \lvert\overline{1}\rangle \equiv \lvert - - -\rangle, \end{aligned} \tag{4.19}$$

for protection against the single-qubit phase errors. Based on Eq. (4.19), $\lvert\psi\rangle$ is encoded as:

$$\alpha\lvert 0\rangle + \beta\lvert 1\rangle \to \alpha\lvert\overline{0}\rangle + \beta\lvert\overline{1}\rangle \equiv \alpha\lvert + + +\rangle + \beta\lvert - - -\rangle, \tag{4.20}$$

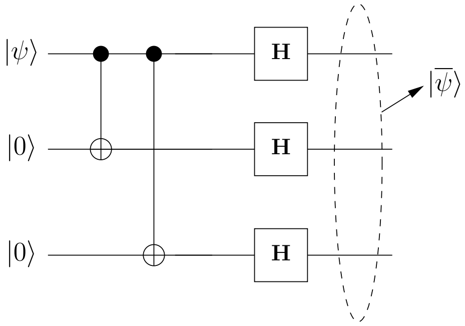


Figure 4.4: Circuit for 3-qubit phase-flip repetition code. The information qubit $|\psi\rangle$ is encoded into $|\overline{\psi}\rangle$ using two auxiliary qubits, which are entangled with $|\psi\rangle$ using CNOT gates. Finally, the **H**-gate transforms the computational basis into the Hadamard basis for phase-error correction.

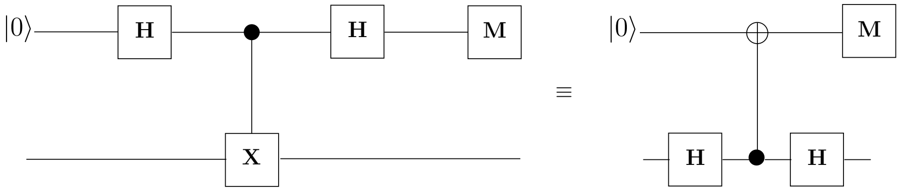


Figure 4.5: Quantum circuit for measuring the **X** operator acting on the bottom qubit [18]. The top qubit is the auxiliary qubit used for computing syndrome. The circuit on the left is the usual construction, while the one on the right is suitable for implementation.

using the quantum circuit of Figure 4.4, where Hadamard gates are used at the end of the circuit of Figure 4.2 to transform the computational basis into the Hadamard basis. The resultant encoded state of Eq. (4.20) is stabilized by the generators $g_1 = \mathbf{XXI}$ and $g_2 = \mathbf{XIX}$, where the computation of the syndrome using the **X** operators is realized in Figure 4.5. We may also observe in these design examples that the Pauli-**X** operators are used in the stabilizer generators for phase-error (**Z**) correction, while the Pauli-**Z** operators are used for bit-error (**X**) correction.

The overall transition from the classical 3-bit repetition code of Section 4.2 to the quantum repetition code is depicted in Figure 4.6, which can be summarized as follows:

- **Encoder**: In classical codes, the information bit may be cloned (or copied) during the encoding process, e.g. in a 3-bit repetition code. This is not permissible in the quantum domain owing to the no-cloning theorem. Alternatively, in quantum codes, the information qubit is entangled with the auxiliary qubits for copying the information in the superposition of basis states, e.g. as in Eq. (4.10) for the 3-qubit bit-flip repetition code and in Eq. (4.19) for the 3-qubit phase-flip repetition code.

- **Channel:** Only bit errors occur during the transmission over a classical channel, e.g. a binary

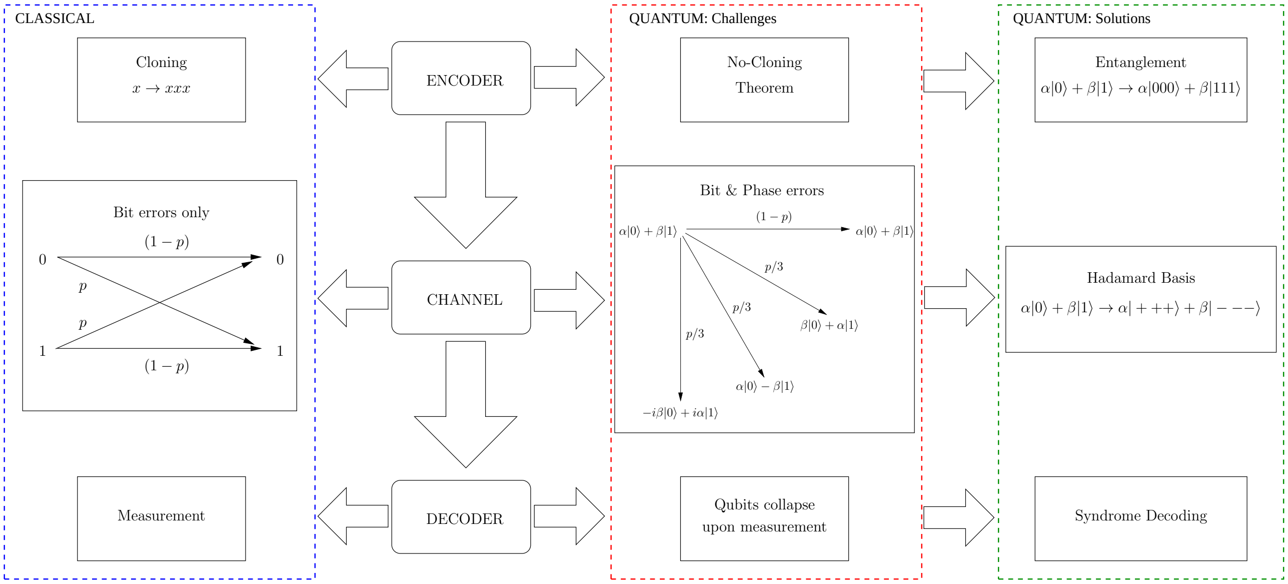

CLASSICAL
Cloning
$x \rightarrow xxx$
Bit errors only
$(1-p)$
0
0
p
p
1
1
$(1-p)$
Measurement
ENCODER
CHANNEL
DECODER
QUANTUM: Challenges
No-Cloning Theorem
Bit & Phase errors
$\alpha|0\rangle + \beta|1\rangle$
$(1-p)$
$\alpha|0\rangle + \beta|1\rangle$
$p/3$
$p/3$
$p/3$
$\beta|0\rangle + \alpha|1\rangle$
$\alpha|0\rangle - \beta|1\rangle$
$-i\beta|0\rangle + i\alpha|1\rangle$
Qubits collapse upon measurement
QUANTUM: Solutions
Entanglement
$\alpha|0\rangle + \beta|1\rangle \rightarrow \alpha|000\rangle + \beta|111\rangle$
Hadamard Basis
$\alpha|0\rangle + \beta|1\rangle \rightarrow \alpha|+++\rangle + \beta|---\rangle$
Syndrome Decoding

Figure 4.6: Transition from the classical to quantum codes.

symmetric channel having a channel crossover probability of p. By contrast, a qubit may experience a bit-flip or a phase-flip as well as both, when subjected to the depolarizing channel having a depolarizing probability of p. Since phase errors in the Hadamard basis $\{|+\rangle, |-\rangle\}$ are similar to the bit errors in the computational basis $\{|0\rangle, |1\rangle\}$, phase errors may be corrected in the same way as the bit errors by exploiting the Hadamard basis.

- **Decoder**: In classical codes, the received bits are measured during the decoding process, e.g. in a 3-bit repetition code a decision may be made on the basis of majority voting. Unfortunately, the qubits collapse upon measurement. Consequently, quantum codes invoke the classical syndrome decoding approach, which circumvents any observation.

Stabilizer generators g_i constituting the stabilizer group $\mathcal{H}$ must exhibit the following two characteristics:

1. **Any two operators in the stabilizer set must commute** so that the stabilizer operators can be applied simultaneously, i.e. we have:

$$g_1 g_2 |\overline{\psi}\rangle = g_2 g_1 |\overline{\psi}\rangle. \tag{4.21}$$

This is because the stabilizer leaves the codeword unchanged as encapsulated below:

$$g_i |\overline{\psi}\rangle = |\overline{\psi}\rangle. \tag{4.22}$$

Hence, evaluating the left-hand and right-hand sides of Eq. (4.21) gives:

$$g_1 g_2 |\overline{\psi}\rangle = g_1 |\overline{\psi}\rangle = |\overline{\psi}\rangle, \tag{4.23}$$

and

$$g_2 g_1 |\overline{\psi}\rangle = g_2 |\overline{\psi}\rangle = |\overline{\psi}\rangle, \tag{4.24}$$

respectively. This further imposes the constraint that the stabilizers should have an even number of places with different non-Identity (i.e. **X**, **Y**, or **Z**) operations. This is derived from the fact that the **X**, **Y** and **Z** operations anti-commute with one another. More explicitly, using Eq. (2.18), it can be shown that:

$$\mathbf{XY} = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix} = \begin{pmatrix} i & 0 \\ 0 & -i \end{pmatrix} = i\mathbf{Z}, \tag{4.25}$$

while we have:

$$\mathbf{YX} = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix} \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} = \begin{pmatrix} -i & 0 \\ 0 & i \end{pmatrix} = -i\mathbf{Z}. \tag{4.26}$$

This implies that:

$$\mathbf{XY} = -\mathbf{YX}. \tag{4.27}$$

Similarly, we can readily show that:

$$\begin{aligned} &\mathbf{YZ} = i\mathbf{X},\ \mathbf{ZY} = -i\mathbf{X} \to \mathbf{YZ} = -\mathbf{ZY} \\ &\mathbf{ZX} = i\mathbf{Y},\ \mathbf{XZ} = -i\mathbf{Y} \to \mathbf{ZX} = -\mathbf{XZ}. \end{aligned} \tag{4.28}$$

Thus, for example the operators **ZZI** and **XYZ** commute, whereas **ZZI** and **YZI** anti-commute.

2. **Generators constituting the stabilizer group $\mathcal{H}$ are closed under multiplication**, i.e. multiplication of the constituent generators g_i yields another generator, which is also part of the stabilizer group $\mathcal{H}$. For example, the full stabilizer group $\mathcal{H}$ of the 3-qubit bit-flip repetition code will also include the operator **IZZ**, which is the product of g_1 and g_2.

4.3.2 Quantum-to-Classical Isomorphism

4.3.2.1 Pauli-to-Binary Isomorphism

QSCs may be characterized in terms of an equivalent classical PCM notation satisfying the commutativity constraint of stabilizers [123, 47] given in Eq. (4.21). This is achieved by mapping the **I**, **X**, **Y** and **Z** Pauli operators onto $(\mathbb{F}_2)^2$ as follows:

$$\mathbf{I} \to (00), \quad \mathbf{X} \to (01), \quad \mathbf{Y} \to (11), \quad \mathbf{Z} \to (10), \tag{4.29}$$

where a binary 1 at the first index represents a **Z** operator, while a binary 1 at the second index represents an **X** operator. More explicitly, the $(n-k)$ stabilizers of an $[n,k]$ stabilizer code constitute the rows of the binary PCM H, which can be represented as a concatenation of a pair of $(n-k) \times n$ binary matrices H_z and H_x based on Eq. (4.29), as given below:

$$H = (H_z|H_x)\,. \tag{4.30}$$

Each row of H corresponds to a stabilizer of $\mathcal{H}$, so that the ith column of H_z and H_x corresponds to the ith qubit and a binary 1 at these locations represents a $\mathbf{Z}$ and $\mathbf{X}$ Pauli operator, respectively, in the corresponding stabilizer. For the 3-qubit bit-flip repetition code, which can only correct bit-flip errors, the PCM H is given by:

$$H = \left(\begin{array}{ccc|ccc} 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 \end{array}\right). \tag{4.31}$$

It must be pointed out here that H_z of Eq. (4.31) is same as the H of the classical repetition code of Eq. (4.9), yielding the same syndrome patterns in Table 4.1 and Table 4.2.

Let us further elaborate the process by considering the $[9,1]$ Shor's code, which consists of the Pauli-$\mathbf{Z}$ as well as the Pauli-$\mathbf{X}$ operators. The corresponding stabilizer generators are given in Table 4.3. They can be mapped onto the binary matrix H as follows:

$$H = \left(\begin{array}{ccccccccc|ccccccccc} 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 \end{array}\right). \tag{4.32}$$

Given the matrix notation of Eq. (4.30), the multiplication of Pauli operators is transformed into the bit-wise addition of the equivalent binary representation. For example, multiplying the set of Pauli operators $\{\mathbf{I}, \mathbf{X}, \mathbf{Z}, \mathbf{Y}\}$ with Pauli-$\mathbf{X}$ is equivalent to the second column of Table 4.4, when the Pauli operators are mapped onto $(\mathbb{F}_2)^2$ according to Eq. (4.29). Furthermore, the commutative property of stabilizers given in Eq. (4.21) is transformed into the orthogonality of rows with respect to the symplectic product (also referred to as a twisted product). If row i is $H_i = (H_{z_i}, H_{x_i})$, where H_{z_i} and H_{x_i} are the binary strings for $\mathbf{Z}$ and $\mathbf{X}$ respectively, then the symplectic product of rows i and i' is given by,

$$H_i \star H_{i'} = (H_{z_i} \cdot H_{x_{i'}} + H_{z_{i'}} \cdot H_{x_i}) \bmod 2. \tag{4.33}$$

This symplectic product is zero if there are even number of places where the non-Identity operators ($\mathbf{X}$, $\mathbf{Y}$ or $\mathbf{Z}$) in the row i and i' are different; thus meeting the commutativity requirement. In other words, if H is written as $H = (H_z|H_x)$, then the symplectic product is satisfied for all the rows only if we have:

$$H_z H_x^T + H_x H_z^T = 0 \bmod 2, \tag{4.34}$$

	Stabilizer
g_1	**ZZIIIIIII**
g_2	**IZZIIIIII**
g_3	**IIIZZIIII**
g_4	**IIIIZZIII**
g_5	**IIIIIIZZI**
g_6	**IIIIIIIZZ**
g_7	**XXXXXXIII**
g_8	**IIIXXXXXX**

Table 4.3: Stabilizers for 9-qubit Shor's code.

+	00	01	10	11
00	00	01	10	11
01	01	00	11	10
10	10	11	00	01
11	11	10	01	00

Table 4.4: $(\mathbb{F}_2)^2$ Addition.

which may be readily verified for the H of Eq. (4.32). Consequently, any classical binary codes satisfying Eq. (4.34) may be used for constructing QSCs. A special class of these stabilizer codes is constituted by the family of Calderbank-Shor-Steane (CSS) codes, which are defined as follows:

An $[n, k_1 - k_2]$ CSS code, which is capable of correcting $(d-1)/2$ bit-flips as well as phase-flips, can be constructed from classical linear block codes $C_1(n, k_1)$ and $C_2(n, k_2)$, if $C_2 \subset C_1$ and both C_1 as well as the dual of C_2, i.e. $C_2^{\perp}$, have a minimum Hamming distance d.

In CSS construction, the PCM H'_z of C_1 is used for correcting bit-flips, while the PCM H'_x of $C_2^{\perp}$ is used for phase-flip correction. Consequently, the PCM of the resultant CSS code assumes the following form:

$$H = \left(\begin{array}{c|c} H'_z & \mathbf{0} \\ \mathbf{0} & H'_x \end{array} \right), \tag{4.35}$$

where we have $H_z = \begin{pmatrix} H'_z \\ \mathbf{0} \end{pmatrix}$, $H_x = \begin{pmatrix} \mathbf{0} \\ H'_x \end{pmatrix}$, where H'_z and H'_x are now $(n-k_1) \times n$ and $k_2 \times n$ binary matrices, respectively. Furthermore, since $C_2 \subset C_1$, the symplectic condition of Eq. (4.34) is reduced

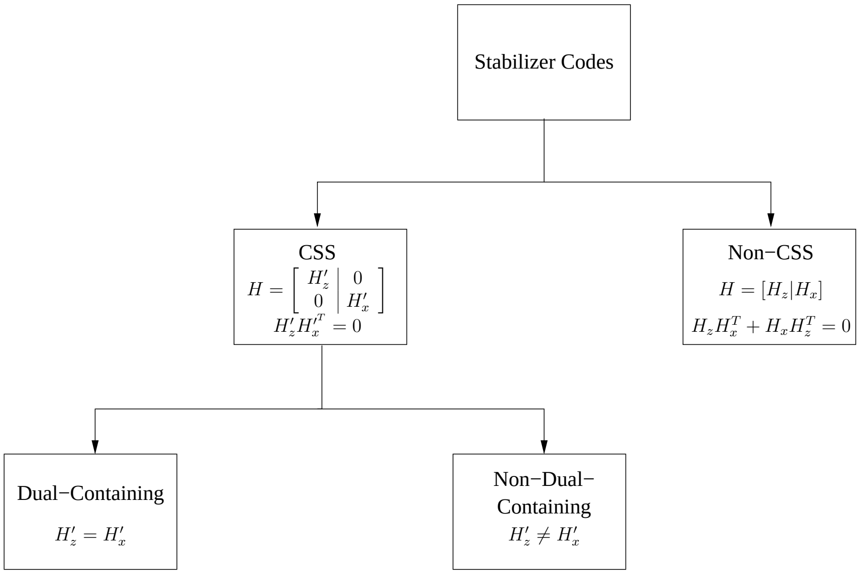


Figure 4.7: Family of stabilizer codes.

to $H'_z H'^T_x = 0$. In this scenario, $(n - k_1 + k_2)$ stabilizers are applied to n qubits. Therefore, the resultant quantum code encodes $(k_1 - k_2)$ information qubits into n qubits. Furthermore, if $H'_z = H'_x$, the resultant structure is termed as a dual-containing (or self-orthogonal) code because $H_z\prime H'^T_z = 0$, which is equivalent to $C_1^\perp \subset C_1$. Hence, the stabilizer codes may be sub-divided into various code structures based on the binary matrix H of Eq. (4.30), as summarized in Figure 4.7.

Let us consider the classical $(7, 4)$ Hamming code, whose PCM is given by:

$$H = \begin{pmatrix} 1 & 1 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 1 \end{pmatrix}. \tag{4.36}$$

Since the H of Eq. (4.36) yields $HH^T = 0$, it is used for constructing the dual-containing rate-1/7 Steane code [35]. The corresponding stabilizer generators are listed in Table 4.5.

Based on the Pauli-to-binary isomorphism encapsulated in Eq. (4.29), a Pauli error $\mathcal{P} \in \mathcal{G}_n$ experienced by an n-qubit block transmitted over a depolarizing channel can be modeled by an effective error-vector P, which is a binary vector of length $2n$. The effective error P may be represented as $P = (P_z, P_x)$, where both P_z and P_x are n-bit long and represent **Z** and **X** errors, respectively. This implies that an **X** error imposed on the tth qubit will yield a 0 and a 1 at the tth and $(n+t)$th index of P, respectively. Similarly, a **Z** error imposed on the tth qubit will give a 1 and a 0 at the tth and $(n+t)$th index of P, respectively, while a **Y** error on the tth qubit will result in a 1 at both the tth as well as $(n+t)$th index of P. The resultant quantum-domain syndrome is given by the symplectic product of H and P, which is formulated as follows:

$$s = H \star P^T = \left(H_z P_x^T + H_x P_z^T\right) \bmod 2, \tag{4.37}$$

	Stabilizer
g_1	**ZZIZZII**
g_2	**ZIZZIZI**
g_3	**IZZZIIZ**
g_4	**XXIXXII**
g_5	**XIXXIXI**
g_6	**IXXXIIX**

Table 4.5: Stabilizers for the Steane code.

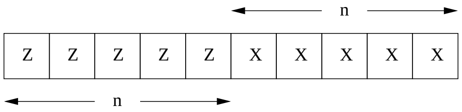


Figure 4.8: Effective classical error P corresponding to the error $\mathcal{P}$ imposed on an n-qubit frame.

where the Pauli-**X** operator is used for correcting **Z** errors, while the Pauli-**Z** operator is used for correcting **X** errors, as previously discussed in the context of 3-qubit bit-flip and phase-flip repetition codes. The resulting syndrome has either a value of 0 or 1. Thus, the quantum-domain syndrome is equivalent to the classical-domain binary syndrome and a basic quantum-domain decoding procedure is similar to the syndrome based decoding of the equivalent classical code [47]. However, due to the degenerate nature of quantum codes (discussed in Section 4.3.3), quantum decoding aims for finding the most likely error coset, while the classical syndrome decoding finds the most likely error.

Hence, an $[n, k]$ QSC associated with $(n - k)$ stabilizers can be effectively modeled using an $(n - k) \times 2n$-element classical PCM satisfying Eq. (4.34). The coding rate of the equivalent classical code R_c can be determined as follows:

$$\begin{aligned} R_c &= \frac{2n - (n-k)}{2n} \\ &= \frac{n+k}{2n} \\ &= \frac{1}{2}\left(1 + \frac{k}{n}\right) \\ &= \frac{1}{2}\left(1 + R_Q\right), \end{aligned} \tag{4.38}$$

where R_Q is its quantum coding rate. Using Eq. (4.38), the coding rate of the classical equivalent of Shors rate-1/9 quantum code is 5/9, while that of the Steane code is 4/7.

$+$	0	1	ω	$\overline{\omega}$
0	0	1	ω	$\overline{\omega}$
1	1	0	$\overline{\omega}$	ω
ω	ω	$\overline{\omega}$	0	1
$\overline{\omega}$	$\overline{\omega}$	ω	1	0

Table 4.6: GF(4) Addition.

$\times$	0	1	ω	$\overline{\omega}$
0	0	0	0	0
1	0	1	ω	$\overline{\omega}$
ω	0	ω	$\overline{\omega}$	1
$\overline{\omega}$	0	$\overline{\omega}$	1	ω

Table 4.7: GF(4) Multiplication.

4.3.2.2 Pauli-to-Quaternary Isomorphism

Since the **I**, **X**, **Y** and **Z** Pauli operators have the equivalent 2-bit representation of Eq. (4.29), they may also be expressed in the Galois Field GF(4) by the equivalent 4-ary symbols. More specifically, the Pauli-to-quaternary isomorphism may be encapsulated as:

$$\mathbf{I} \to 0, \quad \mathbf{X} \to 1, \quad \mathbf{Y} \to \overline{\omega}, \quad \mathbf{Z} \to \omega, \tag{4.39}$$

where 0, 1, ω and $\overline{\omega}$ are the elements of GF(4), which conform to the additive and multiplicative rules of Table 4.6 and Table 4.7[2], respectively. According to this isomorphism, the multiplication of Pauli operators is transformed to the addition of the corresponding elements in GF(4), while the commutativity (symplectic product) criterion is mapped onto the trace[3] inner product [39]. For example, multiplying the set of Pauli operators $\{\mathbf{I}, \mathbf{X}, \mathbf{Z}, \mathbf{Y}\}$ with Pauli-**X** is equivalent to the second column of Table 4.6. On the other hand, the commutative relationship between $\hat{A}$ and $\hat{B}$ in GF(4) is computed using the trace inner product as follows[4]:

$$\mathrm{Tr}\langle \hat{A}, \hat{B} \rangle = \mathrm{Tr}(\hat{A} \times \overline{\hat{B}}) = 0, \tag{4.40}$$

[2] The addition and multiplication rules for GF(p), having a prime p, are the same as the modulo p addition and multiplication, while the rules for GF(p^m), having $m > 1$, do not follow the conventional rules for modulo p^m addition and multiplication. For example, the addition of the elements of GF(4) is equivalent to the bitwise modulo 2 addition of the equivalent 2-bit patterns.

[3] In GF(4), the trace operator maps x to $(x + \overline{x})$. where $\overline{x}$ is the conjugate of x [41].

[4] We denote GF(4) variables with a ˆ on top, e.g. $\hat{x}$.

$\langle,\rangle$	0	1	ω	$\overline{\omega}$
0	0	0	0	0
1	0	1	ω	$\overline{\omega}$
ω	0	$\overline{\omega}$	1	ω
$\overline{\omega}$	0	ω	$\overline{\omega}$	1

Table 4.8: GF(4) Hermitian inner product.

$\mathrm{tr}\langle,\rangle$	0	1	ω	$\overline{\omega}$
0	0	0	0	0
1	0	0	1	1
ω	0	1	0	1
$\overline{\omega}$	0	1	1	0

Table 4.9: GF(4) trace inner product.

where $\langle,\rangle$ represents the Hermitian inner product and $\overline{\hat{B}}$ denotes the conjugate[5] of $\hat{B}$. Moreover, $\mathrm{Tr}(0) = \mathrm{Tr}(1) = 0$, while $\mathrm{Tr}(\omega) = \mathrm{Tr}(\overline{\omega}) = 1$. Explicitly, both the Hermitian inner product and the trace inner product between the elements of GF(4) are tabulated in Table 4.8 and Table 4.9, respectively.

Based on Eq. (4.40), the symplectic product of Eq. (4.33) is transformed into the trace inner product in GF(4). For example, the symplectic product of the ith and i'th row of $\hat{H}$, which is defined in GF(4), is formulated as:

$$\hat{H}_i \star \hat{H}_{i'} = \mathrm{Tr}\langle \hat{H}_i, \hat{H}_{i'} \rangle = \mathrm{Tr}\left(\sum_{t=1}^{n} \hat{H}_{it} \times \overline{\hat{H}}_{i't} \right), \tag{4.41}$$

where $\hat{H}_{it}$ denotes the element in the ith row and tth column of $\hat{H}$.

Let us now prove the equivalence of Eq. (4.33) and Eq. (4.41). Given $H_i = (H_{z_i}, H_{x_i})$ and the mapping of Eq. (4.39), $\hat{H}_i$ may be expressed as:

$$\hat{H}_i = \omega H_{z_i} + H_{x_i}. \tag{4.42}$$

[5] In GF(4), the conjugate operation is defined as $\overline{x} = x^2$ [41]. Consequently, conjugation swaps the elements ω and $\overline{\omega}$, while leaving 0 and 1 intact.

Substituting Eq. (4.42) into Eq. (4.41) yields:

$$\begin{aligned}\hat{H}_i \star \hat{H}_{i'} &= \mathrm{Tr}\langle(\omega H_{z_i} + H_{x_i}), (\omega H_{z_{i'}} + H_{x_{i'}})\rangle \\ &= \mathrm{Tr}\left((\omega H_{z_i} + H_{x_i})\,(\overline{\omega} H_{z_{i'}} + H_{x_{i'}})\right) \\ &= \mathrm{Tr}\left(H_{z_i}H_{z_{i'}} + \omega H_{z_i}H_{x_{i'}} + \overline{\omega}H_{x_i}H_{z_{i'}} + H_{x_i}H_{x_{i'}}\right).\end{aligned} \tag{4.43}$$

Since $\mathrm{Tr}(1) = 0$ and $\mathrm{Tr}(\omega) = \mathrm{Tr}(\overline{\omega}) = 1$, Eq. (4.43) reduces to:

$$\hat{H}_i \star \hat{H}_{i'} = H_{z_i}H_{x_{i'}} + H_{x_i}H_{z_{i'}}, \tag{4.44}$$

which is the same as Eq. (4.33). Consequently, analogous to Eq. (4.37), the syndrome in the quaternary domain is computed as:

$$s_i = \mathrm{Tr}(\hat{s}_i) = \mathrm{Tr}\left(\sum_{t=1}^{n} \hat{H}_{it} \times \overline{\hat{P}_t}\right), \tag{4.45}$$

where s_i is the syndrome corresponding to the ith row of $\hat{H}$ and $\hat{P}_t$ is the tth element of $\hat{P}$, which represents the error inflicted on the tth qubit.

Any arbitrary classical quaternary linear code, which is self-orthogonal with respect to the trace inner product of Eq. (4.41), can be used for constructing a QSC. Since a quaternary linear code is closed under multiplication by the elements of GF(4), this condition reduces to satisfying the Hermitian inner product, rather than the trace inner product [41]. This can be proved as follows.

Let C be a classical linear code in GF(4) having codewords u and v. Furthermore, let us assume that:

$$\langle u, v\rangle = \alpha + \beta\omega. \tag{4.46}$$

For the sake of satisfying the symplectic product, we must have:

$$\mathrm{Tr}\langle u, v\rangle = 0. \tag{4.47}$$

Since $\mathrm{Tr}(\omega) = 1$, Eq. (4.47) is only valid, when β is zero in Eq. (4.46). Furthermore, since the code C is GF(4)-linear, Eq. (4.47) leads to:

$$\mathrm{Tr}\langle u, \overline{\omega} v\rangle = 0, \tag{4.48}$$

which in turn implies that α should also be zero in Eq. (4.46). Hence, for a classical GF(4)-linear code, the Hermitian inner product of Eq. (4.46) must be zero when the trace inner product of Eq. (4.47) is zero. Based on this notion, Calderbank, Rains, Shor and Sloane proposed [41]:

An [n,k] QSC, which is capable of correcting $(d-1)/2$ bit-flips as well as phase-flips, can be constructed in the quaternary domain from a classical self-orthogonal (under the Hermitian inner product) GF(4)-linear block code $C(n,(n-k)/2)$, if the orthogonal code $C^{\perp}(n,(n+k)/2)$ has a minimum Hamming distance d.

The PCM of the resultant QSC is characterized as:

$$\hat{H} = \begin{pmatrix} \hat{H}_c \\ \omega\hat{H}_c \end{pmatrix}, \tag{4.49}$$

	Stabilizer
g_1	**IYZZY**
g_2	**YIYZZ**
g_3	**IXYYX**
g_4	**XIXYY**

Table 4.10: Stabilizers for 5-qubit Hamming code.

Pauli	$(\mathbb{F}_2)^2$	**GF**(4)
I	00	0
X	01	1
Y	11	$\overline{\omega}$
Z	10	ω
Multiplication	Bit-wise Addition	Addition
Commutativity	Symplectic Product	Trace Inner Product

Table 4.11: Quantum-to-classical isomorphism.

where $\hat{H}_c$ is the PCM of the orthogonal code $C^{\perp}(n, (n+k)/2)$. For example, there exists a classical self-orthogonal GF(4)-linear code $C(5, 2)$, whose orthogonal code $C^{\perp}(5, 3)$ is a Hamming code having the PCM $\hat{H}_c$ given by [53]:

$$\hat{H}_c = \begin{pmatrix} 0 & \overline{\omega} & \omega & \omega & \overline{\omega} \\ \overline{\omega} & 0 & \overline{\omega} & \omega & \omega \end{pmatrix}. \tag{4.50}$$

Consequently, the $(5, 1)$ quantum Hamming code can be constructed as:

$$\hat{H} = \begin{pmatrix} 0 & \overline{\omega} & \omega & \omega & \overline{\omega} \\ \overline{\omega} & 0 & \overline{\omega} & \omega & \omega \\ 0 & 1 & \overline{\omega} & \overline{\omega} & 1 \\ 1 & 0 & 1 & \overline{\omega} & \overline{\omega} \end{pmatrix}. \tag{4.51}$$

Based on the Pauli-to-GF(4) mapping of Eq. (4.39), $\hat{H}$ is mapped onto the stabilizer generators listed in Table 4.10.

Hence, a Pauli operator may be expressed in terms of the equivalent binary or quaternary representation, which is summarized in Table 4.11. This in turn facilitates the design of quantum codes from the known classical codes.

4.3.3 Classification of Quantum Errors

The error set of a classical linear block code C having a PCM H can be classified as:

1. **Detected Error Patterns:** These error patterns yield a non-trivial syndrome, i.e. $eH^T \neq 0$, which may be corrected by the code.
2. **Undetected Error Patterns:** This set of error patterns results in a trivial syndrome, i.e. $eH^T = 0$, which cannot be detected by the code. More specifically, an undetected error maps the transmitted codeword onto another valid codeword. Since the resultant codeword still lies in the code space C, it does not trigger a non-zero syndrome. These error patterns are attributed to the small minimum distance of the code.

Analogous to the classical detected error patterns, quantum detected error patterns anti-commute with at least one of the stabilizer generators, which results in a non-trivial syndrome. Similarly, the quantum undetected error patterns commute with all the stabilizer generators, yielding an all-zero syndrome. This commuting set of error patterns is also known as the centralizer (or normalizer) of the stabilizer code having the stabilizer group $\mathcal{H}$, which is denoted as $C(\mathcal{H})$ (or $N(\mathcal{H})$). In particular, the centralizer of an $[n, k]$ QSC is a dual subspace consisting of n-tuple Pauli errors $\mathcal{P} \in \mathcal{G}_n$, which are orthogonal to all the stabilizers of the stabilizer group $\mathcal{H}$. Furthermore, since the $\mathcal{H}$ is itself an Abelian group consisting of mutually orthogonal generators, it is contained in the centralizer, i.e. we have $\mathcal{H} \subset N(\mathcal{H})$. Recall from Section 4.3.1 that the stabilizer generators do not disturb the state of valid codewords. This in turn implies that errors which belong to the stabilizer group, i.e. we have $\mathcal{P} \in \mathcal{H}$, do not corrupt the transmitted codewords and therefore may be classified as the harmless undetected error patterns. This class of errors does not have any classical analogue. By contrast, those error patterns, which lie in the subspace $N(\mathcal{H}) \setminus \mathcal{H}$, are the harmful undetected errors, which map one valid codeword onto another. Hence, as depicted in Figure 4.9, quantum error patterns can be classified as follows:

1. **Detected Errors Patterns:** These error patterns fall outside the normalizer subspace, i.e. they satisfy $\mathcal{P} \in \mathcal{G}_n \setminus N(\mathcal{H})$.
2. **Harmful Undetected Error Patterns:** This set of error patterns is defined as $N(\mathcal{S}) \setminus \mathcal{H}$.
3. **Harmless Undetected Errors Patterns:** These error patterns fall in the stabilizer group $\mathcal{H}$.

The set of harmless undetected error patterns gives quantum codes the intrinsic property of 'degeneracy' [67]. More explicitly, Pauli errors which differ only by the elements of the stabilizer group have the same impact on all the codewords and therefore can be corrected by the same recovery operations. For example, the errors $\mathcal{P}$ and $\mathcal{P}' = g_i\mathcal{P}$ have the same impact on the transmitted codeword, because we have:

$$\mathcal{P}'[|\overline{\psi}\rangle] = g_i\mathcal{P}[|\overline{\psi}\rangle] = \mathcal{P}g_i[|\overline{\psi}\rangle]. \tag{4.52}$$

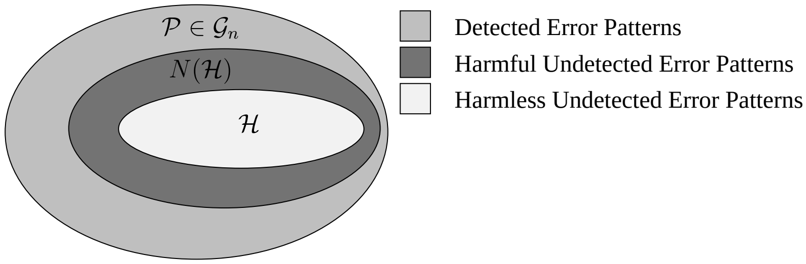

$\mathcal{P} \in \mathcal{G}_n$
$N(\mathcal{H})$
$\mathcal{H}$
Detected Error Patterns
Harmful Undetected Error Patterns
Harmless Undetected Error Patterns

Figure 4.9: Error pattern classification for stabilizer codes.

Since $g_i[|\overline{\psi}\rangle] = |\overline{\psi}\rangle$, we get:

$$\mathcal{P}'[|\overline{\psi}\rangle] = \mathcal{P}[|\overline{\psi}\rangle]. \tag{4.53}$$

Therefore, degenerate errors can be corrected by the same recovery operation. Getting back to our example of the 3-qubit bit-flip repetition code of Section 4.3.1, let $\mathcal{P} = \mathbf{IIX}$ and $\mathcal{P}' = g_1\mathcal{P} = \mathbf{ZZX}$. Both $\mathcal{P}$ as well as $\mathcal{P}'$ corrupt the transmitted codeword of Eq. (4.11) to $\alpha|001\rangle + \beta|110\rangle$. Consequently, $\mathcal{P}$ and $\mathcal{P}'$ do not have to be differentiated and are therefore classified as degenerate errors. Thus, degeneracy enables a quantum code to pack more information as compared to the underlying classical design.

4.4 Quantum Convolutional Codes

Quantum Convolutional Codes (QCCs) are derived from the corresponding classical convolutional codes using stabilizer formalism. This is based on the equivalence between the Classical Convolutional Codes (CCC) and the classical linear block codes with semi-infinite length, which is derived below [124].

Consider a $(2, 1, m)$ classical convolutional code with generators,

$$\begin{aligned} g^{(0)} &= (g_0^{(0)}, g_1^{(0)}, \dots, g_m^{(0)}), \\ g^{(1)} &= (g_0^{(1)}, g_1^{(1)}, \dots, g_m^{(1)}). \end{aligned} \tag{4.54}$$

For an input sequence $[u = (u_0, u_1, u_2, \dots)]$, the output sequences $[v^{(0)} = (v_0^{(0)}, v_1^{(0)}, v_2^{(0)}, \dots)]$ and $[v^{(1)} = (v_0^{(1)}, v_1^{(1)}, v_2^{(1)}, \dots)]$are given as follows:

$$\begin{aligned} v^{(0)} &= u \circledast g^{(0)}, \\ v^{(1)} &= u \circledast g^{(1)}, \end{aligned} \tag{4.55}$$

where $\circledast$ denotes discrete convolution (modulo 2), which implies that for all $l \geq 0$ we have:

$$v_l^{(j)} = \sum_{i=0}^{m} u_{l-i} g_i^{(j)} = u_l g_0^{(j)} + u_{l-1} g_1^{(j)} + \dots + u_{l-m} g_m^{(j)}, \tag{4.56}$$

where $j = 0, 1$ and $u_{l-i} \triangleq 0$ for all $l < i$. The two encoded sequences are multiplexed into a single codeword sequence v given by:

$$v = (v_0^{(0)}, v_0^{(1)}, v_1^{(0)}, v_1^{(1)}, v_2^{(0)}, v_2^{(1)}, \dots) \tag{4.57}$$

This encoding process can also be represented in matrix notation by interlacing the generators $g^{(0)}$ and $g^{(1)}$ and arranging them in matrix form as follows[6],

$$G = \begin{pmatrix} g_0^{(0)(1)} & g_1^{(0)(1)} & \dots & g_m^{(0)(1)} & & & \\ & g_0^{(0)(1)} & g_1^{(0)(1)} & \dots & g_m^{(0)(1)} & & \\ & & g_0^{(0)(1)} & g_1^{(0)(1)} & \dots & g_m^{(0)(1)} & \\ & & \ddots & & \dots & \ddots & \end{pmatrix}, \tag{4.58}$$

where $g_i^{(0)(1)} \triangleq \left(g_i^{(0)} g_i^{(1)} \right)$. The encoding operation of Eq. (4.56) is therefore equivalent to,

$$v = uG. \tag{4.59}$$

Since the information sequence u is of arbitrary length, G is semi-infinite. Furthermore, each row of G is identical to the previous row, but is shifted to the right by two places (since $n = 2$). In practice, u has a finite length N. Therefore, G has N rows and $2(m + N)$ columns for CCC$(2, 1, m)$. For CCC(n, k, m), G can be generalized as follows:

$$G = \begin{pmatrix} G_0 & G_1 & \dots & G_m & & \\ & G_0 & G_1 & \dots & G_m & \\ & & G_0 & G_1 & \dots & G_m \\ & & \ddots & & \dots & \ddots \end{pmatrix}, \tag{4.60}$$

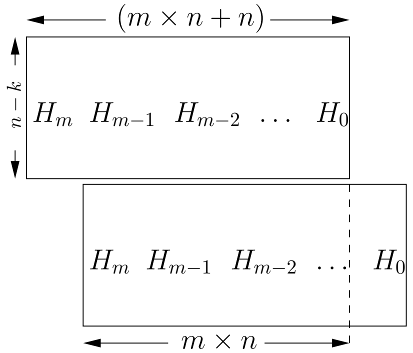
where G_l is a (k x n) submatrix with entries,

$$G_l = \begin{pmatrix} g_{1,l}^{(0)} & g_{1,l}^{(1)} & \dots & g_{1,l}^{(n-1)} \\ g_{2,l}^{(0)} & g_{2,l}^{(1)} & \dots & g_{2,l}^{(n-1)} \\ \vdots & \vdots & & \vdots \\ g_{k,l}^{(0)} & g_{k,l}^{(1)} & \dots & g_{k,l}^{(n-1)} \end{pmatrix}. \tag{4.61}$$

The corresponding PCM H can be represented as a semi-infinite matrix consisting of submatrices H_l with dimensions of $(n - k) \times n$. For a convolutional code having m memory elements, H is given by:

$$H = \begin{pmatrix} H_0 & & & & & \\ H_1 & H_0 & & & & \\ H_2 & H_1 & H_0 & & & \\ \vdots & \vdots & \vdots & & & \\ H_m & H_{m-1} & H_{m-2} & & \dots & H_0 & \\ & H_m & H_{m-1} & H_{m-2} & & \dots & H_0 \\ & & \vdots & \vdots & & & \vdots \end{pmatrix}. \tag{4.62}$$

[6] Blank spaces in the matrix indicate zeros.



$(m \times n + n)$
$n - k$
H_m H_{m-1} H_{m-2} $\ldots$ H_0
H_m H_{m-1} H_{m-2} $\ldots$ H_0
$m \times n$

Figure 4.10: Block-band structure of the semi-infinite classical PCM H.

Therefore, a CCC can be represented as a linear block code with semi-infinite block length. Furthermore, if each row of the submatrices H_l is considered as a single block and $h_{j,i}$ is the ith row of the jth block, then H has a block-band structure after the first m blocks, whereby the successive blocks are time-shifted versions of the first block ($j = 0$) and the adjacent blocks have an overlap of m submatrices. This has been depicted in Figure 4.10 and can be mathematically represented as follows:

$$h_{j,i} = [\mathbf{0}^{j \times n}, h_{0,i}], \ 1 \leq i \leq (n-k), \ 0 \leq j, \tag{4.63}$$

where $\mathbf{0}^{j \times n}$ is a row-vector with $(j \times n)$ zeros.

As discussed in Section 4.3.2, the rows of a classical PCM correspond to the stabilizers of a quantum code. Hence, the quantum stabilizer group $\mathcal{H}$ of an $[n, k, m]$ stabilizer convolutional code is given by [51]:

$$\mathcal{H} = sp\{g_{j,i} = I^{\otimes jn} \otimes g_{0,i}\}, \ 1 \leq i \leq (n-k), \ 0 \leq j, \tag{4.64}$$

where $g_{j,i}$ is the ith stabilizer of the jth block of the stabilizer group $\mathcal{H}$. Furthermore, sp represents a symplectic group, thus implying that all the stabilizers $g_{j,i}$ must be independent and must commute with each other.

As proposed by Forney in [52, 53], CSS-type QCCs can be derived from the classical self-orthogonal binary convolution codes. Let us consider the rate 1/3 QCC of [52, 53], which is constructed from a binary rate-1/3 CCC with generators:

$$G = \left(\begin{array}{ccc|ccc|ccc|ccc|c} 1 & 1 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & \ldots \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & \ldots \\ & & & & \ldots & & & & & & & & \end{array} \right). \tag{4.65}$$

In D-transform notation, these generators are represented as $(1 + D + D^2, 1 + D^2, 1)$. Each generator is orthogonal to all other generators under the binary inner product, making it a self-orthogonal code. Moreover, the dual $C^{\perp}$ has the capability of correcting 1 bit. Therefore, based on the CSS construction, the basic stabilizers of the corresponding single-error correcting $[3,1]$ QCC are as follows:

$$g_{0,1} = (\mathbf{XXX}, \mathbf{XII}, \mathbf{XXI})\,, \tag{4.66}$$

$$g_{0,2} = (\mathbf{ZZZ}, \mathbf{ZII}, \mathbf{ZZI})\,. \tag{4.67}$$

Other stabilizers of $\mathcal{H}$ are the time-shifted versions of these basic stabilizers as depicted in Eq. (4.64).

Let us further consider a non-CSS QCC construction given by Forney in [52, 53]. It is derived from the classical self-orthogonal rate-1/3 quaternary ($\mathbb{F}_4$) convolutional code C having generators $(1+D, 1+wD, 1+\overline{w}D)$, where $\mathbb{F}_4 = \{0, 1, w, \overline{w}\}$. These generators can also be represented as follows:

$$G = \left(\begin{array}{ccc|ccc|ccc|c} 1 & 1 & 1 & 1 & w & \bar{w} & 0 & 0 & 0 & \dots \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & w & \bar{w} & \dots \\ & & & & \dots & & & & & \end{array}\right). \tag{4.68}$$

Since all these generators are orthogonal under the Hermitian inner product, C is self-orthogonal. Therefore, a $[3,1]$ QCC can be derived from this classical code. The basic generators $g_{0,i}$, for $1 \leq i \leq 2$, of the corresponding stabilizer group, $\mathcal{H}$, are generated by multiplying the generators of Eq. (4.68) with w and $\bar{w}$, and mapping 0, w, 1, $\bar{w}$ onto **I**, **X**, **Y** and **Z** respectively. The resultant basic stabilizers are as follows:

$$g_{0,1} = (\mathbf{XXX}, \mathbf{XZY})\,, \tag{4.69}$$

$$g_{0,2} = (\mathbf{ZZZ}, \mathbf{ZYX})\,, \tag{4.70}$$

and all other constituent stabilizers of $\mathcal{H}$ can be derived using Eq. (4.64).

4.5 Entanglement-Assisted Quantum Codes

Let us recall that the classical binary and quaternary codes may only be used for constructing stabilizer codes if they satisfy the symplectic criterion of Eq. (4.34). Consequently, some of the well-known classical codes cannot be explored in the quantum domain. This limitation can be readily overcome by using the entanglement-assisted stabilizer formalism, which exploits pre-shared entanglement between the transmitter and receiver to embed a set of non-commuting stabilizer generators into a larger set of commuting generators.

Figure 4.11 shows the general schematic of a quantum communication system, which incorporates an Entanglement-Assisted Quantum Stabilizer Code (EA-QSC). An $[n, k, c]$ EA-QSC, having a coding rate $R_Q = k/n$ and an entanglement consumption rate $\mathtt{E} = c/n$, encodes the information qubits $|\psi\rangle$ into the coded sequence $|\overline{\psi}\rangle$ with the aid of $(n-k-c)$ auxiliary qubits, which are initialized to the state $|0\rangle$. Furthermore, the transmitter and receiver share c entangled qubits (ebits) before their actual transmission takes place. This may be carried out during the off-peak hours, when the channel is under-utilized, thus efficiently distributing the transmission requirements in time. More specifically,

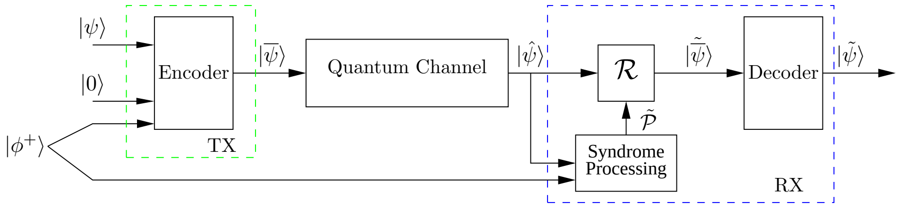


Figure 4.11: System Model: Quantum communication system relying on an entanglement-assisted quantum stabilizer code.

.

the state $|\phi^+\rangle$ of an ebit is given by the following Bell state:

$$|\phi^+\rangle = \frac{|00\rangle^{T_X R_X} + |11\rangle^{T_X R_X}}{\sqrt{2}}, \tag{4.71}$$

where T_X and R_X denotes the transmitter's and receiver's half of the ebit, respectively. Similar to the superdense coding protocol of [28], it is assumed that the receiver's half of the c ebits are transmitted over a noiseless quantum channel, while the transmitter's half of the c ebits together with the $(n-k-c)$ auxiliary qubits are used for encoding the intended k information qubits into n coded qubits. The resultant n-qubit codewords $|\overline{\psi}\rangle$ are transmitted over a noisy quantum channel. The receiver then combines his half of the c noiseless ebits with the received n-qubit noisy codewords $|\hat{\psi}\rangle$ to compute the syndrome, which is used for estimating the error pattern $\tilde{\mathcal{P}}$ incurred on the n-qubit codewords. The rest of the processing at the receiver is the same as that in Figure 4.1.

The entangled state of Eq. (4.71) has unique commutativity properties, which assist us in transforming a set of non-Abelian generators into an Abelian set. The state $|\phi^+\rangle$ is stabilized by the operators $\mathbf{X}^{T_X}\mathbf{X}^{R_X}$ and $\mathbf{Z}^{T_X}\mathbf{Z}^{R_X}$, which commute with each other. Therefore, we have[7]:

$$[\mathbf{X}^{T_X}\mathbf{X}^{R_X}, \mathbf{Z}^{T_X}\mathbf{Z}^{R_X}] = 0. \tag{4.72}$$

However, local operators acting on either of the qubits anti-commute, i.e. we have:

$$\{\mathbf{X}^{T_X}, \mathbf{Z}^{T_X}\} = \{\mathbf{X}^{R_X}, \mathbf{Z}^{R_X}\} = 0. \tag{4.73}$$

Therefore, if we have two single qubit operators $\mathbf{X}^{T_X}$ and $\mathbf{Z}^{T_X}$, which anti-commute with each other, then we can resolve the anti-commutativity by entangling another qubit and choosing the local operators on this additional qubit such that the resultant two-qubit generators ($\mathbf{X}^{T_X}\mathbf{X}^{R_X}$ and $\mathbf{Z}^{T_X}\mathbf{Z}^{R_X}$ for this case) commute. This additional qubit constitutes the receiver half of the ebit. In other words, we entangle an additional qubit for the sake of ensuring that the resultant two-qubit operators have an even number of places with different non-identity operators, which in turn ensures commutativity.

[7] $[a, b]$ represents the commutative relation between a and b, while $\{a, b\}$ denotes the anti-commutative relation.

Let us consider a pair of classical binary codes associated with the following PCMs:

$$H_z = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 \end{pmatrix}, \tag{4.74}$$

and

$$H_x = \begin{pmatrix} 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{pmatrix}, \tag{4.75}$$

which are used to construct a non-CSS quantum code having $H = (H_z|H_x)$. The PCM H does not satisfy the symplectic criterion. The resultant non-Abelian set of Pauli generators are as follows:

$$H_Q = \begin{pmatrix} \mathbf{X} & \mathbf{Z} & \mathbf{X} & \mathbf{I} \\ \mathbf{X} & \mathbf{X} & \mathbf{I} & \mathbf{X} \\ \mathbf{Y} & \mathbf{Z} & \mathbf{Z} & \mathbf{X} \\ \mathbf{X} & \mathbf{Y} & \mathbf{Y} & \mathbf{Z} \end{pmatrix}. \tag{4.76}$$

In Eq. (4.76), the first two generators (i.e. the first and second row) anti-commute, while all other generators commute with each other. This is because the local operators acting on the second qubit in the first two generators anti-commute, while the local operators acting on all other qubits in these two generators commute. In other words, there is a single index (i.e. 2) with different non-Identity operators. To transform this non-Abelian set into an Abelian set, we may extend the generators of Eq. (4.76) with a single additional qubit, whose local operators also anti-commute for the sake of ensuring that the resultant extended generators commute. Therefore, we get:

$$H_Q = \left(\begin{array}{cccc|c} \mathbf{X} & \mathbf{Z} & \mathbf{X} & \mathbf{I} & \mathbf{Z} \\ \mathbf{X} & \mathbf{X} & \mathbf{I} & \mathbf{X} & \mathbf{X} \\ \mathbf{Y} & \mathbf{Z} & \mathbf{Z} & \mathbf{X} & \mathbf{I} \\ \mathbf{X} & \mathbf{Y} & \mathbf{Y} & \mathbf{Z} & \mathbf{I} \end{array}\right), \tag{4.77}$$

where the operators to the left of the vertical bar (|) act on the transmitted n-qubit codewords, while those on the right of the vertical bar act on the receiver's half of the ebits.

4.6 Summary and Conclusions

Qubits cannot be cloned or copied and they collapse to classical bits upon measurement. Moreover, in contrast to a classical channel, which imposes only bit errors, the quantum channel inflicts both bit as well as phase errors. Hence, classical coding techniques cannot be directly applied to the quantum domain. Against this background, in this chapter we have provided insights into the quantum stabilizer formalism, which aids in the development of quantum codes from the known classical codes.

In Section 4.2, we reviewed the classical linear block codes, which is the basis of the stabilizer formalism. We then discussed the stabilizer codes in Section 4.3. More specifically, we laid out the general stabilizer formalism in Section 4.3.1 and detailed the transition from the classical codes to the quantum codes, which was summarized in Figure 4.6. Particularly, with the aid of design examples, namely the 3-qubit bit-flip repetition code and the 3-qubit phase-flip repetition code, it was demonstrated that entanglement can be invoked for circumventing the cloning issue, the Hadamard basis can be exploited for phase-error correction, while the classical syndrome decoding approach can be used for observing the channel errors without reading the actual received qubits. We then derived the equivalence between the quantum codes and the classical binary and quaternary codes in Sections 4.3.2.1 and 4.3.2.2, respectively, which was summarized in Table 4.11. Based on this quantum-to-classical isomorphism, arbitrary classical binary and quaternary codes can be used for constructing QSCs, provided they meet the stringent commutativity criterion, which is characterized by the symplectic product in the binary domain and by the Hermitian inner product in the quaternary domain. Following this design rule, we constructed $[7,1]$ and $[5,1]$ QSCs from the classical dual-containing $(7,4)$ Hamming code and the classical self-orthogonal GF(4)-linear $(5,2)$ code, respectively. Furthermore, various code structures, namely dual-containing CSS, non-dual-containing CSS and non-CSS, associated with the binary PCM of a stabilizer code were summarized in Figure 4.7. In Section 4.3.3, we presented the classification of quantum errors. It was pointed out that quantum codes are inherently degenerate. Consequently, errors, which differ by an element of the stabilizer group, can be corrected with the same recovery operation. Furthermore, errors, which belong to the stabilizer group, constitute the class of harmless undetected errors, which has no classical analogue. In Section 4.4, we laid out the design of QCCs from the CCCs. Since convolutional codes are equivalent to semi-infinite block codes, it was shown that the designs of Section 4.3 can be readily extended to the QCCs. Finally, in Section 4.5, EA-QSCs were presented, which facilitate the design of quantum codes from arbitrary classical codes when they do not meet the commutativity requirement, hence virtually all classical codes can be imported to the quantum domain. However, this is achieved at the cost of pre-shared transmission of entangled qubits, which is a valuable resource and therefore should be minimized.

As discussed in this chapter, QSCs invoke the classical PCM-based syndrome decoding approach. More explicitly, the stabilizer generators of a QSC are used for computing the syndrome, which collapses to a binary 0 or 1 upon measurement. Since the resultant syndrome is binary and the generators of a QSC can be mapped onto an equivalent classical PCM, the observed syndromes are fed to a classical PCM-based syndrome decoder for estimating the channel error. Therefore, in Chapter 5 we will focus our attention on the classical syndrome decoding techniques.

Chapter 5

Classical Syndrome Decoding

5.1 Introduction

In Chapter 4, we provided insights into the construction of stabilizer codes from the known classical codes based on the underlying quantum-to-classical isomorphism. Let us recall that since a Quantum Stabilizer Code (QSC) can be mapped onto an equivalent classical binary or quaternary Parity Check Matrix (PCM), classical PCM-based syndrome decoding may be invoked during the quantum decoding process. More explicitly, the 'syndrome processing' block of Figure 4.1 may be expanded, as shown in Figure 5.1. The process begins with the computation of the syndrome of the received sequence $|\hat{\psi}\rangle$ using the stabilizer generators, which collapse to a binary 0 or 1 upon measurement. For estimating the equivalent channel error $\tilde{P}$ (or $\tilde{\tilde{P}}$ in quaternary domain), the binary syndrome sequence s is fed to a classical PCM-based syndrome decoder, which operates over the equivalent classical PCM associated with the QSC. Finally, the estimated binary (or quaternary) error is mapped onto the equivalent Pauli error $\tilde{\mathcal{P}}$ using the binary-to-Pauli mapping of Eq. (4.29) (or quaternary-to-Pauli mapping of Eq. (4.39)). The classical syndrome decoder of Figure 5.1 is exactly the same decoder, which we would use for any conventional classical code, with the exception of the following two differences:

1. In contrast to the syndrome of a classical code, which is computed as the product of the PCM

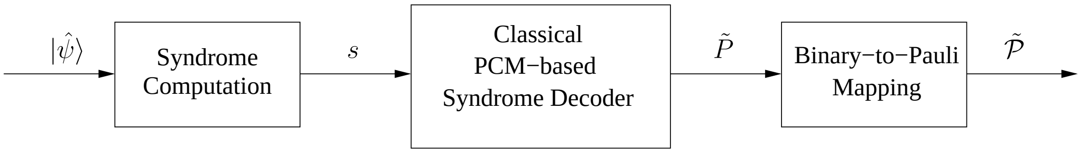


Figure 5.1: Syndrome processing block of Figure 4.1.

and the transpose of the channel error (HP^T), the syndrome of a quantum code is computed using the symplectic product of Eq. (4.37) (or the trace inner product of Eq. (4.45)).

2. The conventional classical decoding aims for finding the most likely error, given the observed syndrome, while quantum decoding aims for finding the most likely error coset, which takes into account the degenerate nature of quantum codes, as discussed in Section 4.3.3.

In this chapter, we will focus our attention on the classical syndrome decoding techniques for the conventional classical codes for transmission over a classical channel, thereby ignoring these two differences. The concepts developed in this chapter together with those of Chapter 4 will be subsequently used in the later chapters in the context of quantum codes. We have also conceived a reduced-complexity syndrome-based decoder in this chapter.

In contrast to conventional codeword decoding, which finds the most likely codeword, having the minimum Hamming distance, syndrome decoding finds the most likely error, having the minimum Hamming weight. The notion of syndrome decoding stems from the Look-Up Table (LUT) based decoding of linear block codes, whereby the syndrome of the received sequence characterizes the inflicted error using a pre-computed LUT [124]. An LUT-based syndrome decoder is in essence a minimum-distance decoder, which finds the error vector having the minimum Hamming weight. By contrast, the soft-decision Maximum Likelihood (ML) codeword decoding of a linear block code requires a brute force attempt for computing the conditional probability for all possible codewords $\overline{x}$, given the received sequence y $\mathrm{P}(\overline{x}|y)$ (or the probability of all possible errors e given the observed syndrome s for syndrome decoding $\mathrm{P}(e|s)$). To circumvent this tedious task, Bahl *et al.* [125] were the first to conceive the syndrome-based[1] code trellis[2] for linear block codes. However, they did not provide a detailed construction. This gap was filled by Wolf in [126], whereby the method of constructing the syndrome-based code trellis for linear block codes was presented. Wolf [126] also proved that in contrast to a brute force ML decoding of a linear block code $C(n, k)$ over GF(q), which would require q^k evaluations, a code trellis requires only $q^{\min\{k,n-k\}}$ states. The ideas presented in [125, 126] for the trellis-based ML codeword decoding of linear block codes are readily applicable to the trellis-based ML syndrome decoding, which relies on the corresponding syndrome-based error trellis. Parallel to these developments, Schalkwijk and Vinckin [127, 128, 129] conceived the idea of a syndrome-based error trellis for convolutional codes. They exploited the inherent symmetries of the trellis structure for reducing the complexity of the decoding hardware required for the hard-decision syndrome decoding of convolutional codes. Later, soft-decision syndrome decoding approaches were presented in [130] and [131], which were based on the error trellis and code trellis, respectively. This concept was further extended to the family of high-rate turbo codes in [132].

[1] We call this trellis syndrome-based because it is constructed from the PCM of the linear block code, while the classic trellis of a convolutional code is constructed using the code generators.

[2] Each path of a code trellis is a valid codeword, while each path of an error trellis is a possible channel error, which would yield a given syndrome sequence. Therefore, a code trellis is used for codeword decoding, while an error trellis is used for syndrome decoding.

The error trellis-based syndrome decoding is of particular significance, because the state probabilities of an error trellis are a function of the channel errors rather than of the coded sequence. Consequently, at high Signal-to-Noise Ratios (SNRs), the syndrome decoder is more likely to encounter a zero-state due to the predominant error-free transmissions. This underlying property of syndrome decoding has been exploited in [133, 134] for developing a Block Syndrome Decoder (BSD) for convolutional codes, which divides the received sequence into erroneous and error-free parts based on the syndrome. More specifically, the BSD only decodes the erroneous blocks, with the initial and final states of the trellis initialized to zero. Therefore, the decoding complexity is substantially reduced at higher SNRs. It also offers a potential for parallelization [135]. The concept of BSD was further extended to turbo codes in [136], where a pre-correction sequence[3] was also computed at each iteration to correct the errors. Consequently, the Hamming weight of the syndrome sequence decreases with ongoing iterations. Thus, the decoding complexity was reduced not only at higher SNRs, but also for the higher-indexed iterations. Furthermore, a syndrome-based Maximum *A-Posteriori* (MAP) decoder was proposed in [137] for designing an adaptive low-complexity decoding approach for turbo equalization. Some other applications of BSD are dealt with in [138, 139, 140].

Inspired by the significant decoding complexity reductions reported for BSD, our novel contribution in this chapter is that we have extended the application of the syndrome-based MAP decoder of [137] together with the BSD of [136] to Turbo Trellis Coded Modulation (TTCM) for the sake of reducing its decoding complexity [6]. The resultant scheme is referred to as BSD-TTCM. We have investigated the performance of our proposed BSD-TTCM for transmission over the Additive White Gaussian Noise (AWGN) channel as well as over an uncorrelated Rayleigh fading channel in this chapter.

The rest of this chapter is organized as follows. In Section 5.2, we detail the LUT-based syndrome decoding method, which forms the basis of the syndrome decoding concept. This is followed by a discussion on the trellis-based syndrome decoding in Section 5.3. Particularly, Section 5.3.1 focuses on the construction of the syndrome-based trellis of linear block codes, while in Section 5.3.2 we extend this syndrome-based trellis formalism to convolutional codes. Section 5.4 deals with the BSD. More specifically, in Section 5.4.1, we lay out the general BSD formalism, while our proposed block-based syndrome decoder designed for TTCM is presented in Section 5.4.2. We then evaluate the performance of our proposed decoder in Section 5.5. Finally, we summarize the chapter in Section 5.6.

5.2 Look-Up Table-based Syndrome Decoding

The LUT-based syndrome decoding derives its philosophy from the standard array-based decoding. For a binary linear block code $C(n, k)$, standard array is a $2^{n-k} \times 2^k$ array, which distributes all the possible 2^n n-tuple vectors into 2^k disjoint subsets of size 2^{n-k}, such that each subset contains only one valid codeword. More specifically, a standard array is constructed as follows:

[3] Pre-correction sequence is an estimated/predicted error sequence, which is used to correct errors in the received information.

- Let $\{\overline{x}_1, \overline{x}_2, \ldots, \overline{x}_{2^k}\}$ be the set of valid codewords (n-bit each) of $C(n,k)$. Place all these 2^k codewords of C in the first row of the standard array, commencing from the all-zero codeword.
- Select any minimum weight vector e_2 from the pool of 2^n n-bit vectors, which is not contained in the first row. List the coset $(e_2 + C)$ in the second row such that $e_2 + \overline{x}_i$ is in the ith column, where $\overline{x}_i$ is the ith valid codeword.
- Select another minimum weight vector e_3 from the remaining pool, i.e. excluding the vectors contained in the first two rows, and list the coset $(e_3 + C)$ in the third row.
- Continue the process for all 2^{n-k} cosets.

The resultant standard array may be expressed as:

$$\begin{pmatrix} \overline{x}_1 = 0 & \overline{x}_2 & \ldots & \overline{x}_i & \ldots & \overline{x}_{2^k} \\ e_2 & e_2 + \overline{x}_2 & \ldots & e_2 + \overline{x}_i & \ldots & e_2 + \overline{x}_{2^k} \\ e_3 & e_3 + \overline{x}_2 & \ldots & e_3 + \overline{x}_i & \ldots & e_3 + \overline{x}_{2^k} \\ \vdots & & & & & \vdots \\ e_{2^{n-k}} & e_{2^{n-k}} + \overline{x}_2 & \ldots & e_{2^{n-k}} + \overline{x}_i & \ldots & e_{2^{n-k}} + \overline{x}_{2^k} \end{pmatrix}, \tag{5.1}$$

where $\{e_1 = 0, e_2, \ldots, e_{2^{n-k}}\}$ constitutes the set of possible minimum-weight errors, which may be identified using the code $C(n,k)$. More explicitly, each row of Eq. (5.1) is a unique coset, whose coset leader is the minimum weight vector given in the first column. We may notice in Eq. (5.1) that adding (modulo 2) the coset leader to any element of the same coset yields the corresponding valid codeword. For example, if we add e_3 to the second element of the third coset according to $e_3 + \overline{x}_2$, we get $\overline{x}_2$. Hence, the coset leader identifies the most likely (having the minimum Hamming weight) error for its coset.

Since the PCM-based syndrome decoding approach discussed in Section 4.2 is an LUT-based syndrome decoder, let us construct the associated standard array. Recall from Section 4.2 that a 3-bit repetition code $C(3,1)$ has two valid codewords, i.e. $\overline{x}_1 = [000]$ and $\overline{x}_2 = [111]$. Using Eq. (5.1) and this set of valid codewords, we get the following standard array:

$$\begin{pmatrix} \overline{x}_1 & \overline{x}_2 \\ e_2 & e_2 + \overline{x}_2 \\ e_3 & e_3 + \overline{x}_2 \\ e_4 & e_4 + \overline{x}_2 \end{pmatrix} = \begin{pmatrix} 000 & 111 \\ 001 & 110 \\ 010 & 101 \\ 100 & 011 \end{pmatrix}. \tag{5.2}$$

Let us assume furthermore that the received vector $y = [011]$ lies in the fourth coset. Consequently, the estimated error is [100] and the corresponding estimated codeword is [011] + [100] = [111], which is the first element of the subset (column) to which y belongs.

Since the standard array is a $(2^{n-k} \times 2^k)$-element array, it imposes huge storage requirements for large linear block codes. This may be alleviated by using the LUT-based syndrome decoding. Let us consider the ith element of the jth coset of the standard array of Eq. (5.1). Its syndrome may be computed as:

$$s = (e_j + \overline{x}_i)H^T = e_j H^T + \overline{x}_i H^T = e_j H^T. \tag{5.3}$$

Hence, each coset is identified by a unique syndrome. Consequently, rather than storing all the 2^k elements of the coset, we can construct an LUT, which only stores the coset leader and the corresponding syndrome, i.e. we have:

$$\begin{pmatrix} e_1 = 0 & 0 \\ e_2 & e_2 H^T \\ e_3 & e_3 H^T \\ e_4 & e_4 H^T \end{pmatrix} = \begin{pmatrix} 000 & 00 \\ 001 & 01 \\ 010 & 10 \\ 100 & 11 \end{pmatrix}. \tag{5.4}$$

This is equivalent to the LUT given in Table 4.1, which also has 2 columns and $2^{n-k} = 4$ rows.

5.3 Trellis-based Syndrome Decoding

Trellis-based syndrome decoding operates over a syndrome-based error trellis. More specifically, each path of an error trellis characterizes a unique channel error for a given syndrome. Consequently, the set of paths of the error trellis for a particular syndrome is the same as the coset of the standard array (Eq. (5.1)) corresponding to that syndrome. When the syndrome is zero, which is equivalent to the first coset of Eq. (5.1), i.e. to set of all valid codewords, the error trellis collapses to a code trellis. The trellis-based syndrome decoding therefore invokes either the classic Viterbi algorithm [141] or the Bahl-Cocke-Jelinek-Raviv (BCJR) decoding (also called MAP decoding) [125] for estimating the most likely channel error for a given syndrome. In this context, let us now have a look at the construction of the syndrome-based error trellis for linear block codes and convolutional codes in Section 5.3.1 and 5.3.2, respectively. Finally, the MAP algorithm will be presented later in Section 5.4.2.2.

5.3.1 Linear Block Codes

Consider a linear block code $C(n, k)$ constituted over GF(q) having an $(n-k) \times n$ PCM H, whose ith column is represented by h_i for $i = \{1, 2, \ldots, n\}$. The syndrome-based trellis of this code is defined by a set of states interconnected by unidirectional edges, where a state is basically represented by an $(n-k)$-bit syndrome. Analogously to a conventional trellis, the edges are drawn between the trellis-states at depth t and those at depth $(t-1)$, for $t = \{0, 1, \ldots, n\}$, with the direction of the edge emerging from the state at depth $(t-1)$ and arriving at the state at depth t. At any trellis-depth t, there are at most q^{n-k} nodes and the lth trellis-state at depth t is denoted as $s_l(t)$. Based on this notation, let us now construct the syndrome-based trellis of Figure 5.2 for a binary code $C(5, 3)$, whose PCM is given by,

$$H = \begin{pmatrix} 1 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 \end{pmatrix} = \begin{pmatrix} h_1 & h_2 & h_3 & h_4 & h_5 \end{pmatrix}. \tag{5.5}$$

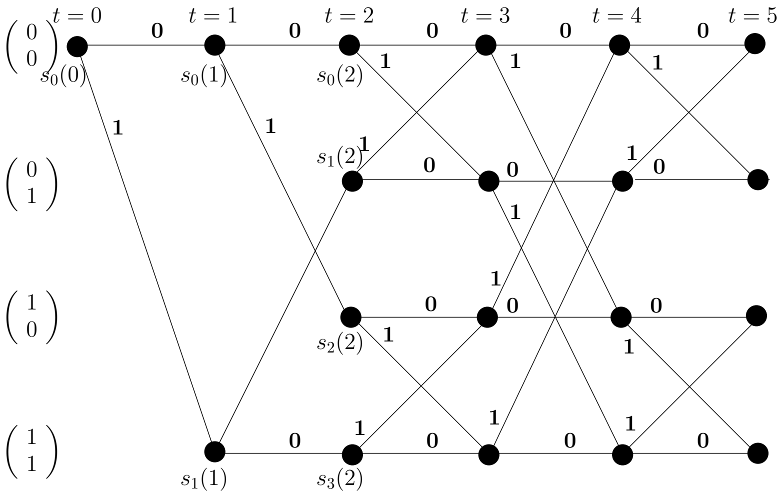


Figure 5.2: Syndrome-based trellis for a linear block code $C(5, 3)$ constructed over GF(2).

The trellis of Figure 5.2 can be constructed as follows:

1. The trellis emerges from the trellis-state $(0, 0)$ at $t = 0$, i.e. we have $s_0(0) = (0, 0)$.
2. We next determine the set of trellis-states for $t = \{1, \ldots, 5\}$ as follows:
   (a) We compute the edges emerging from the trellis-state $s_0(0)$ at trellis-depth $t = 0$ to the states at depth $t = 1$ using the relationship:

$$s_l(t) = s_i(t-1) + \alpha_j h_t, \quad \text{for } j = 0, 1, \ldots, (q-1), \tag{5.6}$$

where $s_l(t)$ is the lth state at depth t, $s_i(t-1)$ is the ith state at depth $(t-1)$ and α_j is an element of GF(q). Since we are using a binary code, there are only two possible values of α_j in Eq. (5.6) i.e. $\alpha_0 = 0$ and $\alpha_1 = 1$. Substituting these values of α_j into Eq. (5.6) at $t = 1$ yields:

$$\begin{aligned} s_0(1) &= s_0(0) + \alpha_0 h_1 \\ &= \begin{pmatrix} 0 \\ 0 \end{pmatrix} + 0.\begin{pmatrix} 1 \\ 1 \end{pmatrix} \\ &= \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \end{aligned} \tag{5.7}$$

$$\begin{aligned} s_1(1) &= s_0(0) + \alpha_1 h_1 \\ &= \begin{pmatrix} 0 \\ 0 \end{pmatrix} + 1.\begin{pmatrix} 1 \\ 1 \end{pmatrix} \\ &= \begin{pmatrix} 1 \\ 1 \end{pmatrix}. \end{aligned} \tag{5.8}$$

Consequently, the trellis-state $s_0(0) = (0, 0)$ at $t = 0$ is connected to the states $s_0(1) = (0, 0)$ and $s_1(1) = (1, 1)$ at $t = 1$, as seen in the trellis of Figure 5.6. The corresponding edges are labeled by 0 and 1 for $\alpha_0 = 0$ and $\alpha_1 = 1$, respectively.

Eq. (5.7) and Eq. (5.8) may also be computed using an alternate approach. We know that for a received vector y and the PCM H, the syndrome vector s is given by,

$$s = yH^T. \tag{5.9}$$

At $t = 1$, we receive only the first element of y i.e. $y = [y_1 0000]^T$. Therefore, Eq. (5.7) and Eq. (5.8) can also be formulated as:

$$s_l(1) = \begin{pmatrix} 1 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} y_1 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} \quad \text{where } y_1 \in \{\alpha_0, \alpha_1\}. \tag{5.10}$$

(b) The process is similarly repeated for $t = 2$, where we have:

$$\begin{aligned} s_l(2) &= s_0(1) + \alpha_j h_2 \\ s_l(2) &= s_1(1) + \alpha_j h_2, \end{aligned} \tag{5.11}$$

for $j = \{0, 1\}$. If the alternate approach of Eq. (5.10) is adopted, Eq. (5.11) may be written as:

$$s_l(2) = \begin{pmatrix} 1 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} y_1 \\ y_2 \\ 0 \\ 0 \\ 0 \end{pmatrix}, \quad \text{where } y_1, y_2 \in \{\alpha_0, \alpha_1\}, \tag{5.12}$$

since we have two received bits y_1 and y_2 at $t = 2$.

(c) This process is repeated until we reach the end of the PCM, i.e $t = 5$.

3. Finally, any paths, which do not terminate at the all-zero syndrome, can be removed, hence resulting in the expurgated code trellis, which is shown in Figure 5.3. Since only the paths terminating at the all-zero syndrome are considered, these paths correspond to the valid codewords. For generating the error trellis, we discard all paths, which do not terminate at the syndrome of the received vector y. Consequently, each path of the expurgated error trellis defines a possible error sequence. In other words, the error trellis collapses to a code trellis, when the syndrome observed is zero.

The above-mentioned process for constructing a syndrome-based trellis of a linear block code may be generalized as follows:

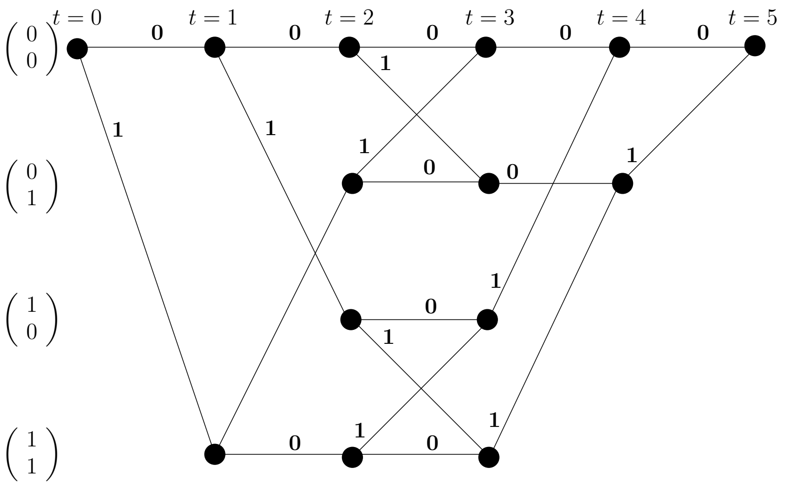


Figure 5.3: Expurgated syndrome-based code trellis for the binary code $C(5,3)$.

1. The trellis starts from an all-zero state, i.e. there is a single state at $t = 0$ denoted by $s_0(0)$, which is equivalent to an all-zero vector of length $(n-k)$.

2. For each $t = \{1, 2, \ldots, n\}$, the trellis-states at depth t are obtained from the set of states at depth $(t-1)$ using the Eq. (5.6). Edges are drawn between the state $s_i(t-1)$ at trellis-depth $t-1$ and the q states $s_l(t-1)$ at depth t, which are labeled by the corresponding value of α_j.

3. For a code trellis, the constructed trellis is expurgated by removing those paths, which do not lead to an all-zero state at depth n. Consequently, each of the q^k paths in the resultant trellis defines a valid codeword. By contrast, for the error trellis, the constructed trellis is expurgated by discarding those paths, which do not terminate at the syndrome observed, i.e. at the syndrome of the received vector y.

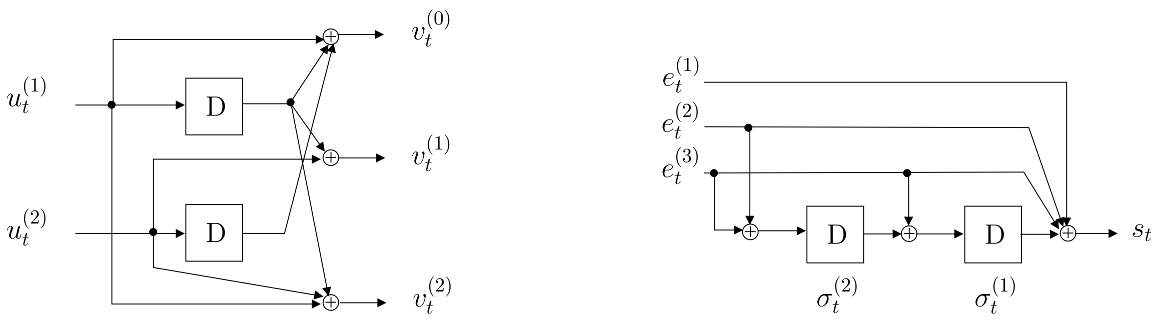
5.3.2 Convolutional Codes

The conventional code trellis of a convolutional code is derived from the generator matrix G. By contrast, an error trellis is constructed using the corresponding PCM, which is known as the syndrome former H^T for convolutional codes. Let us review the example given in [142] for illustrating the trellis construction procedure. We use a rate-2/3 convolutional code, whose generator $G(D)$ is given by:

$$G(D) = \begin{pmatrix} 1+D & D & 1+D \\ D & 1 & 1 \end{pmatrix}. \tag{5.13}$$

The corresponding syndrome former is as follows:

$$H^T(D) = \begin{pmatrix} 1 \\ 1+D^2 \\ 1+D+D^2 \end{pmatrix}. \tag{5.14}$$



(a) Encoder G

(b) Syndrome Former H^T

Figure 5.4: Realization of the encoder and syndrome former given in Eq. (5.13) and Eq. (5.14), respectively.

The circuit of $G(D)$ for $\mathrm{CC}(n,k)$ is realized in Figure 5.4(a), whereby the input information sequence $u_t = (u_t^{(1)}, \ldots, u_t^{(k)})$ at time instant t is encoded by the generator $G(D)$, yielding the output code sequence of $v_t = (v_t^{(1)}, \ldots, v_t^{(n)})$. Let $e_t = (e_t^{(1)}, \ldots, e_t^{(n)})$ be the channel error experienced during transmission. The received sequence $y_t = (y_t^{(1)}, \ldots, y_t^{(n)})$ is therefore given by:

$$y_t^j = v_t^j \oplus e_t^j. \tag{5.15}$$

Analogously to the linear block codes, we may define the resultant syndrome sequence s using the syndrome former H^T as follows[4]:

$$s = yH^T = (v + e)H^T = eH^T. \tag{5.16}$$

This realization of the syndrome former is shown in Figure 5.4(b), which employs the error sequence $e_t = (e_t^{(1)}, \ldots, e_t^{(n)})$ as its input for computing the corresponding syndrome s_t at time instant t using the syndrome former of Eq. (5.14). Consequently, we may construct the trellis for the syndrome former of Figure 5.4(b) as we conventionally do using the encoder G. Furthermore, since the syndrome s_t can either have a value of 0 or 1, we divide the resultant trellis into two sub-trellis modules corresponding to $s_t = 0$ and $s_t = 1$, as shown in Figure 5.5. Here, the state at time t is defined as $\sigma_t = (\sigma_t^{(1)}\sigma_t^{(2)})$ and each branch leading from σ_{t-1} to σ_t is labeled with the error $e_t = (e_t^{(1)}, e_t^{(2)}, e_t^{(3)})$. Let us assume that the received sequence is:

$$\{y_t\}_{t=1}^{3} = \{011 \;\; 011 \;\; 111\}. \tag{5.17}$$

Since the received sequence y_t and the inflicted channel error e_t yield the same syndrome according to Eq. (5.16), we feed y_t into the circuit of Figure 5.4(b) for computing the syndrome. Consequently, using the syndrome former of Figure 5.4(b), whose registers are initialized to the zero state, we get $\{s_t\}_{t=1}^{3} = \{0, 1, 0\}$, while the resultant states are $\sigma_1 = (10)$, $\sigma_2 = (10)$ and $\sigma_3 = (10)$, respectively. The corresponding trellis can be constructed by starting from the state $\sigma_0 = (00)$ and concatenating the sub-trellises of Figure 5.5 based on the value of s_t. More specifically, for the syndrome values of

[4] Recall from Section 4.4 that the PCM of a convolutional code is a semi-infinite matrix. The semi-infinite binary matrix H^T corresponding to $H^T(D)$ of Eq. (5.14) is defined later in Eq. (5.20).

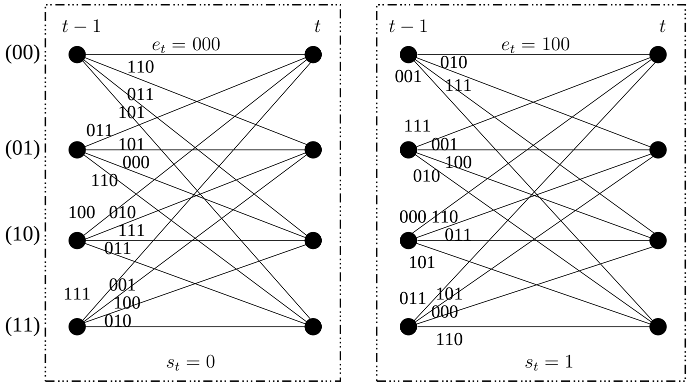


Figure 5.5: Syndrome-based error sub-trellises for H^T corresponding to $s_t = 0$ and $s_t = 1$.

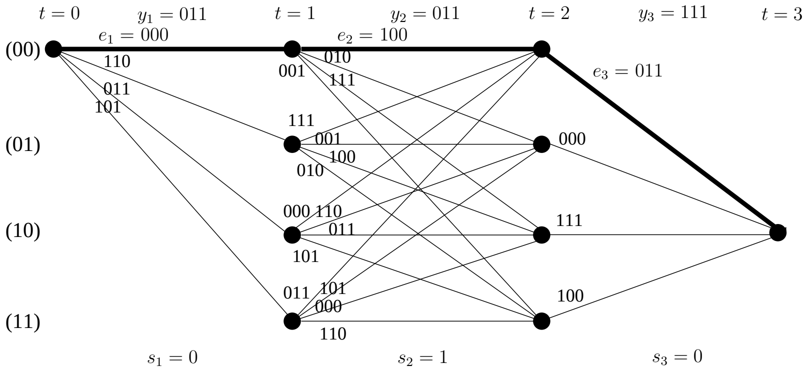


Figure 5.6: Syndrome-based error trellis for H^T corresponding to the received sequence of Eq. (5.17), which is constructed by concatenating the sub-trellises of Figure 5.5.

$\{s_t\}_{t=1}^{3} = \{0, 1, 0\}$, we concatenate the sub-trellis for $s_t = 0$ to that for the state $s_t = 1$, followed by another sub-trellis for the $s_t = 0$. This yields the trellis of Figure 5.6 for the received sequence of Eq. (5.17), which is initialized to the state $\sigma_0 = (00)$ and terminates at $\sigma_3 = (10)$.

The error trellis of Figure 5.6 is equivalent to the conventional code trellis generated using the generator G, which is shown in Figure 5.7. Here the state at time t is $(u_t^{(1)}, u_t^{(2)})$ and each branch is labeled with the coded bits $(v_t^{(1)}, v_t^{(2)}, v_t^{(3)})$. Furthermore, the trellis emerges from and it is terminated at the all-zero states[5]. Each path of Figure 5.6 corresponds to a path of Figure 5.7 and this correspondence is one-to-one relationship [142]. Consider an arbitrary path $\tilde{e} = \{000\ \ 100\ \ 011\}$ of Figure 5.6 (marked with a thick line). Since the received sequence is $y = \{011\ \ 011\ \ 111\}$, the estimated transmitted code sequence $\tilde{v}$ is computed as:

$$\tilde{v}_t^{(j)} = \tilde{e}_t^{(j)} \oplus y_t^{(j)} \ . \tag{5.18}$$

Hence, we have $\tilde{v} = \{011\ \ 111\ \ 100\}$, which is a path of the trellis seen in Figure 5.7 (marked with a thick line). We may, therefore, conclude that for every error path in the error trellis, there is a corresponding unique path in the conventional code trellis. Either the Viterbi or the MAP algorithm

[5] It is assumed that termination bits are used.

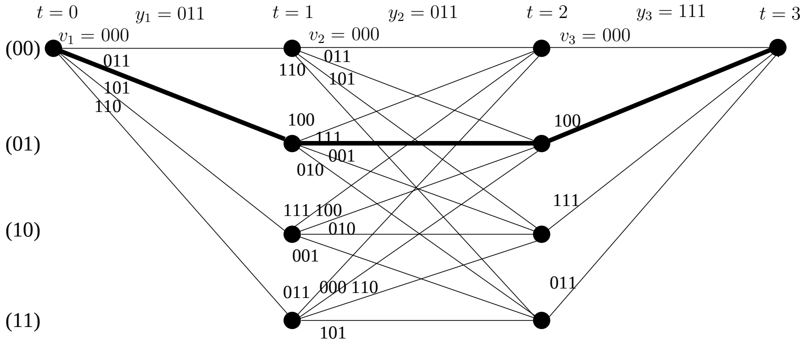


Figure 5.7: Conventional code trellis, generated using the encoder G of Figure 5.4(a), corresponding to Figure 5.6. Each path of Figure 5.6 can be mapped onto a path of the code trellis using Eq. (5.18).

can then be used for determining the most likely error $\tilde{e}$ for the received sequence. The estimated error sequence is then added (bit-wise modulo 2) to the received sequence y as in Eq. (5.18), yielding the most likely transmitted code sequence $\tilde{v}$. It is then passed through the inverse encoder G^{-1} for estimating the most likely information sequence $\tilde{u}$ as follows:

$$\tilde{u} = \tilde{v}G^{-1}. \tag{5.19}$$

The syndrome-based error trellis of Figure 5.6 has the same complexity as the conventional code trellis of Figure 5.7 for soft-decision decoding, which is $\sim nq^k q^m$ for $\text{CC}(n, k, m)$ and grows exponentially upon increasing the value of k (increasing the coding rate R). By contrast, Sidorenko *et al.* [131] proposed a syndrome-based trellis, which has a lower complexity[6] than the conventional trellis for both hard-decision and soft-decision decoding. This construction is derived from the Wolf trellis of [126] conceived for linear block codes, which was discussed in Section 5.3.1. The complexity of the resultant trellis is $\sim nq^{\min(k,n-k)}q^m$, which decreases upon increasing R for $R > \frac{1}{2}$ and it is approximately equivalent to that of the conventional trellis for $R < \frac{1}{2}$. More explicitly, Sidorenko's method [131] of constructing the syndrome-based trellis divides each branch of Figure 5.6 into n stages.

Let us now continue the same example as in Eq. (5.14). Recall from Section 4.4 that a convolutional code is equivalent to a semi-infinite linear block code. For the syndrome former of Eq. (5.14), the associated semi-infinite parity check matrix can be constructed as:

$$H = \begin{pmatrix} 111 & & & \\ 001 & 111 & & \\ 011 & 001 & 111 & \\ & 011 & 001 & 111 \\ & \ddots & & \ddots \end{pmatrix}, \tag{5.20}$$

[6] The complexity of a code trellis is the number of operations (selections and additions) required to decode a block [131].

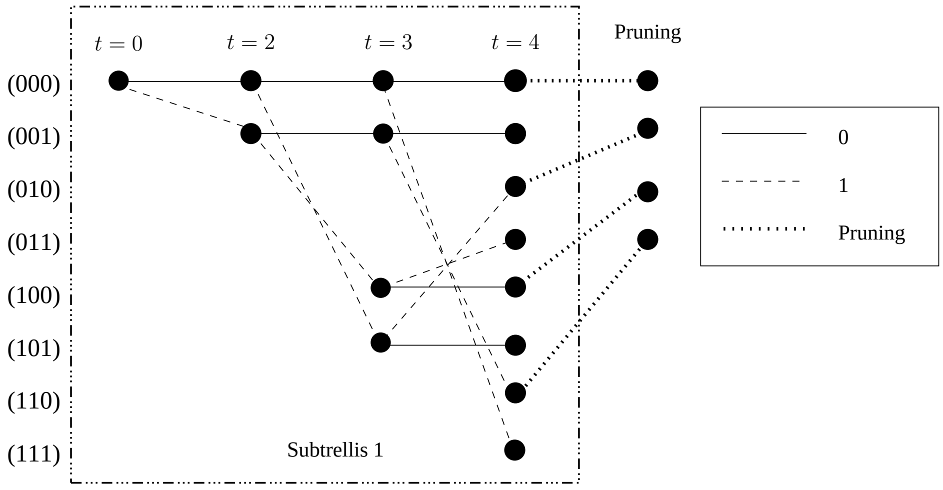


Figure 5.8: Syndrome-based code trellis for the PCM of Eq. (5.20).

which is composed of the time-shifted versions of the basic PCM H_b given by:

$$H_b = \begin{pmatrix} 111 \\ 001 \\ 011 \end{pmatrix}. \tag{5.21}$$

Consequently, we can construct its syndrome-based trellis by interconnecting a series of sub-trellises formed using the basic PCM H_b, where the number of interconnected sub-trellises is equal to the length of the received bit stream.

Figure 5.8 shows the first sub-trellis of the syndrome-based code trellis for the PCM H of Eq. (5.20), which has ($2^3 = 8$) states since the associated H_b has 3 rows (syndromes). The trellis of Figure 5.8 is constructed as follows:

1. The trellis emerges from an all-zero state i.e from $(0,0,0)$ at $t = 0$.

2. The first sub-trellis is constructed using Eq. (5.21). Here each state is labeled by three syndromes, namely $s_3s_2s_1$[7], where s_1, s_2 and s_3 correspond to the first, second and third row of H_b. The states at depth t are obtained from the set of states at depth $(t-1)$ using the relationship:

$$s_l(t) = s_i(t-1) + \alpha_j h_t, \quad \text{for } j = 0, 1, \ldots, (q-1). \tag{5.22}$$

Here, $s_l(t)$ is the lth node at depth t, $s_i(t-1)$ is the ith node at depth $(t-1)$, α_j is the element of $\mathbb{F}_q$ and h_t is the tth column of H_b. Connecting lines are drawn between the node $s_i(t-1)$ and the two nodes at depth t and each line is labeled by the corresponding value of α_j. At $t = 1$, we

[7] In Section 5.3.1, state was represented by $s_1s_2s_3$. Both representations are equivalent.

have:

$$\begin{aligned} s_0(1) &= s_0(0) + \alpha_0 h_1 \\ &= \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} + 0.\begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} \\ &= \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}, \qquad (5.23) \\ s_1(1) &= s_0(0) + \alpha_1 h_1 \\ &= \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} + 1.\begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} \\ &= \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}. \qquad (5.24) \end{aligned}$$

Consequently, the state (000) at $t = 0$ is connected to states (000) and (001) at $t = 1$ via bits 0 and 1, respectively. At $t = 2$, we have:

$$\begin{aligned} s_0(2) &= s_0(1) + \alpha_0 h_2 \\ &= \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} + 0.\begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix} \\ &= \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}, \qquad (5.25) \\ s_1(2) &= s_0(1) + \alpha_1 h_2 \\ &= \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} + 1.\begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix} \\ &= \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix}. \qquad (5.26) \end{aligned}$$

$$\begin{aligned} s_2(2) &= s_1(1) + \alpha_0 h_2 \\ &= \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} + 0. \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix} \\ &= \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}, \end{aligned} \tag{5.27}$$

$$\begin{aligned} s_3(2) &= s_1(1) + \alpha_1 h_2 \\ &= \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} + 1. \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix} \\ &= \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}. \end{aligned} \tag{5.28}$$

Consequently, the state (000) at $t = 1$ is connected to states (000) and (101) at $t = 2$ via bits 0 and 1, respectively, while the state (001) at $t = 1$ is connected to states (001) and (100) at $t = 2$ via bits 0 and 1, respectively. This process is repeated for $t = 3$.

3. Recall that if y is the received bit stream, then the syndrome vector s of a linear block code is formulated as:

$$s = yH^T. \tag{5.29}$$

Since in Eq. (5.20) only the first three ($n = 3$) elements of the first row are non-zero, the first element of s in Eq. (5.29), i.e. s_1, is therefore only affected by the first three received bits. Consequently, at $t = 3$, which marks the end of the first sub-trellis, only those paths are retained for which the first bit of the syndrome is 0. This implies that the specific paths terminating at states (000), (010), (100) and (110) are retained, while those ending at (001), (011), (101) and (111) are discarded. Moreover, since the first element s_1 of the syndrome s (least significant bit of the state) is completely defined at this point, we do not have to represent this bit in the trellis anymore. This syndrome bit is discarded, making room for the fourth syndrome bit, which is set to 0 initially. This process of removing the least significant bit is referred to as pruning in [143]. Therefore, the trellis states are now represented by $s_4 s_3 s_2$ and the valid states at $t = 3$, i.e. (000), (010), (100) and (110), are mapped onto (000), (001), (010) and (011).

4. Step 2 is repeated to construct the second sub-trellis starting from the valid states obtained after pruning in Step 3. The process of generating the sub-trellis and pruning for interconnecting them is repeated until the length of trellis becomes equivalent to that of the received bit stream.

5. The resultant trellis represents all the valid codewords. The most likely valid codeword is determined by finding the specific path having the minimum Hamming distance from the received sequence.

This approach can also be adapted for reducing the complexity of the syndrome-based error trellis [144]. For the error trellis, the pruning is carried out on the basis of the syndrome of the received sequence, rather than the zero syndrome.

5.4 Block Syndrome Decoding

5.4.1 General Formalism

The syndrome associated with a received sequence depends on the specific channel conditions. More explicitly, the syndrome is a function of the channel error sequence e encountered during transmission. Consequently, we are more likely to have zero-valued syndromes at higher SNRs, when the channel error sequence e has longer strings of zeros. If these error-free segments are successfully detected, then the decoding complexity can be substantially reduced by decoding only the erroneous portions. More explicitly, the decoder is switched off, when the transmission is assumed to be error-free, yielding the BSD of [133, 134].

A critical design parameter for the BSD is the minimum number of consecutive zero syndromes (L_{min}) after which the sub-block is deemed to be error-free. It must be long enough to ensure that the performance of BSD is the same as that of a full-complexity decoder. A lower value of L_{min} will result in more error-free blocks, thereby reducing the complexity imposed. However, this will increase the likelihood of false detection, thereby degrading the BER performance of the system. On the other hand, a higher value of L_{min} will give a better BER performance, but at the expense of an increased decoding complexity. Hence, L_{min} strikes a trade-off between the BER performance attained and the complexity imposed. The minimum number of consecutive zero syndromes L_{min} can be further split into the parameters L_{off} and L_{on}, where we have $L_{\text{min}} = \big(L_{\text{off}} + L_{\text{on}} + 1\big)$ [133], as shown in Figure 5.9. Here, L_{off} denotes the number of consecutive zero syndromes after which the decoder can be safely switched off, which therefore defines the end of the previous erroneous sub-block. By contrast, L_{on} denotes the number of stages before the first non-zero syndrome, when the decoder has to be switched on again. Therefore, the start of the next erroneous sub-block is defined by L_{on}. More specifically, if L_0 is the length of the sub-block having at least L_{min} consecutive zero syndromes, then the initial L_{off} symbols of this sub-block are appended to the previous erroneous block and the last L_{on} symbols are appended to the following erroneous block. Only the remaining $(L_0 - L_{\text{min}} + 1)$ symbols are considered error-free. This ensures that the trellis of the erroneous sub-blocks starts from and terminates at the all-zero state.

5.4.2 Block Syndrome Decoder for TTCM

Turbo Trellis-Coded Modulation (TTCM) [145] constitutes a bandwidth-efficient near-capacity joint modulation/coding solution, which relies on the classic turbo coding architecture, but involves the

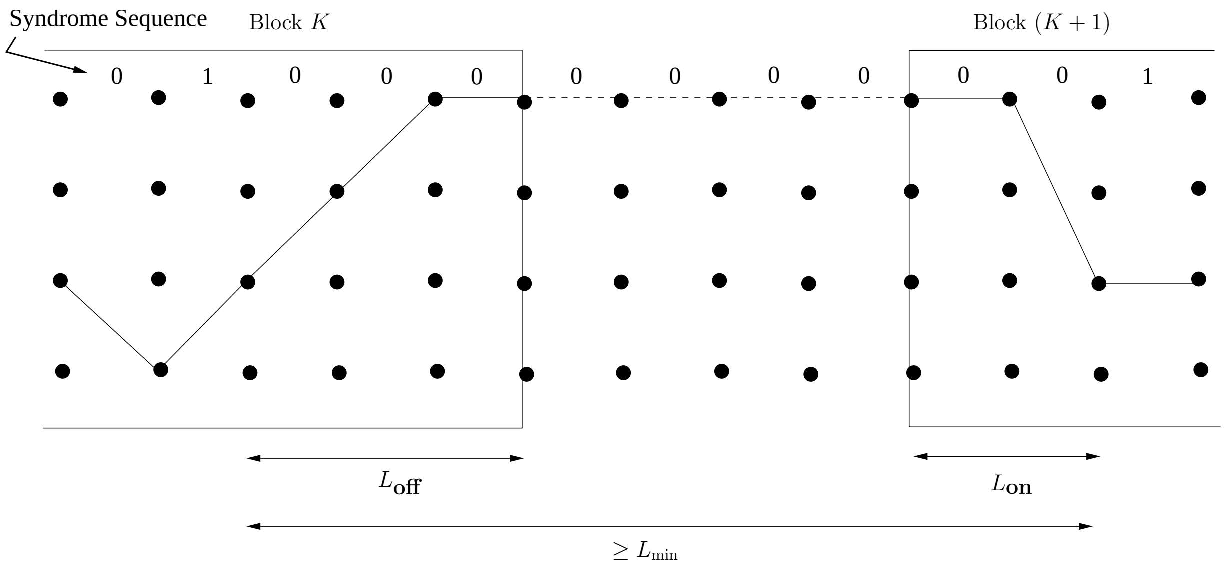


Figure 5.9: Design parameters of BSD [133].

bandwidth-efficient Trellis-Coded Modulation (TCM) [146] instead of the constituent convolutional codes. More explicitly, the constituent TCM codes, which can be optimally designed using EXtrinsic Information Transfer (EXIT) charts [147], are concatenated in a parallel fashion and iterative decoding is invoked at the receiver for exchanging extrinsic information between the pair of TCM decoders. In order to reduce its decoding complexity, we propose to reduce the effective number of decoding iterations by appropriately adapting the syndrome-based block decoding approach of [133, 136] for TTCM.

5.4.2.1 System Model

Figure 5.10 shows the schematic of one of the two constituent decoders of the BSD conceived for TTCM, which we refer to as BSD-TTCM. The received symbol sequence y_t is demapped onto the nearest point x_i in the corresponding 2^n-ary constellation diagram, yielding the hard-demapped symbols $\tilde{y}_t$, i.e. we have:

$$\tilde{y}_t = \arg\min_i (y_t - h_t x_i), \tag{5.30}$$

for $i \in \{0, \ldots, 2^n - 1\}$ and,

$$y_t = h_t x_t + n_t. \tag{5.31}$$

Here, x_t is the complex-valued phasor corresponding to the n-bit transmitted codeword c_t, which is obtained using the 2^n-PSK bit-to-symbol mapper μ as follows:

$$x_t = \mu\left(c_t\right), \tag{5.32}$$

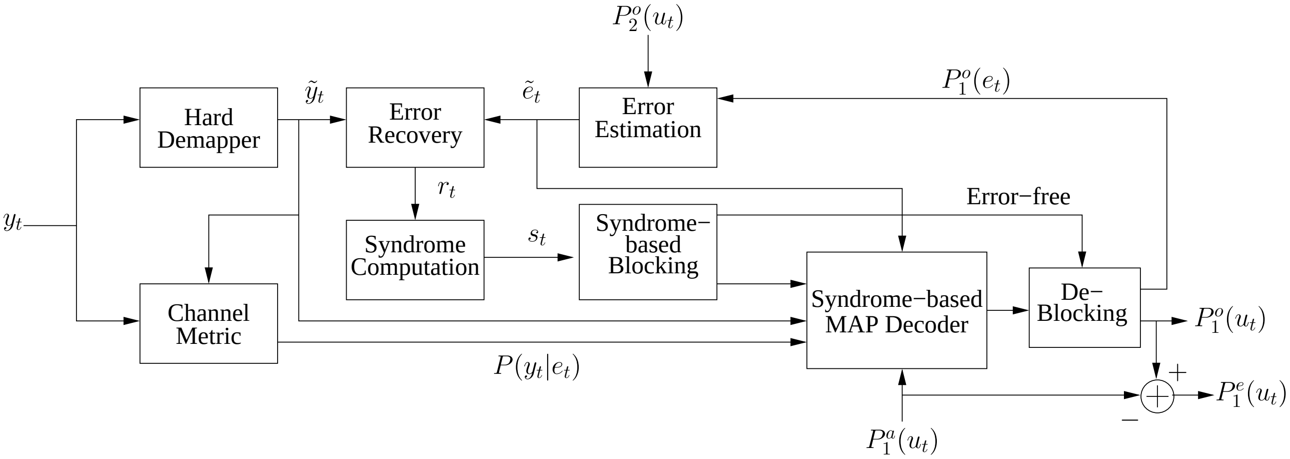


Figure 5.10: Schematic of the proposed BSD-TTCM Decoder. *Only one constituent decoder is shown here.* $\mathrm{P}_i^a[.]$, $\mathrm{P}_i^e[.]$ *and* $\mathrm{P}_i^o[.]$ *are the a-priori, extrinsic and a-posteriori probabilities related to the ith decoder;* e_t *is the channel error on the transmitted symbol and* u_t *is the information part of the tth channel error* e_t*.*

while h_t is the uncorrelated Rayleigh-distributed fading amplitude and n_t is the noise experienced by the tth symbol.

Recall that in TTCM, the odd and even symbols are punctured for the upper and lower TCM encoders, respectively[145]. Consequently, the parity bits of the corresponding hard-demapped punctured symbols are set to zero [145] in the 'Hard Demapper' block of Figure 5.10. Then, a so-called pre-correction sequence $\tilde{e}_t$, which is predicted by the error estimation module, is used for correcting any predicted errors in the hard-demapped output. This sequence is initialized to zero for the first iteration. The syndrome s is computed for the corrected symbol stream r using the syndrome former matrix H^T as follows:

$$s = rH^T, \tag{5.33}$$

where, the jth bit of r_t is related to that of $\tilde{y}_t$ and $\tilde{e}_t$, for $j \in \{0, \ldots, n-1\}$, as follows:

$$r_t^{(j)} = \tilde{y}_t^{(j)} \oplus \tilde{e}_t^{(j)}, \tag{5.34}$$

with $r_t = \left(r_t^{(0)}, \ldots, r_t^{(j)}, \ldots, r_n^{(n-1)}\right)$, $\tilde{y}_t = \left(\tilde{y}_t^{(0)}, \ldots, \tilde{y}_t^{(j)}, \ldots, \tilde{y}_t^{(n-1)}\right)$, and $\tilde{e}_t = \left(\tilde{e}_t^{(0)}, \ldots, \tilde{e}_t^{(j)}, \ldots, \tilde{e}_t^{(n-1)}\right)$.

Then the syndrome is analyzed for sake of dividing the received block into error-free and erroneous sub-blocks. The error-free sub-blocks are then subjected to a hard-decision and only the erroneous sub-blocks are passed to the MAP decoder. Like in the conventional TTCM decoder, both constituent decoders have a similar structure and iterative decoding is invoked for exchanging extrinsic information between the two.

5.4.2.2 Syndrome-Based MAP Decoder

We have invoked the syndrome-based MAP decoder of [137] in the BSD-TTCM of Figure 5.10. In contrast to the conventional MAP decoder, which operates on the basis of the code trellis, its syndrome-based MAP counterpart relies on the error trellis constructed using the syndrome former H^T [127, 142]. More explicitly, each trellis path of a code trellis represents a legitimate codeword. By contrast, each path of an error trellis specifies the hypothetical error sequence causing a departure from a specific legitimate code trellis path. Furthermore, both trellises have the same complexity and every error path in the error trellis uniquely corresponds to a codeword path in the code trellis [142]. The classic MAP algorithm [125] computes the *A-Posteriori* Probability (APP) $\mathrm{P}^o(u_t)$ for every M-ary transmitted information symbol u_t given by $\mathrm{P}^o(u_t) = \mathrm{P}(u_t = m|y_t)$ for $m \in \{0, 1, \ldots, M-1\}$, where $M = 2^{n-1}$, $(n-1)$ is the number of bits in an information symbol and $R = \frac{n-1}{n}$ is the coding rate. However, the syndrome-based MAP computes the APP for every M-ary channel error experienced by the information symbol. In other words, u_t is the transmitted information symbol in the code trellis, whereas, in the error trellis, u_t denotes the M-ary channel error experienced by the information symbol. Therefore, the channel information $\mathrm{P}(y_t|x_t)$ related to the transmitted codeword x_t, is modified to $\mathrm{P}(y_t|e_t)$ for the channel error e_t, which is formulated as:

$$\mathrm{P}(y_t|e_t) = \frac{1}{2\pi\sigma^2} \; . \; e^{-\frac{|y_t - h_t\tilde{x}_t|^2}{2\sigma^2}}, \tag{5.35}$$

where σ^2 is the noise variance per dimension and $\tilde{x}_t$ is given by:

$$\tilde{x}_t = \mu\left(\tilde{c}_t\right), \tag{5.36}$$

for,

$$\tilde{c}_t^{(j)} = \tilde{y}_t^{(j)} \oplus e_t^{(j)}. \tag{5.37}$$

Here, we have $\tilde{c}_t = \left(\tilde{c}_t^{(0)}, \ldots, \tilde{c}_t^{(j)}, \ldots, \tilde{c}_t^{(n)}\right)$ and $e_t = \left(e_t^{(0)}, \ldots, e_t^{(j)}, \ldots, e_t^{(n)}\right)$. The APP of u_t can be calculated in terms of the forward-backward recursive coefficients α_t and β_t as follows:

$$\mathrm{P}^o(u_t) = \sum_{\substack{(\hat{\tau},\tau)\Rightarrow \\ u_t = m}} \gamma_t(\hat{\tau}, \tau) \; . \; \alpha_{t-1}(\hat{\tau}) \; . \; \beta_t(\tau), \tag{5.38}$$

where the summation implies adding all the probabilities associated with those transitions (from state $\hat{\tau}$ to τ) of the error trellis for which $u_t = m$. Furthermore, we have:

$$\begin{aligned} \gamma_t(\hat{\tau}, \tau) &= \mathrm{P}^a(u_t) \; . \; \mathrm{P}(y_t|e_t), \\ \alpha_t(\tau) &= \sum_{\text{all } \hat{\tau}} \gamma_t(\hat{\tau}, \tau) \; . \; \alpha_{t-1}(\hat{\tau}), \\ \beta_{t-1}(\hat{\tau}) &= \sum_{\text{all } \tau} \gamma_t(\hat{\tau}, \tau) \; . \; \beta_t(\tau), \end{aligned} \tag{5.39}$$

where $\mathrm{P}^a(u_t)$ is the *a-priori* probability of the information part of the error e_t, i.e. u_t. At the first iteration, no *a-priori* information is available; hence, it is initialized to be equiprobable, i.e. $\mathrm{P}^a(u_t) = 1/M$.

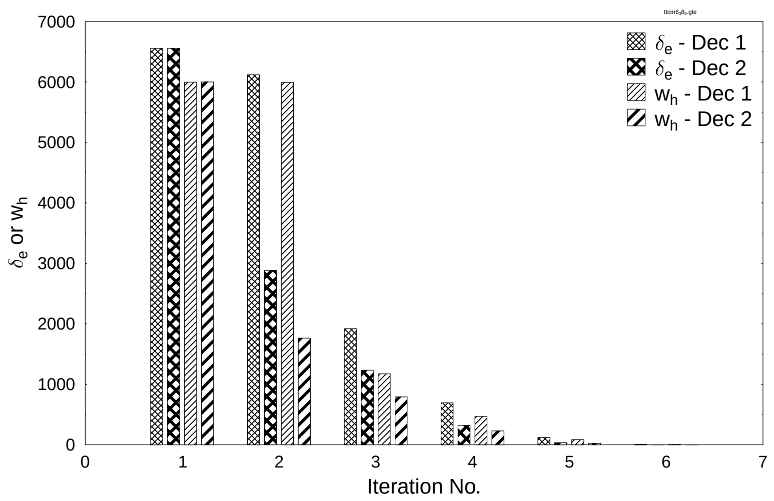


Figure 5.11: Variation in the number of differences (δ_e) between the actual and estimated error and Hamming weight (w_h) of the syndrome with increasing iterations at $E_b/N_0 = 3.8$ dB.

5.4.2.3 Error Estimation

Similar to the bit-wise pre-correction sequence proposed in [136] for turbo codes, we make an estimate of the 2^n-ary symbol error in each iteration to ensure that the Hamming weight of the syndrome decreases with ongoing iterations. While the extrinsic information was used in [136] for the estimating the pre-correction sequence, we have improved the estimation by using the APP instead of the extrinsic. This proceeds as follows:

- The information part of the pre-correction sequence $\tilde{e}_t$ is set to the hard decision of the APP of the information symbol ($\mathrm{P}^o(u_t)$) computed by the other decoder.

- The parity part of $\tilde{e}_t$ is set to the hard decision value of the APP of the codeword ($\mathrm{P}^o(e_t)$) gleaned from the previous iteration of the same decoder, which yields the same information symbol as that computed in the first step.

Figure 5.11 verifies the accuracy of our pre-correction sequence. Here the average number of differences δ_e, between the actual and estimated error, is plotted against the number of iterations at an SNR per bit of $E_b/N_0 = 3.8$ dB, for 1000 frames of 12000 TTCM-8PSK symbols transmitted over an AWGN channel. Both constituent decoders are characterized separately, which are referred to as *Dec 1* and *Dec 2* in Figure 5.11. Observe that the differences decrease at each successive iteration, eventually reaching zero at the 6th iteration. Furthermore, the Hamming weight w_h of the syndrome closely follows the same trend.

5.4.2.4 Syndrome-based Blocking

The Hamming weight of the syndrome sequence of Eq. (5.33) decreases at higher SNRs, since only a few errors are encountered. It also decreases with each successive iteration. This is because the errors are estimated at each iteration and the corresponding correction is applied to the received symbols. In other words, upon increasing the number of iterations or SNR, the syndrome exhibits longer sequences of zeros, which indicates error-free transmission. This fact can be exploited to partition the received block into error-free and erroneous segments, as proposed in [133, 136]. This is achieved by heuristically choosing a design parameter, $L_{\min} = (L_{\text{start}} + L_{\text{end}} + 1)$, which is the minimum number of consecutive zero syndromes after which the sub-block may be deemed error-free. Furthermore, L_{start} and L_{end} define the start and end of the next and previous sub-blocks, respectively. If L_0 is the length of the sub-block having at least $L_{\min}$ consecutive zero syndromes, then the initial $L_{\text{end}} = (L_{\min} - 1)/2$ symbols of this sub-block are appended to the previous erroneous block and the last $L_{\text{start}} = (L_{\min} - 1)/2$ symbols are appended to the following erroneous block [133, 136]. Only the remaining $(L_0 - L_{\min} + 1)$ symbols are considered error-free. This ensures that the trellis of the erroneous sub-blocks starts from and terminates at the zero state. The hypothetical error-free blocks do not undergo further decoding and the corresponding APPs of the error-free trellis segment are set to 1. On the other hand, the erroneous blocks are fed to a MAP decoder with the initial and final states of the decoding trellis set to zero.

It must be mentioned here that the design parameter $L_{\min}$ strikes a trade-off between the BER performance attained and the complexity imposed. A lower value of $L_{\min}$ will result in more error-free blocks, thereby reducing the complexity imposed. However, it will degrade the BER performance of the system. On the other hand, a higher value of $L_{\min}$ will give a better BER performance but at the expense of an increased decoding complexity.

5.5 Results and Discussions

5.5.1 Performance of BSD-TTCM over AWGN Channel

In order to quantify the reduction in decoding complexity achieved with the proposed BSD-TTCM, we have analyzed the performance of TTCM over an AWGN channel using the parameters of Table 5.1. Furthermore, we have heuristically optimized the design parameter $L_{\min}$ while ensuring that the BSD-TTCM yields the same BER as the conventional TTCM decoder. Since the Hamming weight of the syndrome decreases with the SNR, the optimum $L_{\min}$ has to increase with the SNR to ensure that the performance is not compromised. We have particularly focused our attention on the high-SNR region (i.e. $E_b/N_0 \geq 3.5$) and the $L_{\min}$ value was appropriately optimized for every 0.1 dB increment in E_b/N_0, as listed in Table 5.2. It must be mentioned here that the optimum $L_{\min}$ for a particular value of E_b/N_0 depends on the code parameters of Table 5.1 as well as on the channel type. The

Coding Rate	2/3
Modulation	PSK
Interleaver length	$12,000$
Iterations	6

Table 5.1: TTCM parameters.

SNR Range	$L_{\min}$
$E_b/N_0 \leq 3.5$ dB	51
$3.5 < E_b/N_0 \leq 3.6$ dB	111
$3.6 < E_b/N_0 \leq 3.7$ dB	401
$3.7 < E_b/N_0 \leq 3.8$ dB	3001
$3.8 < E_b/N_0 \leq 3.9$ dB	5001

Table 5.2: Optimum $L_{\min}$ for the TTCM of Table 5.1 operating over an AWGN channel.

BER performance of our BSD-TTCM based on the design parameter $L_{\min}$ of Table 5.2 is compared to that of the conventional TTCM decoder in Figure 5.12. Both decoding schemes exhibit a similar performance. The corresponding reduction in the decoding complexity is quantified in Figure 5.13 and Figure 5.14 in terms of:

- **Percentage of No-Decoding**: This quantifies the total number of symbols in the error-free sub-blocks as a percentage of the frame length (i.e. 12000).
- **Equivalent number of iterations**: Each iteration is weighted by the percentage of the symbols that had to be decoded, which quantified the equivalent (or effective) number of iterations.

In Figure 5.13, as E_b/N_0 is increased from 3.0 dB to 3.5 dB for $L_{\min} = 51$, the percentage of non-decoded symbols for each iteration increases, reaching a maximum of 45% for the 6th iteration at 3.5 dB. When we further increase the $L_{\min}$ to 111 at 3.6 dB, the percentage of non-decoded symbols in iterations 2 to 5 decreases, while that in the 6th increases. This is because at this point there are two counter-acting forces:

1. An increased $L_{\min}$ would reduce the number of error-free blocks.
2. An increased E_b/N_0 would decrease the Hamming weight of the syndrome sequence and, therefore, increase the number of error-free blocks.

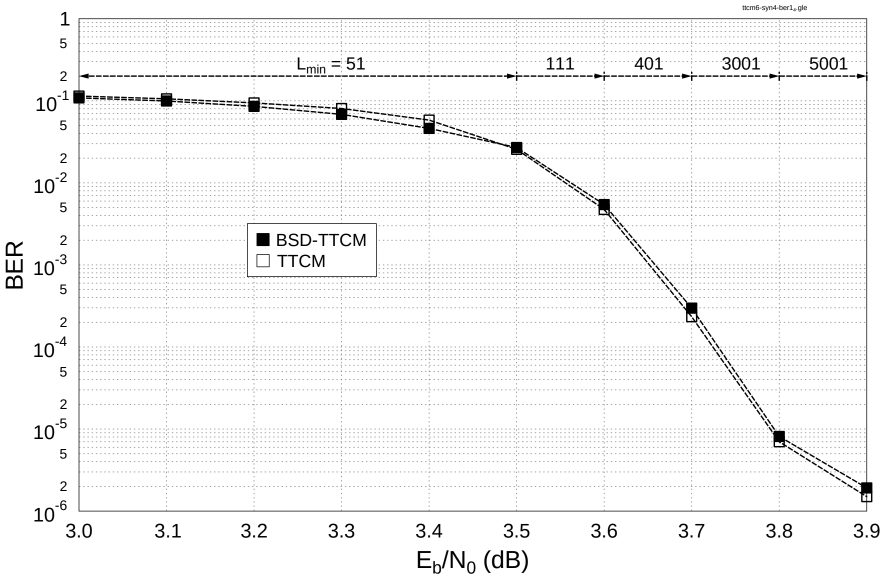


Figure 5.12: Comparison of the BER performance curve of BSD-TTCM with the conventional TTCM decoding over an AWGN channel. The corresponding TTCM parameters are summarized in Table 5.1, while the optimized $L_{\min}$ are listed in Table 5.2.

A similar trend is observed, when E_b/N_0 is increased further. Particularly, at high SNRs, at least a 20% complexity reduction is achieved for the 5th iteration and 45% for the 6th iteration.

Figure 5.14 quantifies the decoding complexity in terms of the equivalent number of decoding iterations. We may observe in Figure 5.14 that increasing the E_b/N_0 from 3.0 dB to 3.5 dB for $L_{\min} = 51$, reduces the number of effective iterations to a minimum of 4.8 at 3.5 dB. This is equivalent to a $(100 \times (6 - 4.8)/6) = 20\%$ reduction in the number of decoding iterations. Then, when $L_{\min}$ is increased to 111 at 3.6 dB, the number of equivalent iterations increases to 5. This corresponds to a reduction of $(100 \times (6 - 5)/6) \approx 17\%$ compared to the maximum of 6 iterations and it is therefore still significant. On average our proposed scheme reduces the effective number of iterations by at least one, i.e. by 17%, for high SNRs. We have also benchmarked the performance of our proposed BSD-TTCM decoder against the conventional hard-decision aided high-SNR Early Termination (ET) criterion of [136] in Figure 5.14. Our proposed scheme outperforms ET by at least 0.5 iteration at high SNRs. The complexity may be further reduced by increasing $L_{\min}$. However, as discussed before, this will incur a BER performance degradation.

5.5.2 Performance of BSD-TTCM over Uncorrelated Rayleigh Fading Channel

We have further investigated the performance of BSD-TTCM in the event of uncorrelated Rayleigh fading channel for the code parameters of Table 5.1. The corresponding optimum design parameter $L_{\min}$ for increasing SNRs in the high-SNR region (i.e. $E_b/N_0 \geq 6.2$) are listed in Table 5.3.

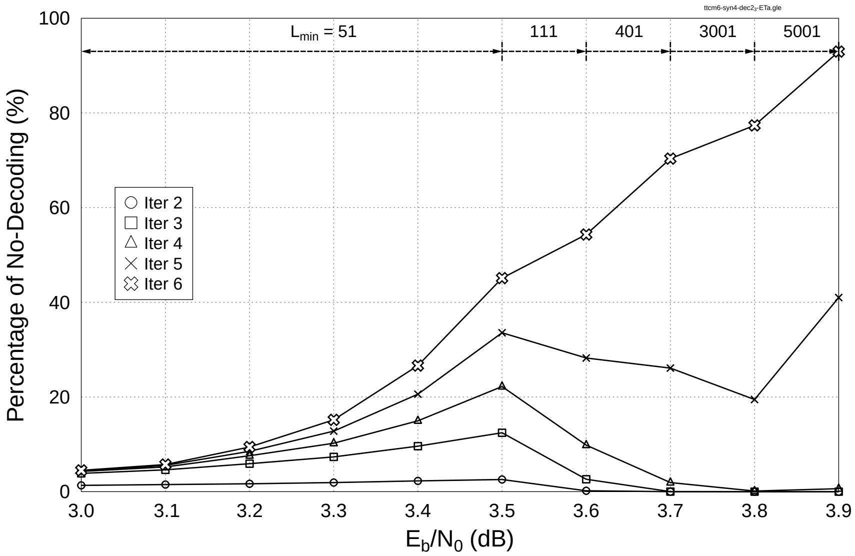


Figure 5.13: Reduction in decoding complexity of the BSD-TTCM of Figure 5.12, quantified in terms of the 'percentage of no-decoding'.

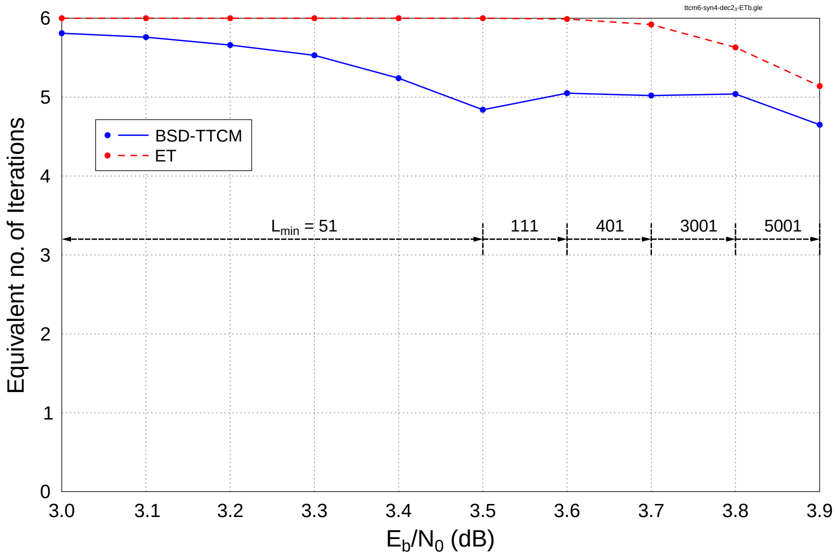


Figure 5.14: Reduction in decoding complexity of the BSD-TTCM of Figure 5.12, quantified in terms of the 'equivalent number of iterations'.

SNR Range	$L_{\min}$
$E_b/N_0 \leq 6.2$ dB	61
$6.2 < E_b/N_0 \leq 6.6$ dB	101
$6.6 < E_b/N_0 \leq 7.0$ dB	301
$7.0 < E_b/N_0 \leq 7.4$ dB	1201
$7.4 < E_b/N_0 \leq 7.8$ dB	4001

Table 5.3: Optimum $L_{\min}$ for the TTCM of Table 5.1 operating over an uncorrelated Rayleigh fading channel.

Figure 5.15 compares the resulting BER performance of the proposed BSD-TTCM with that of the conventional TTCM. As seen from Figure 5.15, syndrome-based blocking does not incur any significant BER degradation and that both decoding schemes exhibit a similar BER performance. The corresponding reduction in the decoding complexity is quantified in Figure 5.16 and Figure 5.17 in terms of the 'percentage of no-decoding' and 'equivalent number of iterations', respectively.

In Figure 5.16, as E_b/N_0 is increased from 5.0 dB to 6.2 dB for $L_{\min} = 61$, the percentage of non-decoded symbols for each iteration increases, reaching about 30% for the 6th iteration at 6.2 dB. As E_b/N_0 is increased further, the percentage of non-decoded symbols decreases during iterations 2 to 4, while it increases in iterations 5 and 6. This behaviour is similar to that observed in Figure 5.13. We may also notice in Figure 5.16 that, in the high-SNR region, at least a 20% complexity reduction is achieved for the 5th iteration and 30% for the 6th. By contrast, at least 20% and 45% complexity reduction was achieved for the 5th and the 6th iteration for transmission over the AWGN channel in Figure 5.13.

Figure 5.17 plots the corresponding decoding complexity in terms of the effective number of iterations. Increasing the E_b/N_0 from 5.0 dB to 6.2 dB for $L_{\min} = 61$ reduces the number of effective iterations to a minimum of 5.27 at 6.2 dB. This is equivalent to a $(100 \times (6 - 5.27)/6) \approx 12\%$ reduction in the number of decoding iterations. Thereafter the number of effective iterations more or less remain the same. Hence, BSD-TTCM yields a decoding complexity reduction of around 12% in the high-SNR region, when operating in the uncorrelated Rayleigh fading channel, which is slightly less than the 17% reduction observed for the AWGN channel in Figure 5.13. Based on these results, it is reasonable to conclude that the decoding complexity of BSD-TTCM is only slightly higher in an uncorrelated Rayleigh fading channel than in an AWGN channel, but the attainable complexity savings are still quite significant. In Figure 5.17, we have also benchmarked the performance of our proposed BSD-TTCM decoder against the conventional hard-decision aided ET criterion of [136]. We may observe in the Figure 5.17 that BSD-TTCM outperforms ET for all SNRs.

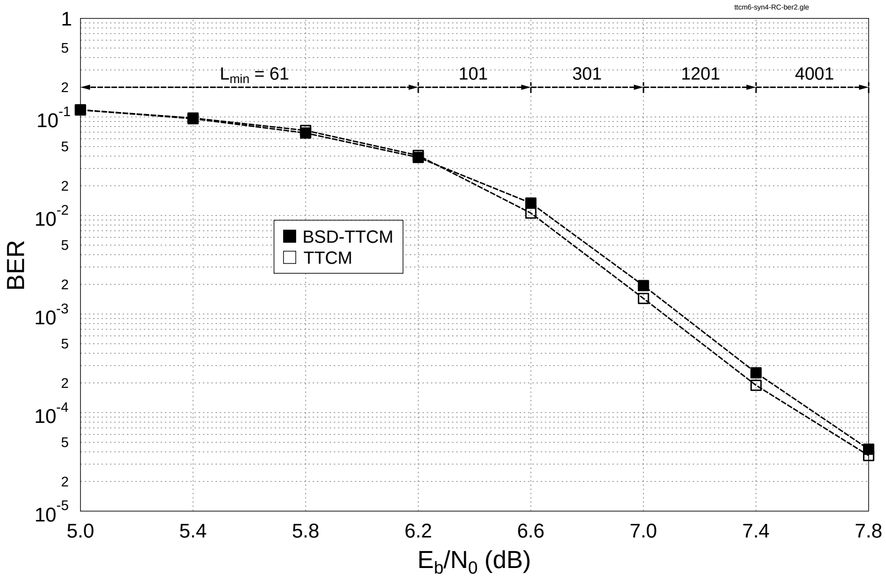


Figure 5.15: Comparison of the BER performance curve of BSD-TTCM with the conventional TTCM decoding over an uncorrelated Rayleigh fading channel. The corresponding TTCM parameters are summarized in Table 5.1, while the optimized $L_{\min}$ are listed in Table 5.3.

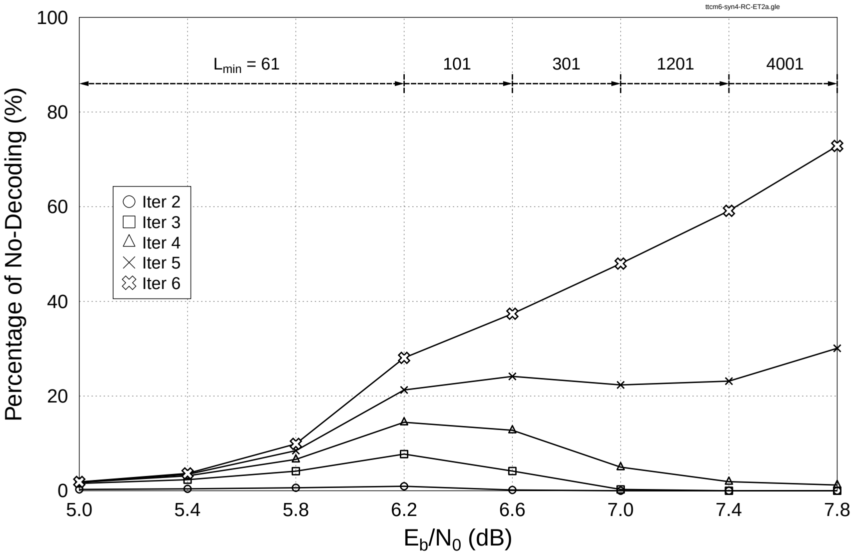


Figure 5.16: Reduction in decoding complexity of the BSD-TTCM of Figure 5.15, quantified in terms of the 'percentage of no-decoding'.

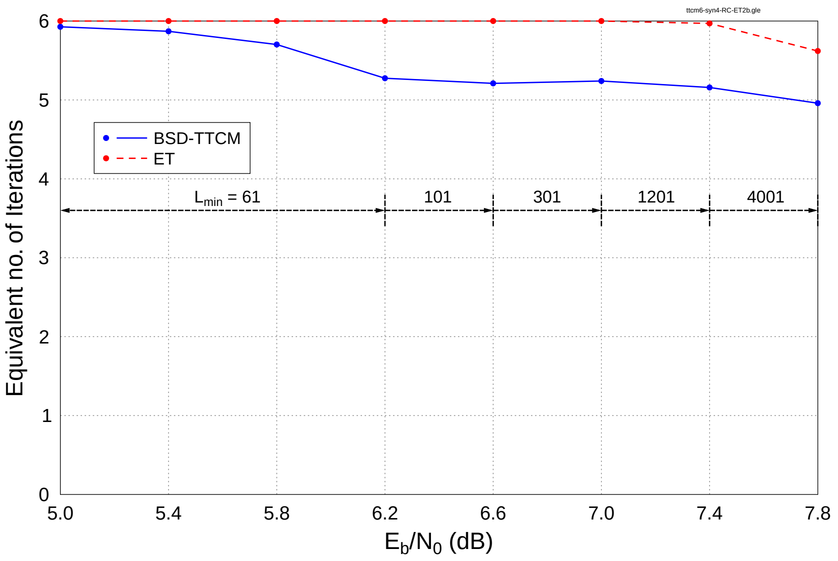


Figure 5.17: Reduction in decoding complexity of the BSD-TTCM of Figure 5.15, quantified in terms of the 'equivalent number of iterations'.

SNR Range	L_{min}
$E_b/N_0 \leq 6.6$ dB	61
$6.6 < E_b/N_0 \leq 7.0$ dB	201
$7.0 < E_b/N_0 \leq 7.4$ dB	281
$7.4 < E_b/N_0 \leq 7.8$ dB	321

Table 5.4: Optimum L_{min} for varying E_b/N_0 over uncorrelated Rayleigh fading channel using a frame of 500 symbols. Other TTCM parameters are same as that of Table 5.1.

5.5.3 Effect of Frame Length on the Performance of BSD-TTCM

The high-SNR ET scheme, which we have used in Section 5.5.1 and 5.5.2 as a benchmarker, offers higher reductions in the decoding complexity when shorter frames are used. Intuitively, shorter frames also improve the effective decoding complexity reduction of the BSD approach. Hence, BSD-TTCM is likely to outperform ET even for short frames. For the sake of substantiating our claim, we have analyzed the performance of the 8-state TTCM relying on 8PSK transmissions over an uncorrelated Rayleigh fading channel using a frame length of 500 TTCM-8PSK symbols and 6 iterations. The corresponding SNR-based L_{min} values are listed in Table 5.4.

The design parameter L_{min} of Table 5.4 is heuristically optimized for a particular E_b/N_0 range, while ensuring that the BER performance curve of the resultant BSD-TTCM scheme is the same as that of the conventional TTCM, which is demonstrated in Figure 5.18. The corresponding decoding

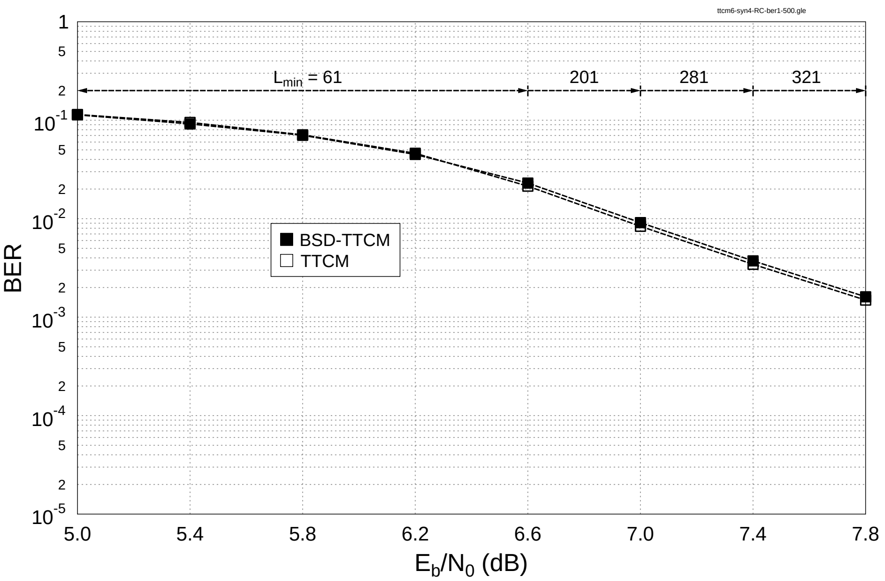


Figure 5.18: Comparison of the BER performance curve of BSD-TTCM with the conventional TTCM decoding over an uncorrelated Rayleigh fading channel using the TTCM parameters of Table 5.1 but with a reduced frame length of 500. The corresponding optimized $L_{\min}$ are listed in Table 5.3.

complexity is analyzed in Figure 5.19 and Figure 5.20. Reducing the frame length from $12,000$ to 500 symbols increases the percentage of non-decoded symbols at each iteration, as we can observed by comparing Figure 5.19 with Figure 5.13. Consequently, BSD-TTCM still out performs the ET technique, as shown in Figure 5.20.

5.6 Summary and Conclusions

Since quantum codes invoke the classical syndrome decoding based approach, as illustrated in Figure 5.1, in this chapter we discussed these decoding techniques operating over a classical channel. Unlike the widely used codeword decoding, which aims for identifying the most likely codeword, syndrome decoding characterizes the most likely channel error sequence. In this context, we commenced our discussions with the LUT-based syndrome decoding in Section 5.2, detailing the motivation behind the LUT-based syndrome decoding approach invoked for classical linear block codes. More specifically, it was demonstrated with the aid of a design example that the standard array-based codeword decoding imposes high storage requirements for long block codes. Fortunately, the same task may be achieved with the aid of an LUT-based decoder, which only requires a memory of size $2^{n-k} \times 2$, as depicted in Eq. (5.2), while the corresponding standard array of Eq. (5.1) has a size of $2^{n-k} \times 2^k$. We then proceeded with the construction of the error trellis of linear block codes and of convolutional codes in Sections 5.3.1 and 5.3.2, respectively. It was shown in Figure 5.6 and Figure 5.7 that the conventional code trellis and the syndrome-based error trellis are equivalent. More explicitly, each

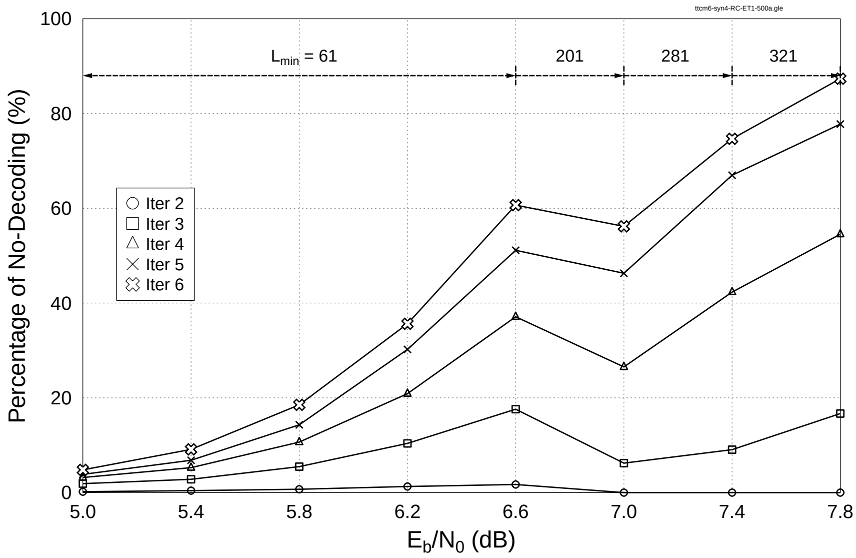


Figure 5.19: Reduction in decoding complexity of the BSD-TTCM of Figure 5.18, quantified in terms of the 'percentage of no-decoding'.

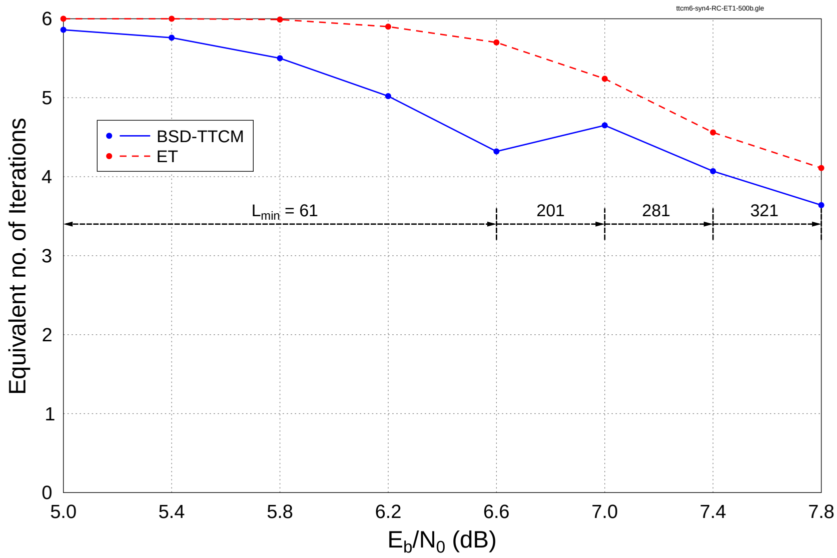


Figure 5.20: Reduction in decoding complexity of the BSD-TTCM of Figure 5.18, quantified in terms of the 'equivalent number of iterations'.

	SNR	No-Decoding			Reduction w.r.t.	Reference
Channel	Range	$\mathtt{I} = 5$	$\mathtt{I} = 6$	$\mathtt{I}_{\text{avg}}$	Early Termination	Figures
AWGN	≥ 3.5	$\geq 20\%$	$\geq 45\%$	≤ 5	≥ 0.5 iteration	Figure 5.13 and Figure 5.14
Rayleigh	≥ 6.2	$\geq 20\%$	$\geq 30\%$	≤ 5.3	≥ 0.5 iteration	Figure 5.16 and Figure 5.17

Table 5.5: Summary of the achieved decoding complexity reduction, when BSD-TTCM is invoked using the simulation parameters of Table 5.1. Decoding complexity is quantified in terms of the percentage of no-decoding as well as the equivalent number of iterations $\mathtt{I}_{\text{avg}}$. Complexity reduction is also compared with the high-SNR early termination technique.

path of a code trellis corresponds to a legitimate codeword, while each path of an error trellis is a legitimate error sequence for a given syndrome. When the syndrome is zero, the error trellis collapses to a code trellis. Finally, in Section 5.4, we laid out the reduced-complexity BSD approach. In particular, we conceived a syndrome-based block decoding approach for TTCM in Section 5.4.2. The proposed BSD-TTCM only decodes the blocks deemed to be erroneous, which are identified using the syndrome sequence, hence reducing the asscoiated decoding complexity. Furthermore, a pre-correction sequence is estimated at each iteration for reducing the decoding complexity of the forthcoming iterations. Finally, we evaluated the performance of our proposed BSD-TTCM for transmission over both an AWGN channel as well as an uncorrelated Rayleigh fading channel in Section 5.5.

In Section 5.5.1, we compared the BER performance of the proposed BSD-TTCM to that of the classic full-complexity TTCM decoder for transmission over an AWGN channel. It was demonstrated in Figure 5.12 that the design parameter L_{min} may be heuristically optimized upon increasing SNR values for the sake of achieving the same BER performance as that of a classic TTCM decoder. We further quantified the achieved complexity reductions in terms of the percentage of non-decoded blocks at each iteration index as well as the number of effective decoding iterations in Figure 5.13 and Figure 5.14, respectively. Quantitatively, we demonstrated in Figure 5.13 that a decoding complexity reduction of at least 17% is attained at high SNRs in terms of the effective number of iterations, with at least 20% and 45% reduction in the 5th and 6th iterations, respectively, for transmission over an AWGN channel. It was also shown in Figure 5.14 that our proposed reduced-complexity technique outperforms the ET scheme at all SNRs, yielding at least a reduction of 0.5 iteration in the high-SNR regime. These results are summarized in Table 5.5[8].

We further extended our analysis to the performance of BSD-TTCM over an uncorrelated Rayleigh fading channel in Section 5.5.2. Analogously to the AWGN channel, the design parameter L_{min} was optimized upon increasing SNR, so that the BSD-TTCM exhibits the same performance as that of a classic decoder, as evidenced in Figure 5.15. It was observed in Figure 5.16 and Figure 5.17 and that

[8] 'With respect to' is abbreviated as 'w.r.t.' in Table 5.5.

the BSD approach offers a slightly less reduction in complexity for transmission over the uncorrelated Rayleigh fading channel than that offered over the AWGN channel. More specifically, the decoding complexity reduction for transmission over a Rayleigh fading channel in Figure 5.16 was found to be about 12% in the high-SNR region, with at least 20% and 30% reduction in the 5th and 6th iterations, respectively. In Section 5.5.3, we furthermore evaluated the performance of our BSD-TTCM scheme for a short frame length of 500 symbols. It was observed in Figure 5.20 that BSD-TTCM outperforms the frequently used ET technique, regardless of the frame length. These results are also tabulated in Table 5.5.

The classical-quantum communication system, which we conceived in Chapter 3, supports only the transmission of classical information. By contrast, quantum communication systems may transmit classical as well as quantum information. This in turn requires efficient Quantum Error Correction Codes (QECCs). QECCs are also vital for reliable quantum computation. In the next chapter, we will proceed with the design of QECCs, which is based on the classical to quantum transition of Chapter 4 and the classical syndrome decoding approach of Chapter 5.

Chapter 6

EXIT-Chart Aided Hashing Bound Approaching Concatenated Quantum Codes

6.1 Introduction

In Chapter 3, we conceived a classical-quantum communication system, relying on near-capacity classical code designs. The designs of Chapter 3 are restricted to the transmission of classical information. By contrast, a more general quantum communication system may transmit both classical as well as quantum information, which in turn requires efficient Quantum Error Correction Codes (QECCs). QECCs are also indispensable for reliable quantum computation. In this spirit, in Chapter 4, we presented the methodology of constructing QECCs from known classical codes, while the associated classical syndrome decoding techniques were discussed in Chapter 5. Pursuing further the design of QECCs, in this chapter we aim for designing Hashing bound approaching QECCs, based on the foundation laid down in Chapters 4 and 5.

Similar to the family of classical error correction codes [124, 68], which aim for operating close to Shannon's capacity limit, QECCs are designed to approach the quantum capacity [148, 149, 150], or more specifically the Hashing bound, which is a lower bound of the achievable quantum capacity. A significant amount of work has been carried out over the last few decades to achieve this objective. However, the field of QECCs is still not as mature as that of their classical counterparts. Recently, inspired by the family of classical near-capacity concatenated codes, which rely on iterative decoding schemes, e.g. [151, 116], substantial efforts have been invested in [55, 88] to construct comparable concatenated quantum codes, which we refer to as serially concatenated Quantum Turbo Codes (QTCs). In this context, the search for the optimal components of a QTC has been so far confined to the analysis of the distance spectra of the constituent Quantum Convolutional Codes (QCCs), followed by intensive Monte-Carlo simulations for determining the convergence threshold of the resultant QTC, as detailed in [55, 88]. While the distance spectrum dominates a turbo code's performance in the Bit Error Rate (BER) floor region, it has a relatively insignificant impact on the convergence properties in the low Signal-to-Noise Ratio (SNR) turbo-cliff region [116]. Therefore, having a good distance spectrum does not guarantee having a near-capacity performance - in fact, often there is a trade-off between them. To circumvent this problem and to dispense with time-consuming Monte-Carlo simulations, we extend the application of EXtrinsic Information Transfer (EXIT) charts to the design of QTCs. More specifically, in the light of the increasing interest in conceiving Hashing bound approaching concatenated quantum code design principles, the novel contributions of this chapter are [4, 3]:

- We have appropriately adapted the conventional non-binary EXIT chart based design approach to the family of QTCs based on the underlying quantum-to-classical isomorphism of Chapter 4. We have analyzed the behaviour of both an unassisted (non-recursive) and of an entanglement-assisted (recursive) inner convolutional code using EXIT charts for demonstrating that, similar to their classical counterparts, recursive inner quantum codes constitute families of QTCs having an unbounded minimum distance[1]. We have further optimized the constituent inner and outer components of the concatenated structure using EXIT charts for the sake of approaching the Hashing bound.

- We have conceived a generically applicable structure for Quantum Irregular Convolutional Codes (QIRCCs), which can be dynamically adapted to a specific application scenario for the sake of facilitating a Hashing bound approaching performance. We also provide a detailed design example by constructing a 10-subcode QIRCC and use it as an outer code in a concatenated structure for evaluating its performance.

This chapter is organized as follows. We commence by outlining our design objectives in Section 6.2. We then present the circuit-based representation of QCCs in Section 6.3, which is essential for understanding the iterative decoding procedure employed by concatenated quantum codes. Then, the system model of a quantum communication system relying on concatenated quantum codes is laid out in Section 6.4.1, followed by the associated degenerate decoding algorithm in Section 6.4.2. We present our proposed EXIT-chart aided near-capacity quantum code design in Section 6.5, followed by our simulation results in Section 6.6. We then proceed with our proposed QIRCC in Section 6.7, whose performance is recorded in Section 6.8. Finally, our conclusions are offered in Section 6.9.

6.2 Design Objectives

An ideal code $\mathcal{C}$ designed for a quantum depolarizing channel may be characterized in terms of the channel's depolarizing probability p and its coding rate R_Q. Here the coding rate R_Q is measured

[1] The unbounded minimum distance of a code implies that its minimum distance increases almost linearly with the interleaver length.

in terms of the number of information qubits transmitted per channel use, i.e. we have $R_Q = k/n$, where k and n are the lengths of the information word and codeword, respectively. Analogously to Shannon's classical capacity, the relationship between p and R_Q for the depolarizing channel is defined by the Hashing bound, which sets a lower limit on the achievable quantum capacity[2]. The Hashing bound is given by [37, 88]:

$$\mathrm{C}_Q(p) = 1 - \mathrm{H}_2(p) - p\log_2(3), \tag{6.1}$$

where $\mathrm{H}_2(p)$ is the binary entropy function. More explicitly, for a given p, if a random code $\mathcal{C}$ of a sufficiently long codeword-length is chosen such that its coding rate obeys $R_Q \leq \mathrm{C}_Q(p)$, then $\mathcal{C}$ may yield an infinitesimally low QuBit Error Rate (QBER) for a depolarizing probability of p. It must be noted here that intuitively a low QBER corresponds to a high fidelity between the transmitted and the decoded quantum state. More explicitly, for a given value of p, $\mathrm{C}_Q(p)$ gives the Hashing limit on the coding rate. Alternatively, for a given coding rate R_Q, where we have $R_Q = \mathrm{C}_Q(p^*)$, p^* gives the Hashing limit on the channel's depolarizing probability. In duality to the classical domain, this may also be referred to as the noise limit. An ideal quantum code should be capable of ensuring reliable transmission close to the noise limit p^*. Furthermore, for any arbitrary depolarizing probability p, its discrepancy with respect to the noise limit p^* may be computed in decibels (dB) as follows [88]:

$$\text{Distance from Hashing bound} \triangleq 10 \times \log_{10}\left(\frac{p^*}{p}\right). \tag{6.2}$$

Consequently, our quantum code design objective is to minimize the discrepancy with respect to the Hashing bound, thereby yielding a Hashing bound approaching code design.

It is pertinent to recall here the Entanglement-Assisted (EA) regime discussed in Section 4.5, where the EA code $\mathcal{C}$ is characterized by an additional parameter c. Here c is the number of entangled qubits pre-shared between the transmitter and the receiver, thus leading to the terminology of being entanglement-assisted[3]. It is assumed furthermore that these pre-shared entangled qubits are transmitted over a noiseless quantum channel. The resultant EA Hashing bound is given by [88, 154]:

$$\mathrm{C}_Q(p) = 1 - \mathrm{H}_2(p) - p\log_2(3) + \mathtt{E}, \tag{6.3}$$

where the so-called entanglement consumption rate is $\mathtt{E} = \frac{c}{n}$. Furthermore, the value of c may be varied from 0 to a maximum of $(n-k)$. For the family of maximally entangled codes associated with $c = (n-k)$, the EA Hashing bound of Eq. (6.3) is reduced to [88, 154]:

$$\mathrm{C}_Q(p) = 1 - \frac{\mathrm{H}_2(p) - p\log_2(3)}{2}. \tag{6.4}$$

Therefore, the resultant Hashing region of the EA communication is bounded by Eq. (6.1) and Eq. (6.4), which is also illustrated in Figure 6.1. To elaborate a little further, let us assume that

[2] Recall from Section 4.3.3 that quantum codes are inherently degenerate in nature because different errors may have the same impact on the quantum state. Due to this degenerate nature, the ultimate capacity of a quantum channel can be higher than that defined by the Hashing bound [152, 153]. However, none of the codes known to date outperform the Hashing bound at practically feasible frame lengths.

[3] A quantum code without pre-shared entanglement, i.e. $c = 0$, may be termed as an unassisted quantum code.

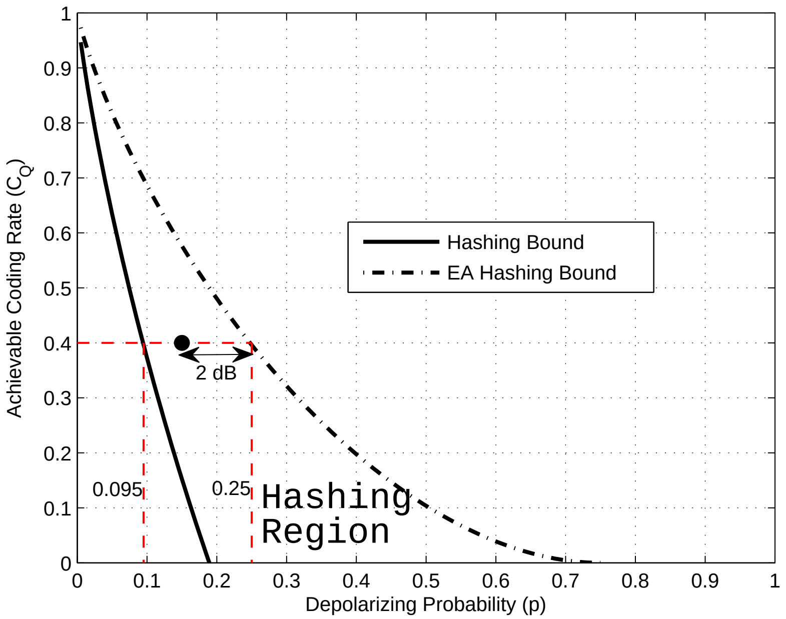


Figure 6.1: Unassisted and EA Hashing bounds characterized by Eq. (6.1) and Eq. (6.4), respectively, for the quantum depolarizing channel. The region enclosed by these two bounds, which is labeled the Hashing region, defines the capacity for varying number of pre-shared entangled qubits (c). At $R_Q = 0.4$, the unassisted Hashing bound gives a noise limit of $p^* = 0.095$, while the maximally entangled Hashing bound increases the limit to $p^* = 0.25$. The circle represents a maximally entangled code with $R_Q = 0.4$, which ensures reliable transmission for $p \leq 0.15$, thus operating at a distance of 2 dB from the noise limit.

the desired coding rate is $R_Q = 0.4$. Then, as gleaned from Figure 6.1, the noise limit for the 'unassisted' quantum code is around $p^* = 0.095$, which increases to around $p^* = 0.25$ with the aid of maximum entanglement, i.e. we have $\mathtt{E} = 1 - R_Q = 0.6$. Furthermore, $0 < \mathtt{E} < 0.6$ will result in bearing noise limits in the range of $0.095 < p^* < 0.25$. Let us assume furthermore that we design a maximally entangled code $\mathcal{C}$ for $R_Q = 0.4$, so that it ensures reliable transmission for $p \leq 0.15$. Based on Eq. (6.2), the performance of this code (marked with a circle in Figure 6.1) is around $[10 \times \log 10(\frac{0.25}{0.15})] = 2$ dB away from the noise limit. We may approach the noise limit more closely by optimizing a range of conflicting design challenges, which are illustrated in the stylized representation of Figure 6.2. For example, we may achieve a lower QBER by increasing the code length. However, this in turn incurs longer delays. Alternatively, we may resort to more complex code designs for reducing the QBER, which may also be reduced by employing codes having lower coding rates or higher entanglement consumption rates, thus requiring more transmitted qubits or entangled qubits. Hence striking an appropriate compromise, which meets these conflicting design challenges, is required.

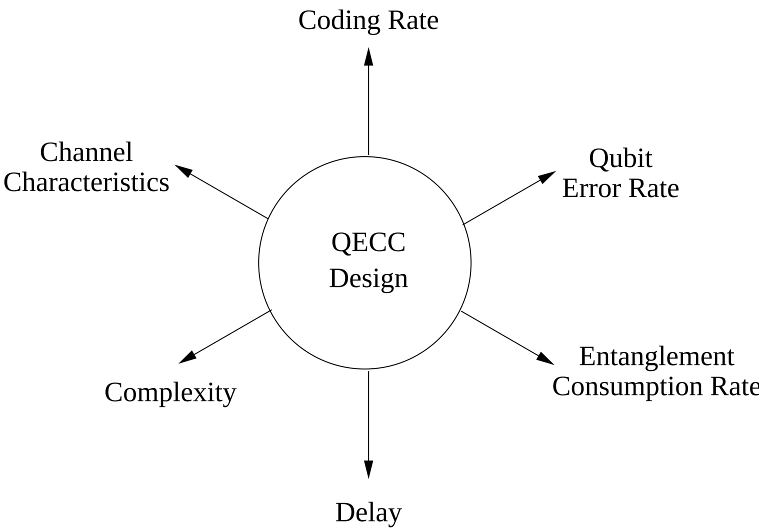


Figure 6.2: Stylized illustration of the conflicting design challenges, which are involved in the design of quantum codes.

6.3 Circuit-Based Representation of Stabilizer Codes

Circuit-based representation of quantum codes facilitates the design of concatenated code structures. More specifically, for decoding concatenated quantum codes it is more convenient to exploit the circuit-based representation of the constituent codes, rather than the conventional trellis-based syndrome decoding. Therefore, in this section, we will review the circuit-based representation of quantum codes. This discussion is based on [55].

Let us recall from Section 5.3.1 that an (n, k) classical linear block code constructed over the code space C maps the information word $x \in \mathbb{F}_2^k$ onto the corresponding codeword $\overline{x} \in \mathbb{F}_2^n$. In the circuit-based representation, this encoding procedure can be encapsulated as follows:

$$C = \{\overline{x} = (x : 0_{n-k})\, V\}, \tag{6.5}$$

where V is an $(n \times n)$-element invertible encoding matrix over $\mathbb{F}_2$ and 0_{n-k} is an $(n-k)$-bit vector initialized to 0. Furthermore, given the generator matrix G and the Parity Check Matrix (PCM) H, the encoding matrix V may be specified as:

$$V = \begin{pmatrix} G \\ \left(H^{-1}\right)^T \end{pmatrix}, \tag{6.6}$$

and its inverse is given by:

$$V^{-1} = \begin{pmatrix} G^{-1} & H^T \end{pmatrix}. \tag{6.7}$$

The encoding matrix V specifies both the code space as well as the encoding operation, while its inverse V^{-1} specifies the error syndrome. More specifically, let $y = \overline{x} + e$ be the received codeword,

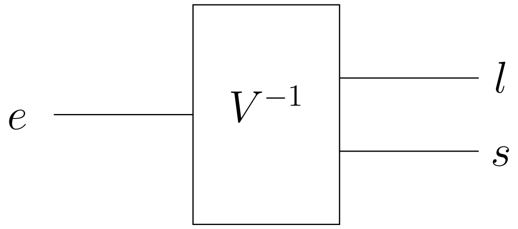


Figure 6.3: Circuit representation of the inverse encoder $eV^{-1} = (l : s)$.

where e is the n-bit error incurred during transmission. Then, passing the received codeword y through the inverse encoder V^{-1} yields:

$$yV^{-1} = (\tilde{x} : s)\,, \tag{6.8}$$

where $\tilde{x} = x + l$ for the logical error $l \in \mathbb{F}_2^k$ inflicted on the information word x and $s \in \mathbb{F}_2^{n-k}$ is the syndrome, which is equivalent to yH^T. Eq. (6.8) may be further decomposed to:

$$\begin{aligned}(\overline{x} + e)\, V^{-1} &= (x + l : s)\,,\\ \overline{x}V^{-1} + eV^{-1} &= (x : 0_{n-k}) + (l : s)\,,\end{aligned} \tag{6.9}$$

which is a linear superposition of the inverse of Eq. (6.5) and $eV^{-1} = (l : s)$. Hence, the inverse encoder V^{-1} decomposes the channel error e into the logical error l and error syndrome s, which is also depicted in Figure 6.3.

Analogously to Eq. (6.5), the unitary encoding operation $\mathcal{V}$ of an $[n, k]$ QSC, constructed over a code space $\mathcal{C}$, which maps the information word (logical qubits) $|\psi\rangle \in \mathbb{C}^{2^k}$ onto the codeword (physical qubits) $|\overline{\psi}\rangle \in \mathbb{C}^{2^n}$, may be mathematically encapsulated as follows:

$$\mathcal{C} = \{|\overline{\psi}\rangle = \mathcal{V}(|\psi\rangle \otimes |0_{n-k}\rangle)\}, \tag{6.10}$$

where $|0_{n-k}\rangle$ are $(n-k)$ auxiliary qubits initialized to the state $|0\rangle$. The unitary encoder $\mathcal{V}$ of Eq. (6.10) carries out an n-qubit Clifford transformation, which maps an n-qubit Pauli group $\mathcal{G}_n$ onto itself under conjugation [155], i.e. we have:

$$\mathcal{V}\mathcal{G}_n\mathcal{V}^\dagger = \mathcal{G}_n. \tag{6.11}$$

In other word, a Clifford operation preserves the elements of the Pauli group under conjugation such that for $\mathcal{P} \in \mathcal{G}_n$, $\mathcal{V}\mathcal{P}\mathcal{V}^\dagger \in \mathcal{G}_n$. Furthermore, any Clifford unitary matrix is completely specified by a combination of Hadamard (**H**) gates, phase (**S**) gates and controlled-NOT (C-NOT) gates, which are defined as follows[4] [18]:

$$\mathbf{H} = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}, \ \mathbf{S} = \begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix},$$

$$\text{C-NOT} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}. \tag{6.12}$$

Hadamard gate preserves the elements of a single-qubit Pauli group $\mathcal{G}_1$ under conjugation as follows:

$$\begin{aligned} \mathbf{X} &\rightarrow \mathbf{HXH}^\dagger = \mathbf{Z}, \\ \mathbf{Z} &\rightarrow \mathbf{HZH}^\dagger = \mathbf{X}, \\ \mathbf{Y} &\rightarrow \mathbf{HYH}^\dagger = -\mathbf{Y}, \end{aligned} \tag{6.13}$$

while phase gate preserves them under conjugation as:

$$\begin{aligned} \mathbf{X} &\rightarrow \mathbf{SXS}^\dagger = \mathbf{Y}, \\ \mathbf{Z} &\rightarrow \mathbf{SZS}^\dagger = \mathbf{Z}, \\ \mathbf{Y} &\rightarrow \mathbf{SYS}^\dagger = -\mathbf{X}, \end{aligned} \tag{6.14}$$

The C-NOT gate of Eq. (6.12) is a 2-qubit gate, where the first qubit is the control qubit, while the second is the target. Since C-NOT is a 2-qubit gate, it acts on the elements of $\mathcal{G}_2$, transforming the standard basis of $\mathcal{G}_2$ under conjugation as given below:

$$\begin{aligned} \mathbf{X}\otimes\mathbf{I} &\rightarrow \mathbf{X}\otimes\mathbf{X}, \\ \mathbf{I}\otimes\mathbf{X} &\rightarrow \mathbf{I}\otimes\mathbf{X}, \\ \mathbf{Z}\otimes\mathbf{I} &\rightarrow \mathbf{Z}\otimes\mathbf{I}, \\ \mathbf{I}\otimes\mathbf{Z} &\rightarrow \mathbf{Z}\otimes\mathbf{Z}. \end{aligned} \tag{6.15}$$

Let us further emphasize on the significance of Clifford encoding operation. *Since $\mathcal{V}$ belongs to the Clifford group, it preserves the elements of the stabilizer group $\mathcal{H}$ under conjugation.* If g_i' is the ith stabilizer of the unencoded state $|\psi\rangle$, then this may be proved as follows:

$$|\psi\rangle \otimes |0_{n-k}\rangle = g_i' \left(|\psi\rangle \otimes |0_{n-k}\rangle\right). \tag{6.16}$$

Encoding $|\psi\rangle$ with $\mathcal{V}$ yields:

$$\mathcal{V}\left(|\psi\rangle \otimes |0_{n-k}\rangle\right) = \mathcal{V}\left(g_i' \left(|\psi\rangle \otimes |0_{n-k}\rangle\right)\right), \tag{6.17}$$

which is equivalent to:

$$\mathcal{V}\left(|\psi\rangle \otimes |0_{n-k}\rangle\right) = \mathcal{V}\left(g_i'\mathcal{V}^\dagger\mathcal{V}\left(|\psi\rangle \otimes |0_{n-k}\rangle\right)\right), \tag{6.18}$$

[4] The standard gates invoked here may not be optimum. There is no evidence in literature pertaining to the optimality of these gates.

since $\mathcal{V}^\dagger\mathcal{V} = \mathbb{I}_n$. Substituting Eq. (6.10) into Eq. (6.18) gives:

$$|\overline{\psi}\rangle = \left(\mathcal{V}g_i'\mathcal{V}^\dagger\right)|\overline{\psi}\rangle. \tag{6.19}$$

Hence, the encoded state $|\overline{\psi}\rangle$ is stabilized by $g_i = \mathcal{V}g_i'\mathcal{V}^\dagger$. From this it appears as if any arbitrary $\mathcal{V}$ (not necessarily Clifford) can be used to preserve the stabilizer subspace, which is not true. Since we assume that the stabilizer group $\mathcal{H}$ is a subgroup of the Pauli group, we impose the additional constraint that $\mathcal{V}$ must yield the elements of Pauli group under conjugation as in Eq. (6.11), which is only true for Clifford operations.

Furthermore, *the Clifford encoding operation also preserves the commutativity relation of stabilizers.* Let g_i' and g_j' be a pair of unencoded stabilizers. Then the above statement can be proved as follows:

$$g_ig_j = \left(\mathcal{V}g_i'\mathcal{V}^\dagger\right)\left(\mathcal{V}g_j'\mathcal{V}^\dagger\right) = \mathcal{V}g_i'g_j'\mathcal{V}^\dagger. \tag{6.20}$$

Since g_i' and g_j' commute, we have:

$$\mathcal{V}g_i'g_j'\mathcal{V}^\dagger = \mathcal{V}g_j'g_i'\mathcal{V}^\dagger. \tag{6.21}$$

Using $\mathcal{V}^\dagger\mathcal{V} = \mathbb{I}_n$, gives:

$$\mathcal{V}g_j'\mathcal{V}^\dagger\mathcal{V}g_i'\mathcal{V}^\dagger. = g_jg_i. \tag{6.22}$$

Since the n-qubit Pauli group forms a basis for the $(2^n \times 2^n)$-element matrices of Eq. (6.12), the Clifford encoder $\mathcal{V}$, which acts on the 2^n-dimensional Hilbert space, can be completely defined by specifying its action under conjugation on the Pauli-$\mathbf{X}$ and $\mathbf{Z}$ operators acting on each of the n qubits, as seen in Eq. (6.13) to (6.15). However, $\mathcal{V}$ and $\mathcal{V}'$, which differ only through a global phase such that $\mathcal{V}' = e^{j\theta}\mathcal{V}$, have the same impact under conjugation. Therefore, global phase has no physical significance in the context of Eq. (6.11) and the n-qubit encoder $\mathcal{V}$ can be completely specified by its action on the binary equivalent of the Pauli operators. More specifically, the n-qubit encoder $\mathcal{V}$ maps $\mathcal{P} \in \mathcal{G}_n$ to $\mathcal{P}' \in \mathcal{G}_n$ as follows:

$$\mathcal{P}' = \mathcal{V}\mathcal{P}\mathcal{V}^\dagger, \tag{6.23}$$

which is equivalent to mapping the equivalent $2n$-bit vector $P = [\mathcal{P}]$ onto $P' = [\mathcal{P}']$, where $[.]$ denotes the effective Pauli group G_n such that $P = [\mathcal{P}]$ differs from $\mathcal{P}$ by a multiplicative constant, i.e. we have $P = \mathcal{P}/\{\pm 1, \pm i\}$, and the elements of G_n are represented by $2n$-tuple binary vectors based on the mapping given in Eq. (4.29). Consequently, Eq. (6.23) can be re-written as:

$$P' = [\mathcal{P}'] = [\mathcal{V}\mathcal{P}\mathcal{V}^\dagger]. \tag{6.24}$$

Since the effective mapping of Eq. (6.24) is binary, i.e. $P \to P'$, we may characterize it in terms of an equivalent $(2n \times 2n)$-element binary matrix V, which is defined as follows:

$$P' = [\mathcal{V}\mathcal{P}\mathcal{V}^\dagger] = [\mathcal{P}]V = PV. \tag{6.25}$$

Furthermore, since the Clifford transformation $\mathcal{V}$ preserves the commutativity relationship, the associated binary matrix V is always a symplectic matrix [55]. As a consequence of this equivalence, any

Clifford unitary can be efficiently simulated on a classical system as stated by the Gottesman-Knill theorem [156].

We next define $\mathcal{V}$ by specifying its action on the elements of the Pauli group $\mathcal{G}_n$. More precisely, we consider $2n$ n-qubit unencoded operators $Z_i, X_i, \ldots, Z_n, X_n$, where Z_i and X_i represents the Pauli $\mathbf{Z}$ and $\mathbf{X}$ operator, respectively, acting on the ith qubit and the identity $\mathbf{I}$ on all other qubits. The unencoded operators $Z_{k+1}, \ldots, Z_n$ stabilizes the unencoded state of Eq. (6.10), i.e. $(|\psi\rangle \otimes |0_{n-k}\rangle)$, and are therefore called the unencoded stabilizer generators. On the other hand, $X_{k+1}, \ldots, X_n$ are the unencoded pure errors since X_i anti-commutes with the corresponding unencoded stabilizer generator Z_i, yielding an error syndrome of 1. Furthermore, the unencoded logical operators acting on the information qubits are $Z_i, X_i, \ldots, Z_k, X_k$, which commute with the unencoded stabilizers $Z_{k+1}, \ldots, Z_n$. The encoder $\mathcal{V}$ maps the unencoded operators $Z_i, X_i, \ldots, Z_n, X_n$ onto the encoded operators $\overline{Z}_i, \overline{X}_i, \ldots, \overline{Z}_n, \overline{X}_n$, which may be represented as follows:

$$\overline{X}_i = \left[\mathcal{V} X_i \mathcal{V}^\dagger\right] = [X_i]\, V, \qquad \overline{Z}_i = \left[\mathcal{V} Z_i \mathcal{V}^\dagger\right] = [Z_i]\, V. \tag{6.26}$$

Since Clifford transformations do not perturb the commutativity relation of the operators, the resultant encoded stabilizers $\overline{Z}_{k+1}, \ldots, \overline{Z}_n$ are equivalent to the stabilizers g_i of Eq. (4.14), while $\overline{X}_{k+1}, \ldots, \overline{X}_n$ are the pure errors t_i of the resultant stabilizer code, which trigger a non-trivial syndrome. Moreover, $\overline{Z}_i, \overline{X}_i, \ldots, \overline{Z}_k, \overline{X}_k$ are the encoded logical operators, which commute with the stabilizers g_i. Logical operators merely map one codeword onto the other, without affecting the codespace $\mathcal{C}$ of the stabilizer code. It also has to be mentioned here that the stabilizer generators g_i together with the encoded logical operations constitute the normalizer of the stabilizer code. The $(2n \times 2n)$-element binary symplectic encoding matrix V is therefore given by:

$$V = \begin{pmatrix} \overline{Z}_1 \\ \vdots \\ \overline{Z}_k \\ \overline{Z}_{k+1} \\ \vdots \\ \overline{Z}_n \\ \overline{X}_1 \\ \vdots \\ \overline{X}_k \\ \overline{X}_{k+1} \\ \vdots \\ \overline{X}_n \end{pmatrix} = \begin{pmatrix} \overline{Z}_1 \\ \vdots \\ \overline{Z}_k \\ g_1 \\ \vdots \\ g_{n-k} \\ \overline{X}_1 \\ \vdots \\ \overline{X}_k \\ t_1 \\ \vdots \\ t_{n-k} \end{pmatrix}, \tag{6.27}$$

where the Pauli $\mathbf{Z}$ and $\mathbf{X}$ operators are mapped onto the classical bits using the Pauli-to-binary isomorphism of Section 4.3.2.1.

Analogously to the classical inverse encoder of Eq. (6.8), the inverse encoder of a quantum code is the Hermitian conjugate $\mathcal{V}^\dagger$. Let $|\hat{\psi}\rangle = \mathcal{P}|\overline{\psi}\rangle$ be the received codeword such that $\mathcal{P}$ is the n-qubit

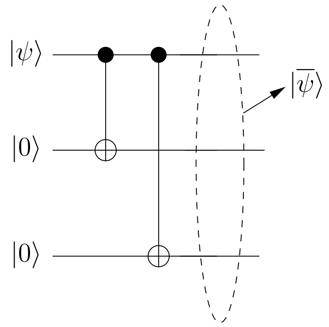


Figure 6.4: Encoding Circuit for 3-qubit bit-flip repetition code.

channel error. Then, passing the received codeword $|\hat{\psi}\rangle$ through the inverse encoder $\mathcal{V}^\dagger$ yields:

$$\begin{aligned}\mathcal{V}^\dagger\mathcal{P}|\overline{\psi}\rangle &= \mathcal{V}^\dagger\mathcal{P}\mathcal{V}(|\psi\rangle \otimes |0_{(n-k)}\rangle) \\ &= (\mathcal{L}|\psi\rangle) \otimes (\mathcal{S}|0_{(n-k)}\rangle),\end{aligned} \tag{6.28}$$

where $\mathcal{V}^\dagger\mathcal{P}\mathcal{V} \equiv (\mathcal{L} \otimes \mathcal{S})$ and $\mathcal{L} \in \mathcal{G}_k$ denotes the error imposed on the information word, while $\mathcal{S} \in \mathcal{G}_{n-k}$ represents the error inflicted on the remaining $(n-k)$ auxiliary qubits. In the equivalent binary representation, Eq. (6.28) may be modeled as follows:

$$PV^{-1} = (L : S)\,, \tag{6.29}$$

where we have $P = [\mathcal{P}]$, $L = [\mathcal{L}]$ and $S = [\mathcal{S}]$.

Let us now derive the encoding matrix V for the 3-qubit bit-flip repetition code $C(3,1)$. Recall from Section 4.3.2.1 that the corresponding binary PCM H is given by:

$$H = \left(\begin{array}{ccc|ccc} 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 \end{array}\right), \tag{6.30}$$

and the circuit of Figure 6.4 constitutes the encoder. Its unencoded operators are as follows:

$$\begin{pmatrix} Z_1 \\ Z_2 \\ Z_3 \\ X_1 \\ X_2 \\ X_3 \end{pmatrix} = \begin{pmatrix} \mathbf{ZII} \\ \mathbf{IZI} \\ \mathbf{IIZ} \\ \mathbf{XII} \\ \mathbf{IXI} \\ \mathbf{IIX} \end{pmatrix} \equiv \left(\begin{array}{ccc|ccc} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{array}\right). \tag{6.31}$$

A C-NOT gate is then applied to the second qubit, which is controlled by the first. As seen in Eq. (6.15), the C-NOT gate copies Pauli $\mathbf{X}$ operator forward from the control qubit to the target qubit, while **Z** is copied in the opposite direction. Therefore, we get:

$$\begin{pmatrix}\mathbf{ZII}\\\mathbf{IZI}\\\mathbf{IIZ}\\\mathbf{XII}\\\mathbf{IXI}\\\mathbf{IIX}\end{pmatrix}\xrightarrow{\text{C-NOT}(1,2)}\begin{pmatrix}\mathbf{ZII}\\\mathbf{ZZI}\\\mathbf{IIZ}\\\mathbf{XXI}\\\mathbf{IXI}\\\mathbf{IIX}\end{pmatrix}\equiv\left(\begin{array}{ccc|ccc}1&0&0&0&0&0\\1&1&0&0&0&0\\0&0&1&0&0&0\\\hline 0&0&0&1&1&0\\0&0&0&0&1&0\\0&0&0&0&0&1\end{array}\right). \tag{6.32}$$

Another C-NOT gate is then applied to the third qubit, which is also controlled by the first, yielding:

$$\begin{aligned}\begin{pmatrix}\mathbf{ZII}\\\mathbf{ZZI}\\\mathbf{IIZ}\\\mathbf{XXI}\\\mathbf{IXI}\\\mathbf{IIX}\end{pmatrix}\xrightarrow{\text{C-NOT}(1,3)}\begin{pmatrix}\mathbf{ZII}\\\mathbf{ZZI}\\\mathbf{ZIZ}\\\mathbf{XXX}\\\mathbf{IXI}\\\mathbf{IIX}\end{pmatrix}&\equiv\left(\begin{array}{ccc|ccc}1&0&0&0&0&0\\1&1&0&0&0&0\\1&0&1&0&0&0\\\hline 0&0&0&1&1&1\\0&0&0&0&1&0\\0&0&0&0&0&1\end{array}\right)\\&=V.\end{aligned} \tag{6.33}$$

We can also express the binary matrix V of Eq. (6.33) in decimal notation as:

$$V \equiv \{32, 48, 40, 7, 2, 1\}_{10}\ , \tag{6.34}$$

where each index corresponds to a row of V, e.g. the first row (1 0 0 0 0 0) is equivalent to the decimal number 32.

As gleaned from Eq. (6.33), the stabilizer generators of the 3-qubit bit-flip repetition code are $g_1 = \mathbf{ZZI}$ and $g_2 = \mathbf{ZIZ}$ according to Eq. (6.27), where $k = 1$ and $n = 3$. More explicitly, rows 2 and 3 of V constitute the PCM H of Eq. (6.30). The encoded logical operators are $\overline{Z}_1 = \mathbf{ZII}$ and $\overline{X}_1 = \mathbf{XXX}$, which commute with the stabilizers g_1 and g_2. Finally, the pure errors are $t_1 = \mathbf{IXI}$ and $t_2 = \mathbf{IIX}$, which anti-commute with g_1 and g_2, respectively, yielding a non-trivial syndrome.

Based on the above discussion, we now proceed to lay out the circuit-based model for a convolutional code, which is given in [55]. As discussed in Section 4.4, convolutional codes are equivalent to linear block codes associated with semi-infinite block lengths. More specifically, as illustrated in Figure 4.10, the PCM H of a $C(n, k, m)$ convolutional code has a block-band structure, where the adjacent blocks have an overlap of m submatrices. Similarly, the encoder V of a classical convolutional code can be built from repeated applications of a linear invertible seed transformation U, which is an $(n + m) \times (n + m)$-element encoding matrix, as shown in Figure 6.5. The inverse encoder V^{-1} can be easily obtained by moving backwards in time, i.e. by reading Figure 6.5 from right to left. Let us further elaborate by stating that at time instant j, the seed transformation matrix U takes as its input the memory bits $m_{j-1} \in \mathbb{F}_2^m$, the logical bits $l_j \in \mathbb{F}_2^k$ and the syndrome bits $s_j \in \mathbb{F}_2^{n-k}$ to generate the output bits $e_j \in \mathbb{F}_2^n$ and the memory state m_j. More explicitly, we have:

$$(m_j : e_j) = (m_{j-1} : l_j : s_j)\,U\ , \tag{6.35}$$

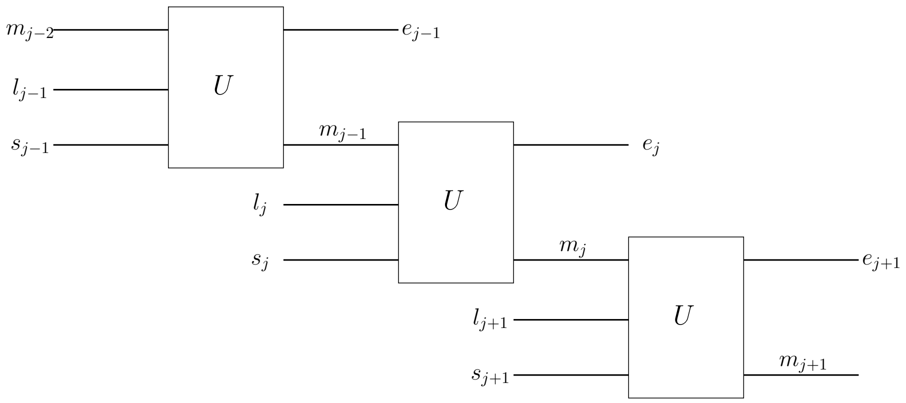


Figure 6.5: Circuit representation of the encoder V of a convolutional code [55].

and the overall encoder is formulated as [55]:

$$\begin{aligned} V &= U_{[1,\ldots,n+m]} U_{[n+1,\ldots,2n+m]} \cdots U_{[(N-1)n+1,\ldots,Nn+m]} \ , \\ &= \prod_{j=1}^{N} U_{[(j-1)n+1,\ldots,jn+m]}, \end{aligned} \tag{6.36}$$

where N denotes the length of the convolutional code and $U_{[(j-1)n+1,\ldots,jn+m]}$ acts on $(n+m)$ bits, i.e. $(m_{j-2} : l_{j-1} : s_{j-1})$. For an $C[n, k, m]$ quantum convolutional code, the seed transformation U is a $2(n+m) \times 2(n+m)$-element symplectic matrix and Eq. (6.35) may be re-written as:

$$(M_j : P_j) = (M_{j-1} : L_j : S_j)\, U \ , \tag{6.37}$$

where M represents the memory state with an m-qubit Pauli operator.

The aforementioned methodology conceived for constructing the circuit-based model of unassisted quantum codes may be readily extended to the class of EA codes [88]. The unitary encoding operation $\mathcal{V}$ of an $C[n, k, c]$ EA-QSC, which acts on the n transmitter qubits, i.e k information qubits, a auxiliary qubits and c pre-shared entangled qubits, may be mathematically modeled as follows:

$$\mathcal{C} = \{|\overline{\psi}\rangle = \mathcal{V}(|\psi\rangle^{T_X} \otimes |0_a\rangle^{T_X} \otimes |\phi_c^+\rangle^{T_X R_X})\} \ , \tag{6.38}$$

where the superscripts T_X and R_X denote the transmitter's and receiver's qubits, respectively. Furthermore, $|0_a\rangle^{T_X}$ are a auxiliary qubits initialized to the state $|0\rangle$, where $a = (n-k-c)$, and $|\phi_c^+\rangle^{T_X R_X}$ are the c entangled qubits. Analogously to Eq. (6.28), the inverse encoder of an EA quantum code $\mathcal{V}^\dagger$ gives:

$$\begin{aligned} \mathcal{V}^\dagger \mathcal{P} |\overline{\psi}\rangle &= \mathcal{V}^\dagger \mathcal{P} \mathcal{V}(|\psi\rangle^{T_X} \otimes |0_a\rangle^{T_X} \otimes |\phi_c^+\rangle^{T_X R_X}) \\ &= (\mathcal{L}^{T_X} |\psi\rangle^{T_X}) \otimes (\mathcal{S}^{T_X} |0_a\rangle^{T_X} \otimes (\mathcal{E}^{T_X} |\phi_c^+\rangle^{T_X R_X}), \end{aligned} \tag{6.39}$$

where $\mathcal{L}^{T_X} \in \mathcal{G}_k$ denotes the error imposed on the information word, while $\mathcal{S}^{T_X} \in \mathcal{G}_a$ represents the error inflicted on the transmitter's a auxiliary qubits and $\mathcal{E}^{T_X} \in \mathcal{G}_c$ is the error corrupting the transmitter's half of c ebits. The equivalent binary representation of Eq. (6.39) is given by:

$$PV^{-1} = (L : S : E)\,, \tag{6.40}$$

where we have $P = [\mathcal{P}^{T_X}]$, $L = [\mathcal{L}^{T_X}]$, $S = [\mathcal{S}^{T_X}]$ and $E = [\mathcal{E}^{T_X}]$. Similarly, Eq. (6.37) can be re-modeled as follows:

$$(M_j : P_j) = (M_{j-1} : L_j : S_j : E_j)\,U\ . \tag{6.41}$$

6.4 Concatenated Quantum Codes

6.4.1 System Model

Figure 6.6 shows the general schematic of a quantum communication system relying on a pair of concatenated quantum stabilizer codes. In our work, both the inner as well as the outer codes are assumed to be convolutional codes. Furthermore, analogously to the classical concatenated codes, the inner code must be recursive, while both the inner as well the outer code must be non-catastrophic. Using a recursive inner code is essential for the sake of ensuring that the resultant families of codes have an unbounded minimum distance. On the other hand, the non-catastrophic nature of both the inner and the outer codes guarantees that a decoding convergence to an infinitesimally low error rate is achieved. It was found in [54, 87] that QCCs cannot be simultaneously recursive and non-catastrophic. In order to overcome this problem, Wilde *et al.* [95, 88] proposed to employ EA inner codes, which are recursive as well as non-catastrophic. Therefore, the inner code should be an entanglement-assisted recursive and non-catastrophic code, while the outer code can be either an unassisted or an entanglement-assisted non-catastrophic code.

At the transmitter, the intended quantum information $|\psi_1\rangle$ is encoded by an $C[n_1, k_1]$ outer encoder $\mathcal{V}_1$ using $(n_1 - k_1)$ auxiliary qubits, which are initialized to the state $|0\rangle$, as depicted in Eq. (6.10). The encoded qubits $|\overline{\psi}_1\rangle$ are passed through a quantum interleaver (π). The resultant permuted qubits $|\psi_2\rangle$ are fed to an $C[n_2, k_2]$ inner encoder $\mathcal{V}_2$, which encodes them into the codewords $|\overline{\psi}_2\rangle$ using $(n_2 - k_2)$ auxiliary qubits initialized to the state $|0\rangle$[5]. The n-qubit codewords $|\overline{\psi}_2\rangle$, where we have $n = n_1 n_2$, are then serially transmitted over a quantum depolarizing channel, which imposes an n-tuple error $\mathcal{P}_2 \in \mathcal{G}_n$ on the transmitted codewords.

At the receiver, the received codeword $|\hat{\psi}_2\rangle = \mathcal{P}_2|\overline{\psi}_2\rangle$ is passed through the inverse encoder $\mathcal{V}_2^{\dagger}$, which yields the corrupted information word of the inner encoder $\mathcal{L}_2|\psi_2\rangle$ and the associated $(n_2 - k_2)$-qubit syndrome $\mathcal{S}_2|0_{(n_2-k_2)}\rangle$ as depicted previously in Eq. (6.28), where $\mathcal{L}_2$ denotes the error imposed on the logical qubits of the inner encoder, while $\mathcal{S}_2$ represents the error inflicted on the remaining $(n_2 - k_2)$ qubits. The corrupted logical qubits of the inner encoder are de-interleaved, resulting

[5] Please note that this is a general schematic. The inner code can be either an unassisted or an EA code. However, it is advisable to use an EA inner code for the sake of ensuring an unbounded minimum distance of the resultant concatenated code.

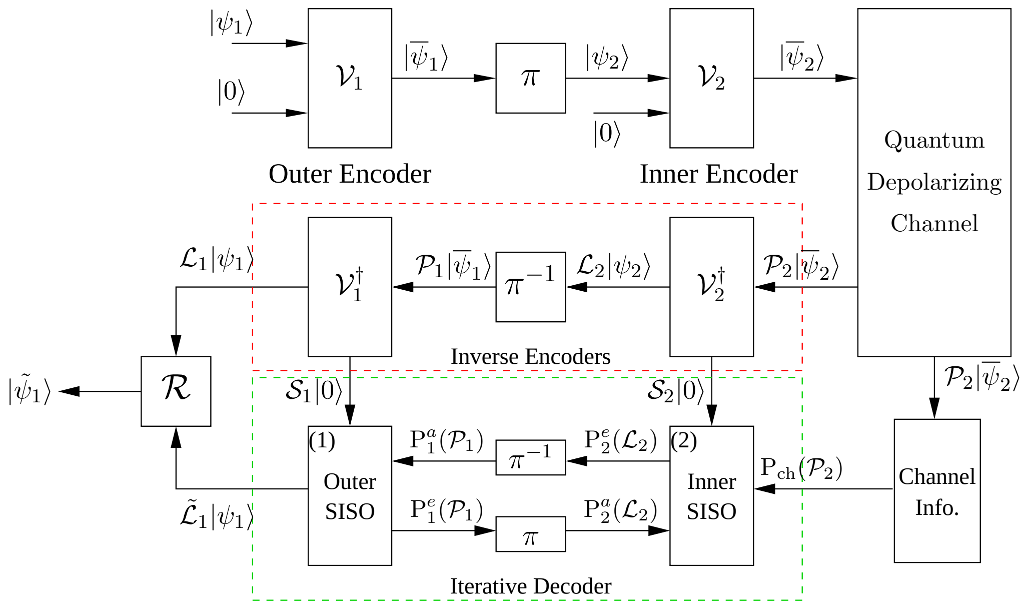

$|\psi_1\rangle$
$|0\rangle$
$\mathcal{V}_1$
$|\overline{\psi}_1\rangle$
π
$|\psi_2\rangle$
$|0\rangle$
$\mathcal{V}_2$
$|\overline{\psi}_2\rangle$
Outer Encoder
Inner Encoder
Quantum Depolarizing Channel
$\mathcal{L}_1|\psi_1\rangle$
$\mathcal{V}_1^\dagger$
$\mathcal{P}_1|\overline{\psi}_1\rangle$
π^{-1}
$\mathcal{L}_2|\psi_2\rangle$
$\mathcal{V}_2^\dagger$
$\mathcal{P}_2|\overline{\psi}_2\rangle$
Inverse Encoders
$|\tilde{\psi}_1\rangle$
$\mathcal{R}$
$\mathcal{S}_1|0\rangle$
$\mathcal{S}_2|0\rangle$
$\mathcal{P}_2|\overline{\psi}_2\rangle$
(1)
Outer SISO
$\mathrm{P}_1^a(\mathcal{P}_1)$
π^{-1}
$\mathrm{P}_2^e(\mathcal{L}_2)$
(2)
Inner SISO
$\mathrm{P}_{\mathrm{ch}}(\mathcal{P}_2)$
Channel Info.
$\tilde{\mathcal{L}}_1|\psi_1\rangle$
$\mathrm{P}_1^e(\mathcal{P}_1)$
π
$\mathrm{P}_2^a(\mathcal{L}_2)$
Iterative Decoder

Figure 6.6: System Model: Quantum communication system relying on concatenated quantum stabilizer codes. $\mathrm{P}_i^a(.)$, $\mathrm{P}_i^e(.)$ *and* $\mathrm{P}_i^o(.)$ *denote the a-priori, extrinsic and a-posteriori probabilities related to the ith decoder.*

in $\mathcal{P}_1|\overline{\psi}_1\rangle$, which is then passed through the inverse outer encoder $\mathcal{V}_1^\dagger$. This gives the corrupted information word of the outer encoder $\mathcal{L}_1|\psi_1\rangle$ and the associated (n_1-k_1)-qubit syndrome $\mathcal{S}_1|0_{(n_1-k_1)}\rangle$.

The next step is to estimate the error $\mathcal{L}_1$ for the sake of ensuring that the original logical qubit $|\psi_1\rangle$ can be restored by applying the recovery operation $\mathcal{R}$. For estimating $\mathcal{L}_1$, both the syndromes $\mathcal{S}_2|0_{(n_2-k_2)}\rangle$ and $\mathcal{S}_1|0_{(n_1-k_1)}\rangle$ are fed to the syndrome-based[6] inner and outer Soft-In Soft-Out (SISO) decoders [68], respectively, which engage in iterative decoding [55, 88] in order to yield the estimated error $\tilde{\mathcal{L}}_1$. The corresponding block is marked as 'Iterative Decoder' in Figure 6.6. Here, $\mathrm{P}_i^a(.)$, $\mathrm{P}_i^e(.)$ and $\mathrm{P}_i^o(.)$ denote the *a-priori*, *extrinsic* and *a-posteriori* probabilities [68] related to the ith decoder. Based on this notation, the turbo decoding process can be summarized as follows:

- The inner SISO decoder of Figure 6.6 uses the channel information $\mathrm{P}_{\mathrm{ch}}(\mathcal{P}_2)$, the *a-priori* information gleaned from the outer decoder $\mathrm{P}_2^a(\mathcal{L}_2)$ (initialized to be equiprobable for the first iteration) and the syndrome $\mathcal{S}_2$ to compute the *extrinsic* information $\mathrm{P}_2^e(\mathcal{L}_2)$. For a coded sequence of length N, we have $\mathcal{P}_2 = [\mathcal{P}_{2,1}, \mathcal{P}_{2,2}, \ldots, \mathcal{P}_{2,t}, \ldots, \mathcal{P}_{2,N}]$, where $\mathcal{P}_{2,t} = [\mathcal{P}_{2,t}^1, \mathcal{P}_{2,t}^2, \ldots, \mathcal{P}_{2,t}^n]$. The channel information $\mathbf{P}_{\mathrm{ch}}\left(P_{2,t}\right)$ is computed assuming that each qubit is independently transmitted over a quantum depolarizing channel having a depolarizing probability of p, whose channel transition probabilities are given by [55]:

$$\mathbf{P}_{\mathrm{ch}}\left(P_{2,t}^i\right) = \begin{cases} 1-p, & \text{if } \mathcal{P}_{2,t}^i = \mathbf{I} \\ p/3, & \text{if } \mathcal{P}_{2,t}^i \in \{\mathbf{X}, \mathbf{Z}, \mathbf{Y}\}. \end{cases} \tag{6.42}$$

[6] See Section 5.4.2.2 for details of syndrome-based SISO or Maximum *A-Posteriori* (MAP) decoder.

- $\mathrm{P}_2^e(\mathcal{L}_2)$ is passed through the quantum de-interleaver (π^{-1}) of Figure 6.6 to generate the *a-priori* information for the outer decoder $\mathrm{P}_1^a(\mathcal{P}_1)$.

- Based on both the *a-priori* information $\mathrm{P}_1^a(\mathcal{P}_1)$ and on the syndrome $\mathcal{S}_1$, the outer SISO decoder of Figure 6.6 computes both the *a-posteriori* information $\mathrm{P}_1^o(\mathcal{L}_1)$ and the *extrinsic* information $\mathrm{P}_1^e(\mathcal{P}_1)$.

- $\mathrm{P}_1^e(\mathcal{P}_1)$ is then interleaved to obtain $\mathrm{P}_2^a(\mathcal{L}_2)$, which is fed back to the inner SISO decoder of Figure 6.6. This iterative procedure continues, until either convergence is achieved or the maximum affordable number of iterations is reached.

- Finally, a qubit-based Maximum *A-Posteriori* (MAP) decision is made for determining the most likely error coset $\mathcal{L}_1$. It must be mentioned here that both the inner and outer SISO decoders employ the degenerate decoding approach of [55], which aims for finding the 'most likely error coset' rather than the 'most likely error' acting on the logical qubits $\mathcal{L}_i$, as we will discuss in the next section.

6.4.2 Degenerate Iterative Decoding

As discussed in Section 4.3.3, quantum codes exhibit the intrinsic property of degeneracy, which is also obvious from Eq. (6.28). More explicitly, we have:

$$\mathcal{S}|0_{n-k}\rangle = \mathcal{S}_1|0\rangle \otimes \cdots \otimes \mathcal{S}_{n-k}|0\rangle. \tag{6.43}$$

Since, we have $\mathcal{S}_j \in \{\mathbf{I}, \mathbf{X}, \mathbf{Y}, \mathbf{Z}\}$, we can re-write Eq. (6.43) as follows [55]:

$$\mathcal{S}|0_{n-k}\rangle \equiv \epsilon|s_1\rangle \otimes \cdots \otimes |s_{n-k}\rangle, \tag{6.44}$$

where $\epsilon \in \{\pm 1, \pm i\}$, and:

$$\begin{aligned} s_j &= 0 \quad \text{if } \mathcal{S}_j = \mathbf{I} \text{ or } \mathcal{S}_j = \mathbf{Z}, \\ s_j &= 1 \quad \text{otherwise.} \end{aligned} \tag{6.45}$$

For example, if $\mathcal{S}_1 = \mathbf{Y}$ and $\mathcal{S}_j = \mathbf{I}$ for $j \neq 1$, since $\mathbf{Y} = i\mathbf{XZ}$, we get $\mathcal{S}|0_{n-k}\rangle = i|1\rangle \otimes |0\rangle \otimes \cdots \otimes |0\rangle$.

Observing the $(n-k)$ syndrome qubits of Eq. (6.44) collapses them to the classical syndrome $s = \{s_1, \ldots, s_{n-k}\}$, which is equivalent to the symplectic product of P and H, i.e. $s = (P \star H_j)_{1 \leq j \leq n-k}$. More precisely, the syndrome sequence $|0_{n-k}\rangle$ is invariant to the $\mathbf{Z}$-component of $\mathcal{S}$ since $\mathbf{Z}|0\rangle = |0\rangle$. Let S be the effective $2(n-k)$-bit error on the syndrome, which may be decomposed as $S = S^x + S^z$, where S^x and S^z are the $\mathbf{X}$ and $\mathbf{Z}$ components of S, respectively. Then s only reveals S^x. Hence, two distinct error sequences $P = (L : S^x + S^z)V$ and $P' = (L : S^x + S'^z)V$, which only differ in the $\mathbf{Z}$-component of $\mathcal{S}$, yield the same syndrome s. Furthermore, it must be noted that both P and P' have the same logical error L. Therefore, P and P' differ only by the stabilizer group and are known

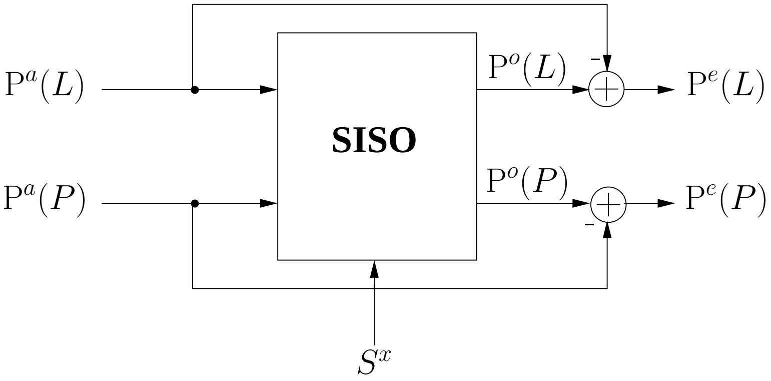


Figure 6.7: General schematic of a SISO decoder. $\mathrm{P}^a(.)$, $\mathrm{P}^e(.)$ *and* $\mathrm{P}^o(.)$ *denote the a-priori, extrinsic and a-posteriori probabilities.*

as degenerate errors, which do not have to be distinguished, since they can be corrected by the same recovery operation L^{-1}.

The classic syndrome-based MAP decoder aims for finding the most likely error for a given syndrome, which may be modeled as:

$$L(S) = \mathrm{argmax}_L \mathrm{P}(L|S), \tag{6.46}$$

where $\mathrm{P}(L|S)$ denotes the probability of experiencing the logical error L imposed on the transmitted qubits, given that the syndrome of the received qubits is S. By contrast, quantum codes employ degenerate decoding, which aims for finding the most likely error coset $C(L, S^x)$ associated with the observed syndrome S^x. The coset $C(L, S^x)$ is defined as [55]:

$$C(L, S^x) = \{P = (L : S^x + S^z)V\} \quad \forall S^z \in \{\mathbf{I}, \mathbf{Z}\}^{n-k}. \tag{6.47}$$

Therefore, a degenerate MAP decoder yields:

$$L(S^x) = \mathrm{argmax}_L \mathrm{P}(L|S^x), \tag{6.48}$$

where we have:

$$\mathrm{P}(L|S^x) \equiv \sum_{S^z \in \{\mathbf{I}, \mathbf{Z}\}^{n-k}} \mathrm{P}(L|(S^x + S^z)). \tag{6.49}$$

The iterative decoder of Figure 6.6 consists of two serially concatenated SISO decoders, which employ the aforementioned degenerate decoding approach. Figure 6.7 shows the general schematic of a SISO decoder, where the Pauli operators $\mathcal{P}$, $\mathcal{L}$ and $\mathcal{S}$ are replaced by the effective operators P, L and S^x, respectively. The SISO decoder of Figure 6.7 yields the *a-posteriori* information pertaining to the logical error L and channel error P based on the classic forward-backward recursive coefficients α and β, which are computed over the quantum circuit of Figure 6.8 (analogous to a classic trellis) as follows [55]:

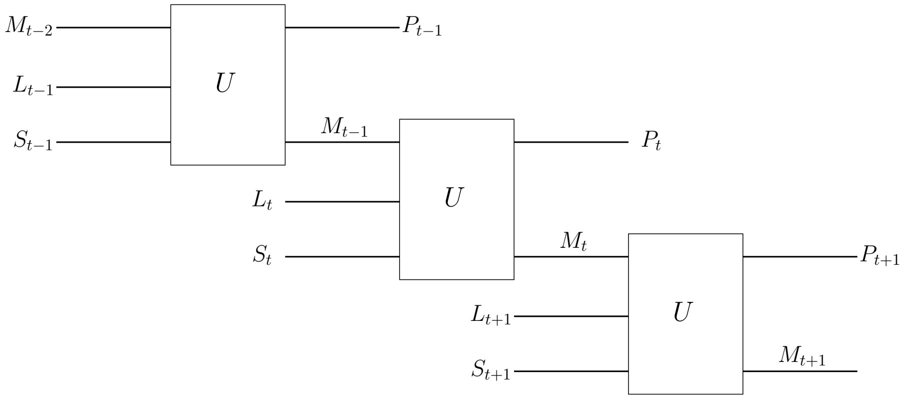


Figure 6.8: Circuit representation of the encoder V of a quantum convolutional code [55]. The operation of the tth seed transformation can be characterized as $(M_t : P_t) = (M_{t-1} : L_t : S_t)U$, where $M_t \in G_m$, $P_t \in G_n$, $L_t \in G_k$ and $S_t \in G_{n-k}$, while U is a $2(n+m) \times 2(n+m)$-element binary symplectic matrix.

- For a coded sequence of duration N, let us denote the channel error sequence by $P = [P_1, P_2, \ldots, P_t, \ldots, P_N]$ and the logical error sequence by $L = [L_1, L_2, \ldots, L_t, \ldots, L_N]$, where we have $P_t \in G_n$ and $L_t \in G_k$. More explicitly, $P_t = [P_t^1, P_t^2, \ldots, P_t^n]$ and $L_t = [L_t^1, L_t^2, \ldots, L_t^k]$.

- Let us decompose the $2(n+m) \times 2(n+m)$-element binary symplectic matrix U of Figure 6.8 as $U = (U_M : U_P)$, where U_M is the binary matrix formed by the first $2m$ columns of U, while U_P is the binary matrix formed by the last $2n$ columns of U. Therefore, Eq. (6.37) can be decomposed as:

$$M_t = (M_{t-1} : L_t : S_t)\, U_M, \tag{6.50}$$

$$P_t = (M_{t-1} : L_t : S_t)\, U_P. \tag{6.51}$$

- The forward recursive coefficient $\alpha_t\left(M_t\right)$ is the probability that the memory state of the tth seed transformation U in the circuit of Figure 6.8 is M_t at time instant t, given the syndrome sequence up to this point, i.e. $S^x_{\leq t} \triangleq \left(S^x_j\right)_{0 \leq j \leq t}$, which can be mathematically modeled as:

$$\begin{aligned} \alpha_t\left(M_t\right) &\triangleq \mathrm{P}\left(M_t | S^x_{\leq t}\right), \\ &\propto \sum_{\substack{(\mu,\lambda,\sigma)\Rightarrow \\ M_t=(\mu:\lambda:\sigma)U_M}} \mathrm{P}^a\left(L_t = \lambda\right) \mathrm{P}^a\left(P_t = (\mu : \lambda : \sigma)U_P\right) \alpha_{t-1}\left(M_{t-1} = \mu\right), \end{aligned} \tag{6.52}$$

where the summation implies adding all the probabilities associated with those specific values of the memory state $M_{t-1} = \mu \in G_m$, of the logical error $L_t = \lambda \in G_k$ and of the error inflicted on the auxiliary qubits $S_t = \sigma \in G_{n-k}$, which yield a particular value of M_t according to Eq. (6.50). Furthermore, σ can be decomposed into the $\mathbf{X}$ and $\mathbf{Z}$ components such that $\sigma = \sigma_x + \sigma_z$, having $\sigma_x = S^x_t$, while σ_z can assume all possible values, as seen in Eq. (6.49).

- The backward recursive coefficient $\beta_t\left(M_t\right)$ is the probability that the memory state of the tth seed transformation U in the circuit of Figure 6.8 is M_t given the future syndrome sequence, i.e. $S^x_{>t} \triangleq \left(S^x_j\right)_{t<j\leq N}$, which can be encapsulated as:

$$\begin{aligned}\beta_t\left(M_t\right) &\triangleq \mathrm{P}\left(M_t|S^x_{>t}\right), \\ &\propto \sum_{\lambda,\sigma} \mathrm{P}^a\left(L_{t+1}=\lambda\right)\mathrm{P}^a\left(P_{t+1}=(M_t:\lambda:\sigma)U_P\right)\beta_{t+1}\left(M_{t+1}=\left(M_t:\lambda:\sigma\right)U_M\right), \quad (6.53)\end{aligned}$$

where the summation implies adding the probabilities for all possible values of logical errors $L_{t+1}=\lambda\in G_k$ plus the errors inflicted on the auxiliary qubits $S_{t+1}=\sigma\in G_{n-k}$, given furthermore that $\sigma=\sigma_x+\sigma_z$, having $\sigma_x=S^x_{t+1}$.

- Finally, we have the *a-posteriori* probabilities $\mathrm{P}^o(L_t)$ and $\mathrm{P}^o(P_t)$, which are given by:

$$\begin{aligned}\mathrm{P}^o(L_t) &\triangleq \mathrm{P}(L_t|S^x), \\ &\propto \sum_{\mu,\sigma}\mathrm{P}^a(L_t)\mathrm{P}^a(P_t=(\mu:L_t:\sigma)U_P)\alpha_{t-1}\left(M_{t-1}=\mu\right)\beta_t\left(M_t=\left(\mu:L_t:\sigma\right)U_M\right), \quad (6.54)\\ \mathrm{P}^o(P_t) &\triangleq \mathrm{P}(P_t|S^x), \\ &\propto \sum_{\substack{\mu,\lambda,\sigma\Rightarrow\\ P_t=(\mu:\lambda:\sigma)U_P}}\mathrm{P}^a(P_t)\mathrm{P}^a(L_t=\lambda)\alpha_{t-1}\left(M_{t-1}=\mu\right)\beta_t\left(M_t=\left(\mu:\lambda:\sigma\right)U_M\right), \quad (6.55)\end{aligned}$$

where $S^x\triangleq\left(S^x_t\right)_{0\leq t\leq N}$, while the memory state $M_{t-1}=\mu\in G_m$, logical error $L_t=\lambda\in G_k$ and the error inflicted on the auxiliary qubits $S_t=\sigma\in G_{n-k}$, given further that $\sigma=\sigma_x+\sigma_z$, having $\sigma_x=S^x_t$.

- The marginalized probabilities $\mathrm{P}^o(L^j_t)$, for $j\in\{0,k-1\}$, and $\mathrm{P}^o(P^j_t)$, for $j\in\{0,n-1\}$, are then computed from $\mathrm{P}^o(L^j_t)$ and $\mathrm{P}^o(P^j_t)$, respectively. The *a-priori* information is then removed in order to yield the *extrinsic* probabilities [88]. Therefore, in the logarithmic domain, we have:

$$\ln[\mathrm{P}^e(L^j_t)] = \ln[\mathrm{P}^o(L^j_t)] - \ln[\mathrm{P}^a(L^j_t)], \quad (6.56)$$

$$\ln[\mathrm{P}^e(P^j_t)] = \ln[\mathrm{P}^o(P^j_t)] - \ln[\mathrm{P}^a(P^j_t)]. \quad (6.57)$$

It has to be mentioned here that the property of degeneracy is only an attribute of auxiliary qubits and the ebits of an EA code do not contribute to it. This is because both **X** as well as **Z** errors acting on the transmitter's half of ebits give distinct results when measured in the Bell basis, i.e. $\mathcal{E}^{T_X}|\phi^+_c\rangle^{T_XR_X}$ gives four distinct Bell states for $\mathcal{E}^{T_X}_j\in\{\mathbf{I},\mathbf{X},\mathbf{Z},\mathbf{Y}\}$. Consequently, degeneracy is a function of the number of auxiliary qubits a and reduces to zero for $a=0$.

6.5 EXIT-Chart Aided Quantum Code Design

The EXIT chart analysis not only allows us to dispense with the time-consuming Monte-Carlo simulations, but also facilitates the design of capacity approaching codes without resorting to the tedious analysis of their distance spectra. Therefore, they have been extensively employed for designing near-capacity classical codes [157, 158, 159], as we previously demonstrated in Chapter 3. Let us further recall from Section 3.5.1 that the EXIT chart of a serially concatenated scheme visualizes the exchange of four Mutual Information (MI) terms, i.e. the average *a-priori* MI of the outer decoder I_A^1, the average *a-priori* MI of the inner decoder I_A^2, the average *extrinsic* MI of the outer decoder I_E^1, and the average *extrinsic* MI of the inner decoder I_E^2. More specifically, I_A^1 and I_E^1 constitute the EXIT curve of the outer decoder, while I_A^2 and I_E^2 yield the EXIT curve of the inner decoder. The MI transfer characteristics of both the decoders are plotted in the same graph, with the x and y axes of the outer decoder swapped. The resultant EXIT chart quantifies the improvement in the mutual information as the iterations proceed, which can be viewed as a stair-case-shaped decoding trajectory. An open tunnel between the two EXIT curves ensures that the decoding trajectory reaches the $(1, y)$ point of perfect convergence.

In this section, we extended the application of EXIT charts to the quantum domain by appropriately adapting the conventional non-binary EXIT chart generation technique [160, 114] for the quantum syndrome decoding approach. Before proceeding with the application of EXIT charts for quantum codes, let us elaborate on the quantum-to-classical isomorphism of Section 4.3.2, which forms the basis of our EXIT chart aided approach. As discussed in Section 4.3.2.1, a Pauli error operator $\mathcal{P}$ experienced by an N-qubit frame transmitted over a depolarizing channel can be modeled by an effective error-vector P, which is a binary vector of length $2N$. The first N bits of P denote Z errors, while the remaining N bits represent $\mathbf{X}$ errors. More explicitly, an $\mathbf{X}$ error imposed on the 1st qubit will yield a 0 and a 1 at the 1st and $(N + 1)$th index of P, respectively. Similarly, a $\mathbf{Z}$ error imposed on the 1st qubit will give a 1 and a 0 at the 1st and $(N+1)$th index of P, respectively, while a $\mathbf{Y}$ error on the 1st qubit will result in a 1 at both the 1st as well as $(N+1)$th index of P. Since a depolarizing channel characterized by the probability p incurs $\mathbf{X}$, $\mathbf{Y}$ and $\mathbf{Z}$ errors with an equal probability of $p/3$, the effective error-vector P reduces to two Binary Symmetric Channels (BSCs) having a crossover probability of $2p/3$, where we have one channel for the $\mathbf{Z}$ errors and the other for the $\mathbf{X}$ errors. Hence, a quantum depolarizing channel has been considered analogous to a BSC [47, 161], whose capacity is given by:

$$\mathrm{C}_C^{\mathrm{BSC}}(p) = 1 - \mathrm{H}_2(2p/3), \tag{6.58}$$

where H_2 is the binary entropy function. Let us recall from Eq. (4.38) that the code rate R_Q of an $[n, k]$ QSC is related to the equivalent classical code rate R_C as follows [47, 81]:

$$R_C = \frac{1}{2}\left(1 + R_Q\right). \tag{6.59}$$

The corresponding quantum capacity can be computed by substituting Eq. (6.58) into Eq. (6.59) such that $R_C = \mathrm{C}_C^{\mathrm{BSC}}(p)$, which yields [47, 81]:

$$\mathrm{C}_Q^{\mathrm{BSC}}(p) = 1 - 2\mathrm{H}_2(2p/3). \tag{6.60}$$

However, the two BSCs constituting a quantum depolarizing channel are not entirely independent. There is an inherent correlation between the $\mathbf{X}$ and $\mathbf{Z}$ errors [47], which is characterized in Table 6.1.

	$\mathbf{Z} = 0$	$\mathbf{Z} = 1$
$\mathbf{X} = 0$	$1 - p$	$p/3$
$\mathbf{X} = 1$	$p/3$	$p/3$

Table 6.1: Correlation between **X** and **Z** errors on th ith qubit in terms of the corresponding probability of occurrence.

This correlation is taken into account by the iterative decoder of Section 6.4.2. Alternatively, a quantum depolarization channel can also be considered equivalent to a 4-ary symmetric channel based on the Pauli-to-quaternary isomorphism of Section 4.3.2.2. More explicitly, the ith and $(N+i)$th index of P constitute the 4-ary symbol according to the Pauli-to-quaternary isomorphism of Section 4.3.2.2. Hence, the corresponding classical capacity is equivalent to the maximum rate achievable over each half of the 4-ary symmetric channel, or more explicitly the normalized capacity of a 4-ary symmetric channel [47, 81]. Since the **X**, **Y** and **Z** errors occur with an equal probability of $p/3$ over a depolarizing channel, the normalized capacity of the equivalent 4-ary classical channel is given by:

$$\begin{aligned}\mathrm{C}_C^{\text{4-ary}}(p) &= \frac{1}{2}\left(2 + (1-p)\log_2(1-p) + \frac{p}{3}\log_2\left(\frac{p}{3}\right) + \frac{p}{3}\log_2\left(\frac{p}{3}\right) + \frac{p}{3}\log_2\left(\frac{p}{3}\right)\right) \\ &= \frac{1}{2}\left(2 + (1-p)\log_2(1-p) + p\log_2(p) - p\log_2(3)\right) \\ &= \frac{1}{2}\left(2 - \mathrm{H}_2(p) - p\log_2(3)\right). \end{aligned} \tag{6.61}$$

Substituting Eq. (6.61) into Eq. (6.59) yields the Hashing bound of Eq. (6.1), i.e. we have:

$$\mathrm{C}_Q(p) = 1 - \mathrm{H}_2(p) - p\log_2(3). \tag{6.62}$$

Recall that a quantum code is equivalent to a classical code through Eq. (4.30). Additionally, the decoding of a quantum code is essentially carried out with the aid of the equivalent classical code by exploiting the additional property of degeneracy. Particularly, quantum codes employ the syndrome decoding approach of Chapter 5, which yields information about the **error-sequence** rather than the information-sequence or coded qubits, hence avoiding the observation of the latter sequences, which would collapse them back to the classical domain.

Since a quantum code has an equivalent classical representation and the depolarizing channel is analogous to a BSC, we employ the EXIT chart technique to design concatenated quantum codes that can approach the Hashing bound. The major difference between the EXIT charts conceived for the classical and quantum domains is that while the former models the *a-priori* information concerning the input bits of the inner encoder (and similarly the output bits of the outer encoder), the latter models the *a-priori* information concerning the corresponding error-sequence, i.e. the error-sequence related to the input qubits of the inner encoder L_2 (and similarly the error-sequence related to the output qubits of the outer encoder P_1).

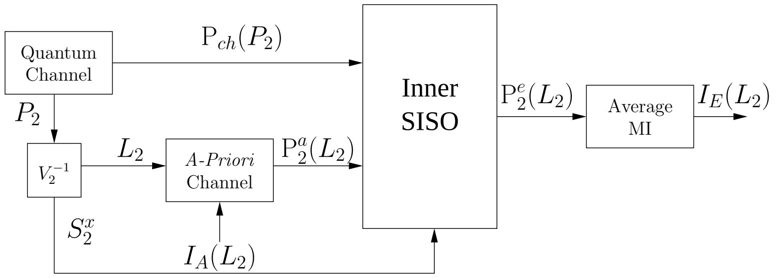


Figure 6.9: System model for generating the EXIT chart of the inner decoder.

Similar to the classical EXIT charts, it is assumed that the interleaver length is sufficiently high to ensure that [116, 68]:

- the *a-priori* values are fairly uncorrelated; and
- the *a-priori* information has a Gaussian distribution.

Figure 6.9 shows the system model used for generating the EXIT chart of the inner decoder. Here, a quantum depolarizing channel having a depolarizing probability of p generates the error sequence P_2, which is passed through the inverse inner encoder V_2^{-1}. This yields both the error imposed on the logical qubits L_2 and the syndrome S_2^x. The *a-priori* channel block then models the *a-priori* probability $\mathrm{P}_2^a(L_2)$ such that the average MI between the actual error L_2 and the *a-priori* probabilities $\mathrm{P}_2^a(L_2)$ is given by $I_A(L_2)$ [116, 68, 117]. More explicitly, we have $I_A(L_2) = I[L_2, \mathrm{P}_2^a(L_2)]$, where I denotes the average MI function. Moreover, the ith and $(N+i)$th bits of the effective error vector L_2 can be visualized as 4-ary symbols. Consequently, similar to classical non-binary EXIT charts [160, 114], the logarithmic *a-priori* probability is modeled using an independent Gaussian distribution with a mean of $\sigma_A^2/2$ and variance of σ_A^2, assuming that the **X** and **Z** errors constituting the 4-ary symbols are independent[7]. Based on the channel probability $\mathrm{P}_{\mathrm{ch}}(P_2)$, on the syndrome S_2^x and on the *a-priori* probability $\mathrm{P}_2^a(L_2)$, the inner SISO decoder generates the *extrinsic* probability $\mathrm{P}_2^e(L_2)$ by using the degenerate decoding approach of Section 6.4.2. Finally, the *extrinsic* average MI $I_E(L_2) = I[L_2, \mathrm{P}_2^e(L_2)]$ between L_2 and $\mathrm{P}_2^e(L_2)$ is computed. Since the equivalent classical capacity of a quantum channel is given by the capacity achievable over each half of the 4-ary symmetric channel, $I_E(L_2)$ is the normalized MI of the 4-ary symbols, which can be computed based on [114, 115] as:

$$I_E(L_2) = \frac{1}{2}\left(2 + \mathrm{E}\left[\sum_{\mathrm{m}=0}^{3} \mathrm{P}_2^e(L_2^{j(\mathrm{m})}) \log_2 \mathrm{P}_2^e(L_2^{j(\mathrm{m})})\right]\right), \tag{6.63}$$

where E is the expectation (or time average) operator and $L_2^{j(\mathrm{m})}$ is the m$^{\mathrm{th}}$ hypothetical error imposed on the logical qubits. More explicitly, since the error on each qubit is represented by an equivalent

[7] Under the idealized asymptotic conditions of having an infinite-length interleaver, $I_A(L_2)$ may be accurately modeled by the Gaussian distribution. As and when shorter interleavers are used, the Gaussian assumption becomes less accurate, hence in practice a histogram-based approximation may be relied upon.

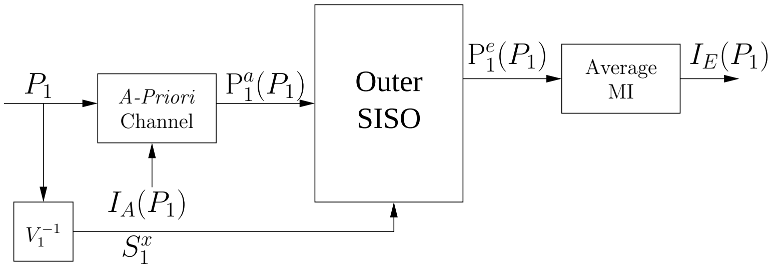


Figure 6.10: System model for generating the EXIT chart of the outer decoder [4].

pair of classical bits, $L_2^{j(\mathrm{m})}$ is a 4-ary classical symbol associated with m $\in \{0, 3\}$. The process is repeated for a range of $I_A(L_2) \in [0, 1]$ values for the sake of obtaining the *extrinsic* information transfer characteristics at the depolarizing probability p. The resultant inner EXIT function T_2 of the specific inner decoder may be defined as follows:

$$I_E(L_2) = T_2[I_A(L_2), p], \tag{6.64}$$

which is a function of the channel's depolarizing probability p.

The system model used for generating the EXIT chart of the outer decoder is depicted in Figure 6.10. As inferred from Figure 6.10, the EXIT curve of the outer decoder is independent of the channel's output information. The *a-priori* information is generated by the *a-priori* channel based on P_1 (error on the physical qubits of the second decoder) and $I_A(P_1)$, which is the average MI between P_1 and $\mathrm{P}_1^a(P_1)$. Furthermore, as for the inner decoder, P_1 is passed through the inverse outer encoder V_1^{-1} to compute S_1^x, which is fed to the outer SISO decoder to yield the *extrinsic* probability $\mathrm{P}_1^e(P_1)$. The average MI between P_1 and $\mathrm{P}_1^e(P_1)$ is then calculated similar to Eq. (6.63) as follows:

$$I_E(P_1) = \frac{1}{2}\left(2 + \mathrm{E}\left[\sum_{\mathrm{m}=0}^{3} \mathrm{P}_1^e(P_1^{j(\mathrm{m})}) \log_2 \mathrm{P}_1^e(P_1^{j(\mathrm{m})})\right]\right). \tag{6.65}$$

The resultant EXIT chart is characterized by the following MI transfer function:

$$I_E(P_1) = T_1[I_A(P_1)], \tag{6.66}$$

where T_1 is the outer EXIT function, which is dependent on the specific outer decoder, but it is independent of the depolarizing probability p.

Finally, the MI transfer characteristics of both decoders characterized by Eq. (6.64) and Eq. (6.66) are plotted in the same graph, with the x and y axes of the outer decoder swapped. For the sake of approaching the achievable capacity of Figure 6.1, *our EXIT-chart aided design aims for creating a narrow, but marginally open tunnel between the EXIT curves of the inner and outer decoders at the highest possible depolarizing probability (analogous to the lowest possible SNR for a classical channel).* For a given noise limit p^* and the desired code parameters, this may be achieved in two steps. We first find that specific inner code, which yields the largest area under its EXIT-curve at the noise limit p^*.

Once the optimal inner code is selected, we find the optimal outer code, whose EXIT-curve gives the best match with the chosen inner code. The narrower the tunnel-area between the inner and outer decoder's EXIT curve, the lower is the deviation from the achievable capacity or the Hashing bound, which may be quantified using Eq. (6.2).

6.6 Results and Discussions I

6.6.1 Accuracy of EXIT Chart Predictions

In order to verify the accuracy of our EXIT-chart based approach, we have analyzed the convergence behaviour of a rate-1/9 QTC, consisting of two identical rate-1/3 QCCs, whose parameters are listed in Table 6.2. More specifically, for both the inner and outer decoders, we have used the configuration termed as "PTO1R" in [95, 88], which is a non-catastrophic but quasi-recursive code.

Our first aim was to predict the convergence threshold using EXIT charts, which would otherwise require time-consuming Word Error Rate (WER) or QBER simulations. Convergence threshold can be determined by finding the maximum depolarizing probability p, which yields a marginally open EXIT tunnel between the EXIT curves of the inner and outer decoder; hence, facilitating an infinitesimally low error rate. Figure 6.11 shows the EXIT curves for the inner and outer decoders, where the area under the EXIT curve of the inner decoder decreases upon increasing p. Eventually, the inner and outer curves crossover, when p is increased to $p = 0.13$. More explicitly, increasing p beyond 0.125, closes the EXIT tunnel. Hence, the convergence threshold is around $p = 0.125$. Figure 6.12 further shows two decoding trajectories superimposed on the EXIT chart of Figure 6.11 at $p = 0.125$. We have used a $30,000$-qubit long interleaver. As seen from Figure 6.12, the trajectory successfully reaches the $(x, y) = (1, y)$ point of the EXIT chart. This in turn guarantees an infinitesimally low WER/QBER at $p = 0.125$ for an interleaver of infinite length.

We have further verified the validity of our EXIT chart predictions using WER simulations. Figure 6.13 shows the WER performance curve for the simulation parameters of Table 6.2. The achievable performance of Figure 6.13 improves upon increasing the number of iterations from one through to eight. More specifically, the turbo-cliff region starts to emerge around $p = 0.125$ (marked with a dashed-line in Figure 6.13), whereby the WER drops as the iterations proceed. Therefore, our EXIT chart predictions of Figure 6.11 follow the Monte-Carlo simulation results of Figure 6.13 to a reasonable degree.

6.6.2 Entanglement-Assisted and Unassisted Inner Codes

All non-catastrophic QCCs are non-recursive [55]. Therefore, the resultant families of QTCs have a bounded minimum distance and they do not have a true iterative threshold. To circumvent this limitation of QTCs, Wilde *et al.* [95, 88] proposed to employ EA inner codes, which are recursive as well

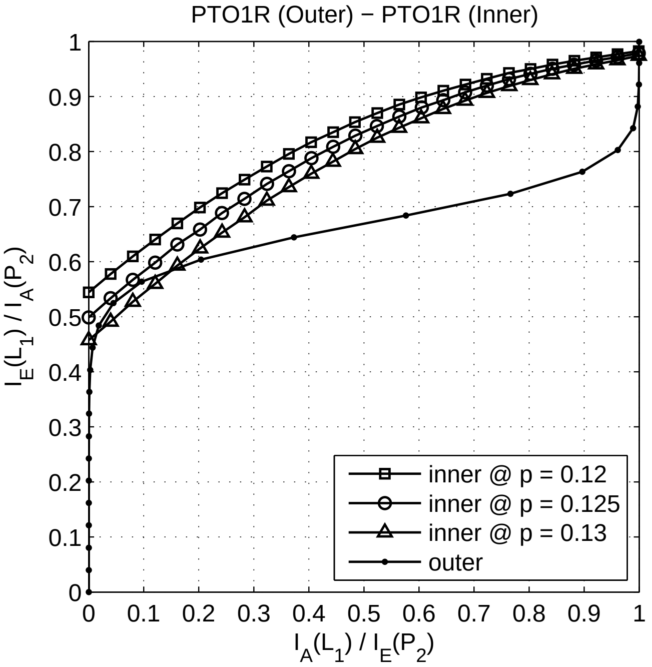


Figure 6.11: The EXIT curves of a QTC parametrized by the increasing depolarizing probability p. The parameters of the inner and outer QCC are listed in Table 6.2.

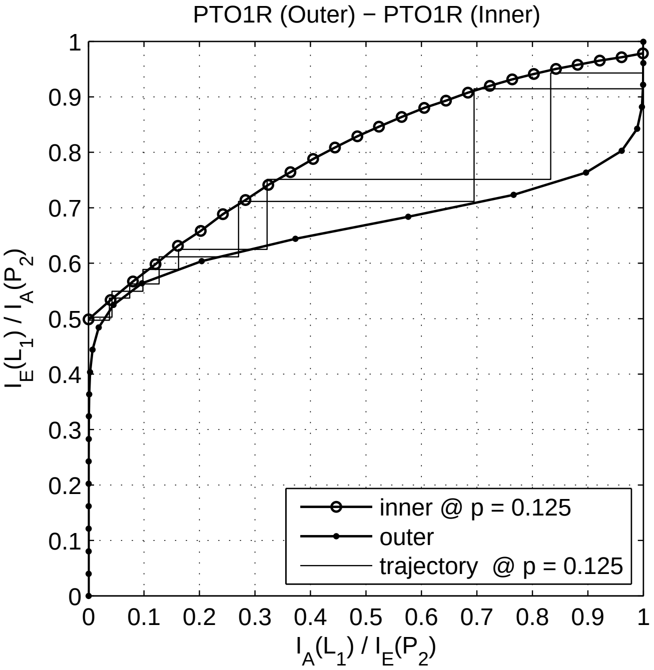


Figure 6.12: The EXIT chart of a QTC with decoding trajectories at $p = 0.125$. We have used the QTC of Table 6.2 with an increased interleaver length of $30,000$ qubits.

Coding rate	$R = 1/9$
Entanglement consumption rate	$\mathtt{E} = 0$
Interleaver length	$N = 3,000$ qubits
Iterations	$\mathtt{I} = 8$
Inner QCC	
Coding rate	$R_i = 1/3$
Entanglement consumption rate	$\mathtt{E}_i = 0$
Memory	3
Seed transformation	$U_i = \{1355, 2847, 558, 2107, 3330, 739, 2009, 286, 473, 1669, 1979, 189\}_{10}$
Outer QCC	
Coding rate	$R_o = 1/3$
Entanglement consumption rate	$\mathtt{E}_o = 0$
Memory	3
Seed transformation	$U_o = \{1355, 2847, 558, 2107, 3330, 739, 2009, 286, 473, 1669, 1979, 189\}_{10}$

Table 6.2: Simulation parameters of the concatenated scheme of Figure 6.6. The inner/outer QCC is the "PTO1R" configuration of [95, 88], which is an unassisted rate-1/3 QCC (quasi-recursive and non-catastrophic).

as non-catastrophic. The resulting families of EA-QTCs have an unbounded minimum distance [95, 88], i.e. their minimum distance increases almost linearly with the interleaver length. Here, we verify this by analyzing the inner decoder's EXIT curves for both the unassisted (non-recursive) and the EA (recursive) inner convolutional codes.

For classical recursive inner codes, the inner decoder's EXIT curve reaches the $(x, y) = (1, 1)$ point[8], which guarantees perfect decoding convergence to a vanishingly low WER/QBER as well as having an unbounded minimum distance for the family of QTCs [55] based on these inner codes.

[8] Note that we only need $(x, y) = (1, y)$ for achieving decoding convergence to an infinitesimally low error rate. However, this requires an outer code having a sufficiently large minimum distance for the sake of ensuring that the outer code's EXIT curve does not intersect with that of the inner code before reaching the $(1, y)$ point. Unfortunately, an outer code having a large minimum distance would result in an EXIT curve having a large open-tunnel area. Thus, it will operate far from the capacity.

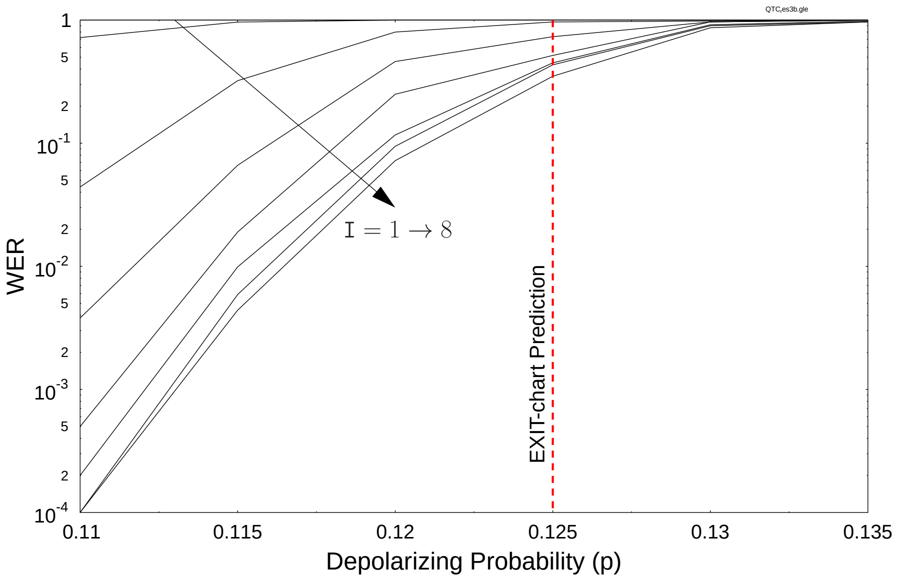


Figure 6.13: Achievable WER performance of the QTC of Figure 6.6 as the number of iterations ($\mathtt{I}$) is increased from 1 through to the 8. Simulation parameters are summarized in Table 6.2. The dashed-line at $p = 0.125$ marks the convergence threshold predicted using the EXIT chart of Figure 6.12.

Consequently, the resulting families of QTCs have unbounded minimum distance and hence an arbitrarily low WER/QBER can be achieved for an infinitely long interleaver. This also holds true for the recursive QCCs, as shown in Figure 6.14. In this figure, we compare the inner decoder's EXIT curves of both the unassisted QCC (non-recursive) and the EA-QCC (recursive) of [95], which are labeled "PTO1R" and "PTO1REA", respectively. More explicitly, PTO1REA is the maximally-entangled[9] version of PTO1R, which has the same configuration as that of the PTO1R given in Table 6.2, but with an increased entanglement consumption rate of 2/3. We may observed in Figure 6.14 that, for the PTO1R configuration, decreasing the depolarizing probability from $p = 0.14$ to $p = 0.12$ shifts the inner decoder's EXIT curve upwards and towards the $(1, 1)$ point. Hence, the EXIT curve will manage to reach the $(1, 1)$ point only at very low values of depolarizing probability. By contrast, the EXIT curve of PTO1REA always terminates at $(1, 1)$, regardless of the value of p. Therefore, provided an open EXIT tunnel exists and the interleaver length is sufficiently long, the decoding trajectories of an EA-QTC will always reach the $(1, 1)$ point; thus, guaranteeing an arbitrarily low WER/QBER for the family of QTCs based on these inner codes. Particularly, the performance improves upon increasing the interleaver length; thus, implying that the minimum distance increases upon increasing the interleaver length. Therefore, the resultant QTCs have an unbounded minimum distance.

[9] For a maximally entangled code, we have $c = (n - k)$, resulting in an entanglement consumption rate of $(1 - R_Q)$, where R_Q is its coding rate.

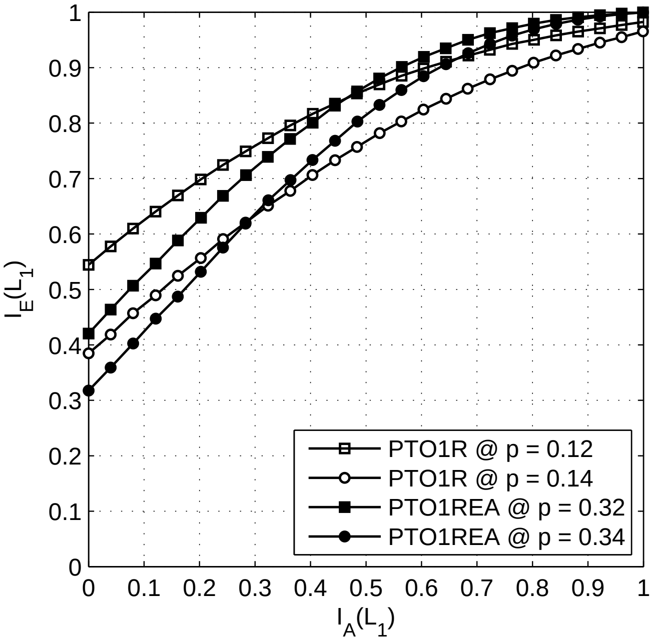


Figure 6.14: Comparison of the inner EXIT curves of both unassisted and entanglement-assisted QCCs, labeled as PTO1R and PTO1REA respectively. PTO1R is the inner code of Table 6.2, while PTO1REA is the same as PTO1R but requires 2 e-bits.

6.6.3 Optimized Quantum Turbo Code Design

The QTC design of [95, 88] characterized in Figure 6.12 exhibits a large area between the inner and outer decoder's EXIT curves. The larger the 'open-tunnel' area, the farther the WER/QBER performance from the achievable noise limit p^* [68]. For an unassisted rate-1/9 code, the noise limit is $p^* = 0.16028$ according to the Hashing bound of Figure 6.1. Consequently, the design of Figure 6.12 operates within $\left[10 \times \log_{10}(\frac{0.125}{0.16028})\right] = 1.1$ dB of the noise limit. Various other distance spectra based QTCs investigated in [88], which exhibit a wide EXIT chart tunnel analogous to Figure 6.12, operate within 0.9 dB of the Hashing bound. For the sake of achieving a Hashing bound approaching performance, we minimize the area between the inner and outer EXIT curves, so that a narrow, but still marginally open tunnel exists at the highest possible depolarizing probability.

Design Objective: *Congenially with the rate-*$1/9$ *QTC of [88], find the optimal inner and outer components of a rate-*$1/9$ *QTC relying on a maximally-entangled inner code (recursive and non-catastrophic) and an unassisted outer code (non-catastrophic), both having a memory of* 3 *and a rate of* $1/3$*. The resultant QTC has an entanglement consumption rate of* $6/9$ *(or equivalently* $2/3$*), for which the corresponding noise limit is* $p^* = 0.3779$ *according to Eq. (6.3). Consequently, the optimized QTC should operate at a channel depolarizing probability close to the noise limit* $p^* = 0.3779$*.*

For the sake of finding the optimized inner and outer components, which minimize the area between

Coding rate	$R = 1/9$
Entanglement consumption rate	$\mathtt{E} = 6/9$
Interleaver length	$N = 3,000$ qubits
Iterations	$\mathtt{I} = 15$
Inner QCC	
Coding rate	$R_i = 1/3$
Entanglement consumption rate	$\mathtt{E}_i = 2/3$
Memory	3
Seed transformation	$U_i = \{4091, 3736, 2097, 1336, 1601, 279, 3093, 502, 1792, 3020, 226, 1100\}_{10}$
Outer QCC	
Coding rate	$R_o = 1/3$
Entanglement consumption rate	$\mathtt{E}_o = 0$
Memory	3
Seed transformation	$U_o = \{1048, 3872, 3485, 2054, 983, 3164, 3145, 1824, 987, 3282, 2505, 1984\}_{10}$

Table 6.3: Simulation parameters of the optimized QTC, having a rate-1/3 EA-QCC (recursive and non-catastrophic) as the inner code, while an unassisted rate-1/3 QCC (non-catastrophic) is used as the outer code.

the corresponding EXIT curves at a depolarizing probability close to the noise limit, we randomly selected both the inner and outer encoders from the Clifford group according to the algorithm of [162]. Based on this design criterion, we found the optimal inner and outer code pair, whose seed transformations (decimal representation) are given by:

$$U_i = \{4091, 3736, 2097, 1336, 1601, 279, 3093, 502, 1792, 3020, 226, 1100\}_{10}; \tag{6.67}$$

$$U_o = \{1048, 3872, 3485, 2054, 983, 3164, 3145, 1824, 987, 3282, 2505, 1984\}_{10}. \tag{6.68}$$

Code parameters of our optimized design are also summarized in Table 6.3.

Figure 6.15 shows the EXIT curves of our optimized QTC at the convergence threshold of $p = 0.35$. As observed in Figure 6.15, a marginally open EXIT tunnel exists between the two curves, which would facilitate decoding trajectories to reach the $(1, 1)$ point of perfect convergence. Hence, our optimized QTC has a convergence threshold of $p = 0.35$, which is only $\left[10 \times \log_{10}(\frac{0.35}{0.3779})\right] = 0.3$ dB from the noise

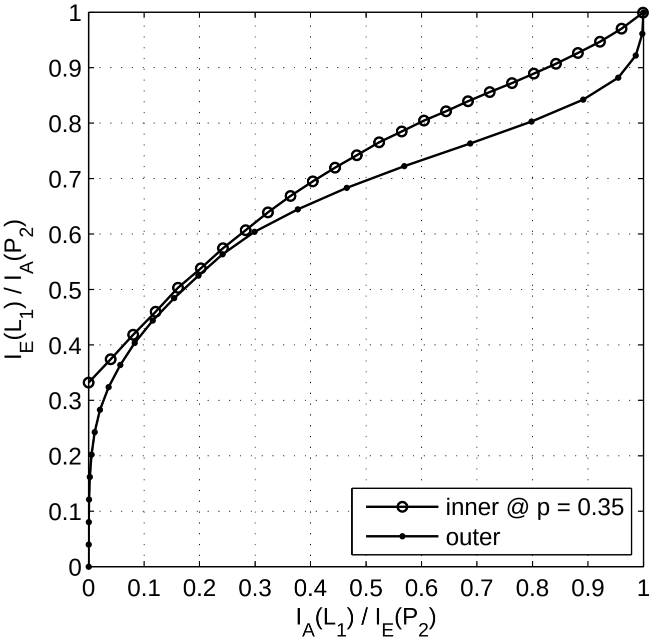


Figure 6.15: The EXIT curves of the optimized rate-1/9 QTC. The parameters of the inner and outer QCC are listed in Table 6.3.

limit of $p^* = 0.3779$. We have quantified the corresponding achievable performance in Figure 6.16, where the interleaver length was increased from 1500 to 12,000. All other simulation parameters are kept the same as that of Table 6.3. We have recorded both the WER as well as the QBER performance. Analogous to the classical turbo codes, increasing the interleaver length improves the attainable performance for $p < 0.35$, hence providing a better performance at the cost of an increased delay, which was also pointed out in Figure 6.2 in the context of our conflicting design challenges.

In Figure 6.17, we further compare our optimized design, using the parameters of Table 6.3, to the "PTO1REA-PTO1R" configuration of [88], whose parameters are summarized in Table 6.4. Both configurations employ a rate-1/3 EA-QCC (recursive and non-catastrophic) as the inner code, while an unassisted rate-1/3 QCC (non-catastrophic) is used as the outer code. For the "PTO1REA-PTO1R" design, the turbo cliff region emerges around $p = 0.31$, which is within 0.9 dB of the noise limit. It may also be observed in Figure 6.17 that our optimized design outperforms the "PTO1REA-PTO1R" in terms of having a better convergence threshold, albeit at the cost of a higher error-floor. This is because our optimized outer code has a low minimum distance of only 3. Its truncated distance spectrum is as follows:

$$\begin{aligned} D(x) =& 2x^3 + 19x^4 + 108x^5 + 530x^6 + 2882x^7 + 14179x^8 + 62288x^9 + 243234x^{10} + \\ & 845863x^{11} + 1165784x^{12} + 2501507x^{13} + 744394x^{14}. \end{aligned}$$

By contrast, the truncated distance spectrum of "PTO1R", which has a minimum distance of 5, is

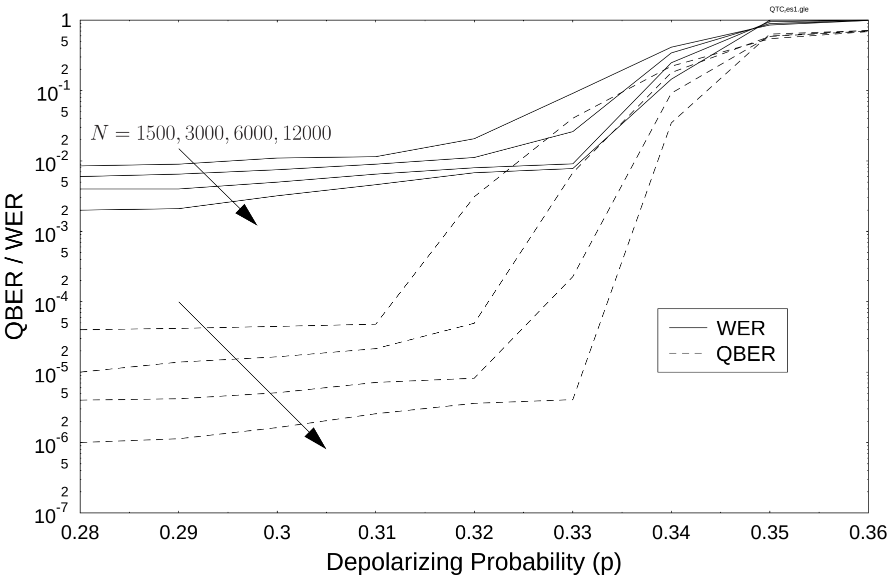


Figure 6.16: Achievable QBER and WER performance of the optimized QTC of Figure 6.6 as the interleaver length (N) is increased from 1500 through $12,000$. All other simulation parameters are summarized in Table 6.3.

given by [88]:

$$D(x) = 11x^5 + 47x^6 + 253x^7 + 1187x^8 + 6024x^9 + 30529x^{10} + 153051x^{11} + 771650x^{12}.$$

Consequently, as gleaned from Figure 6.17, the "PTO1REA-PTO1R" configuration has a much lower error floor ($< 10^{-6}$), since the outer code "PTO1R" has a higher minimum distance. However, this enlarges the area between the inner and outer decoder's EXIT curves; thus, driving the performance farther away from the achievable capacity. Hence, there is a trade-off between the minimization of the error floor and achieving a Hashing bound approaching performance. More specifically, while the distance-spectrum based design primarily aims for achieving a lower error floor, the EXIT-chart based design strives for achieving a near-capacity performance. Note that the error floor can also be reduced by employing a longer interleaver. Furthermore, we invoked a maximum of 8 decoding iterations for the "PTO1REA-PTO1R" design, while a maximum of 15 iterations were used for our optimized design. Again, this can be attributed to the wide EXIT tunnel that exists between the inner and outer decoders' EXIT curves of the 'PTO1REA-PTO1R" construction. More explicitly, the wider the gap between the inner and outer decoders' EXIT curves, the faster the decoding convergence, since a lower number of decoding iterations are invoked for reaching the $(1, 1)$ point of perfect convergence. In this section, we did not intend to carry out an exhaustive code search for finding the best code. Instead, our intention was to optimize our design in terms of its convergence threshold for the sake of demonstrating the explicit benefit of our EXIT-chart technique conceived for approaching the Hashing bound. Nevertheless, in the next section, we have conceived QIRCC, which reduces the error floor.

Coding rate	$R = 1/9$
Entanglement consumption rate	$\mathtt{E} = 6/9$
Interleaver length	$N = 3,000$ qubits
Iterations	$\mathtt{I} = 8$
Inner QCC	
Coding rate	$R_i = 1/3$
Entanglement consumption rate	$\mathtt{E}_i = 2/3$
Memory	3
Seed transformation	$U_i = \{1355, 2847, 558, 2107, 3330, 739, 2009, 286, 473, 1669, 1979, 189\}_{10}$
Outer QCC	
Coding rate	$R_o = 1/3$
Entanglement consumption rate	$\mathtt{E}_o = 0$
Memory	3
Seed transformation	$U_o = \{1355, 2847, 558, 2107, 3330, 739, 2009, 286, 473, 1669, 1979, 189\}_{10}$

Table 6.4: Simulation parameters of the "PRO1REA-PTO1R" configuration of [95, 88], which is used as a benchmark. Similar to the QTC of Table 6.3, inner code is a rate-1/3 EA-QCC (recursive and non-catastrophic), while the outer code is an unassisted rate-1/3 QCC (non-catastrophic).

6.7 Quantum Irregular Convolutional Codes

In this section we further pursue our design objective, i.e. *to find the optimal outer code $\mathcal{C}$ having a coding rate R_o, which gives the best match with the given inner code, i.e. whose EXIT curve yields a marginally open tunnel with the given inner decoder's EXIT curve at a depolarizing probability close to the Hashing bound.* For the sake of achieving this objective, a feasible design option could be to create the outer EXIT curves of all the possible convolutional codes to find the optimal code $\mathcal{C}$, which gives the best match, as we did in Section 6.6.3. To circumvent this exhaustive code search, in this section we propose to invoke QIRCCs for achieving EXIT-curve matching.

Similar to the classical Irregular Convolutional Code (IRCC) of [119], our proposed QIRCC employs a family of $\mathcal{Q}$ subcodes $\mathcal{C}_q$, $q \in \{1, 2, \ldots, \mathcal{Q}\}$, for constructing the target code $\mathcal{C}$. Due to its inherent flexibility, the resultant QIRCC provides a better EXIT-curve match than any single code, when used as the outer component in the concatenated structure of Figure 6.6. The q^{th} subcode has a coding rate

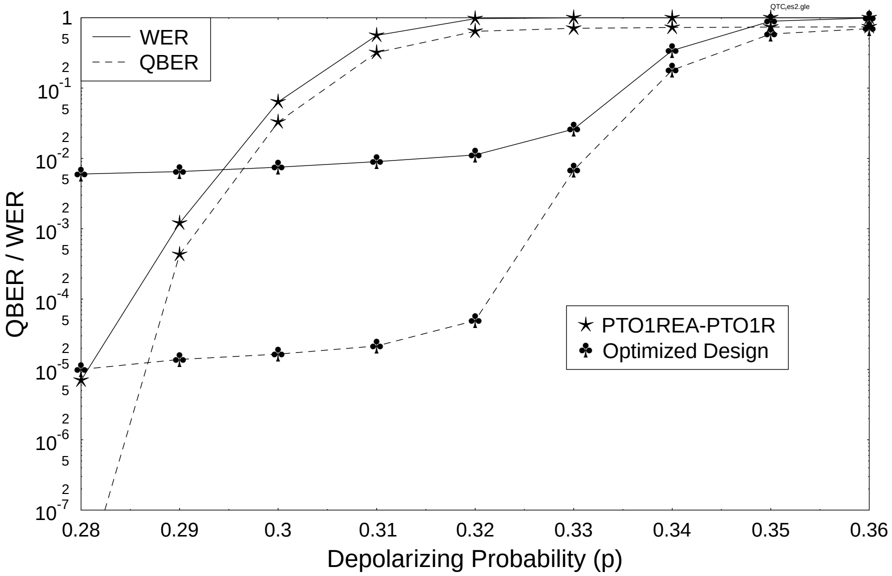


Figure 6.17: Comparison of the achievable QBER/WER performance of our optimized QTC having the parameters of Table 6.3 and the "PRO1REA-PTO1R" configuration of [95, 88], whose parameters are summarized in Table 6.4.

of r_q and it encodes a specifically designed fraction of the original information qubits to $\varrho_q N$ encoded qubits. Here, N is the total length of the coded frame. More specifically, for a $\mathcal{Q}$-subcode IRCC, ϱ_q is the q^{th} IRCC weighting coefficient satisfying the following constraints [118, 119]:

$$\sum_{q=1}^{\mathcal{Q}} \varrho_q = 1 \ , \ R_o = \sum_{q=1}^{\mathcal{Q}} \varrho_q r_q \ , \ \varrho_q \in [0,1], \forall q \ , \tag{6.69}$$

which can be conveniently represented in the following matrix form:

$$\begin{aligned} \begin{pmatrix} 1 & 1 & \dots & 1 \\ r_1 & r_2 & \dots & r_{\mathcal{Q}} \end{pmatrix} \begin{pmatrix} \varrho_1 & \varrho_2 \dots & \varrho_{\mathcal{Q}} \end{pmatrix}^T &= \begin{pmatrix} 1 \\ R_o \end{pmatrix} \\ \mathbf{r}\, \boldsymbol{\varrho} &= \mathbf{R} \ . \end{aligned} \tag{6.70}$$

Hence, as shown in Figure 6.18, the input stream is partitioned into $\mathcal{Q}$ sub-frames[10], which are assembled back into a single N-qubit stream after encoding.

In the context of classical IRCCs, the subcodes $\mathcal{C}_q$ are constructed from a mother code [118, 119]. More specifically, high-rate subcodes are obtained by puncturing the mother code, while the lower rates are obtained by adding more generators. However, unlike classical codes, puncturing is not easy to implement for quantum codes, since the resultant punctured code must satisfy the symplectic criterion, as in [163]. In this context, in order to design the constituent subcodes of our proposed QIRCC, we

[10] This is only true if all subcodes are active. If $\varrho_q = 0$ for the q^{th} subcode, then $\mathcal{C}_q$ is not activated. Therefore, the input stream is only divided among the active subcodes.

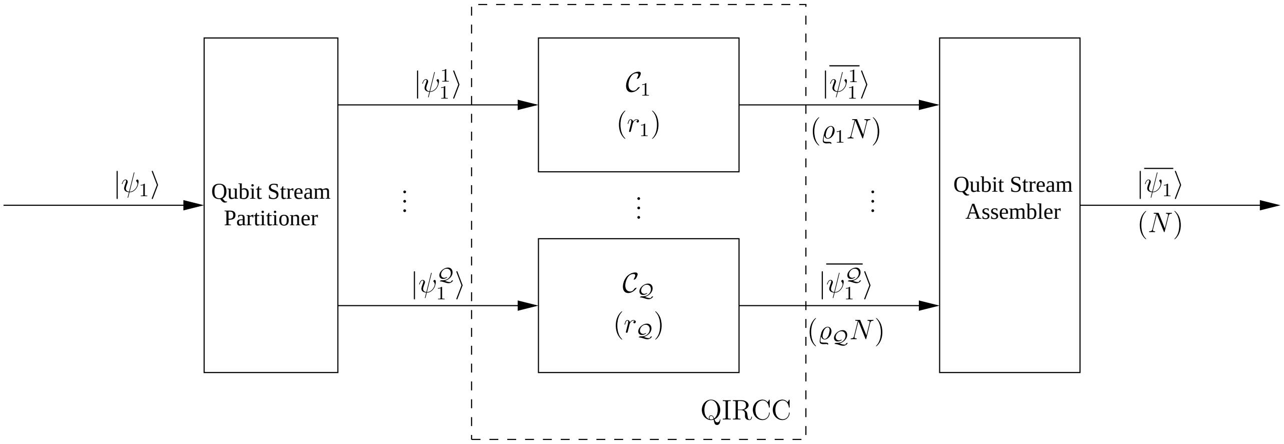

$|\psi_1\rangle$
Qubit Stream Partitioner
$|\psi_1^1\rangle$
$|\psi_1^{\mathcal{Q}}\rangle$
$\mathcal{C}_1$
(r_1)
$\mathcal{C}_{\mathcal{Q}}$
$(r_{\mathcal{Q}})$
$|\overline{\psi_1^1}\rangle$
$(\varrho_1 N)$
$|\overline{\psi_1^{\mathcal{Q}}}\rangle$
$(\varrho_{\mathcal{Q}} N)$
QIRCC
Qubit Stream Assembler
$|\overline{\psi_1}\rangle$
(N)

Figure 6.18: Structure of a $\mathcal{Q}$-subcode QIRCC encoder.

selected 5 strong randomly-constructed memory-3 quantum convolutional codes with quantum code rates $\{1/4, 1/3, 1/2, 2/3, 3/4\}$, which met the non-catastrophic criterion of [55]. More explicitly, for the sake of achieving a random construction for the Clifford encoder specifying the quantum convolutional code, we used the classical random walk algorithm over the $(n+m)$-qubit Clifford group as in [162]. The seed transformations of the resultant subcodes having rates $\{1/4, 1/3, 1/2, 2/3, 3/4\}$ are given below:

$$\begin{aligned}
U_1 &= \{9600, 691, 11713, 4863, 1013, 6907, 1125, 828, 10372, 6337, 5590, 11024, 12339, 3439\}_{10},\\
U_2 &= \{3968, 1463, 2596, 3451, 1134, 3474, 657, 686, 3113, 1866, 2608, 2570\}_{10},\\
U_3 &= \{848, 1000, 930, 278, 611, 263, 744, 260, 356, 880\}_{10},\\
U_4 &= \{529, 807, 253, 1950, 3979, 2794, 956, 1892, 3359, 2127, 3812, 1580\}_{10},\\
U_5 &= \{62, 6173, 4409, 12688, 7654, 10804, 1763, 15590, 6304, 3120, 2349, 1470, 9063, 4020\}_{10}. \qquad (6.71)
\end{aligned}$$

The EXIT curves of these QIRCC subcodes are shown in Figure 6.19, whereby the memory-3 subcodes of Eq. (6.71) are indicated by solid lines. Furthermore, in order to facilitate accurate EXIT curve matching with a sufficiently versatile and diverse set of inner EXIT functions, we also selected 5 weak randomly-constructed memory-1 subcodes for the same range of coding rates, i.e. $\{1/4, 1/3, 1/2, 2/3, 3/4\}$. The corresponding seed transformations are as follows:

$$\begin{aligned}
U_6 &= \{475, 194, 526, 422, 417, 988, 426, 611, 831, 84\}_{10},\\
U_7 &= \{26, 147, 149, 99, 112, 184, 64, 139\}_{10},\\
U_8 &= \{37, 55, 58, 35, 57, 54\}_{10},\\
U_9 &= \{57, 248, 99, 226, 37, 93, 244, 54\}_{10},\\
U_{10} &= \{469, 634, 146, 70, 186, 969, 387, 398, 807, 452\}_{10}, \qquad (6.72)
\end{aligned}$$

and their EXIT curves are plotted in Figure 6.19 with the dotted lines. It must be mentioned here that the range of coding rates chosen for the QIRCC subcodes can be expanded such that the EXIT curves

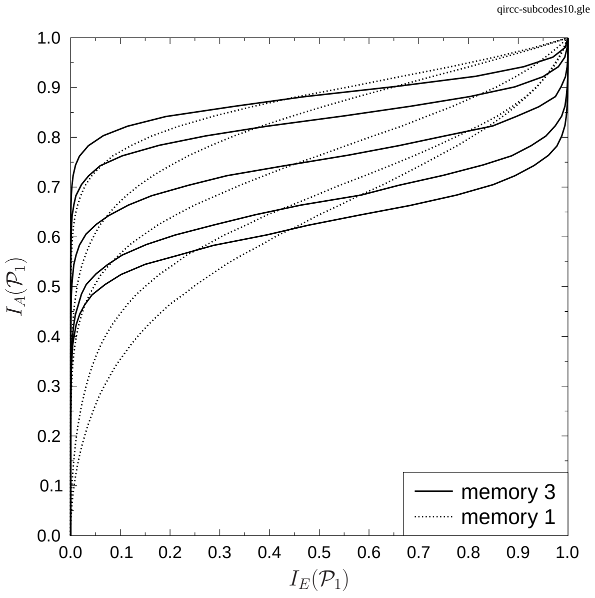


Figure 6.19: Outer EXIT curves (inverted) of our QIRCC subcodes having code rates $\{1/4, 1/3, 1/2, 2/3, 3/4\}$ for both memory-3 as well as memory-1.

cover a larger portion of the EXIT plot, which would further improve curve matching. However, this increases the encoding and decoding structual complexity.

Based on our proposed QIRCC, relying on the 10 subcodes specified by Eq. (6.71) and (6.72), the input bit stream is divided into 10 fractions corresponding to the 10 different-rate subcodes. The specific optimum fractions to be encoded by these codes are found by dynamic programming. More specifically, since the QCCs belong to the class of linear codes, the EXIT curves of the 10 subcodes, given in Figure 6.19, are superimposed onto each other after weighting by the appropriate fraction-based weighting coefficients, which are determined by minimizing the area of the open EXIT-tunnel. To elaborate a little further, the transfer function of the QIRCC is given by the weighted sum of each subcode's transfer function as shown below:

$$I_E(P_1) = T_1[I_A(P_1)] = \sum_{q=1}^{\mathcal{Q}} \varrho_q \; T_1^q \left[I_A(P_1)\right], \tag{6.73}$$

where $T_1^q\left[I_A(P_1)\right]$ is the transfer function of the q^{th} subcode. For a given inner EXIT curve and outer code rate R_Q, we employ the curve matching algorithm of [118, 119] for optimizing the weighting coefficients $\boldsymbol{\varrho}$ of our proposed QIRCC such that the square of the error between the inner and inverted outer EXIT curves is minimized subject to Eq. (6.69). More explicitly, the error function may be modeled as:

$$e(i) = T_2[i, p] - T_1^{-1}[i], \tag{6.74}$$

where $p = (p^* - \epsilon)$ given that p^* is the noise limit defined by the Hashing bound and ϵ is an arbitrarily small number. The corresponding matrix-based notation may be formulated as [118, 119]:

$$\mathbf{e} = \mathbf{b} - \mathbf{A}\boldsymbol{\varrho}, \tag{6.75}$$

where we have:

$$\mathbf{b} = \begin{pmatrix} T_2[i_1, p] \\ T_2[i_2, p] \\ \vdots \\ T_2[i_{\mathrm{N}}, p] \end{pmatrix}, \text{and}$$

$$\mathbf{A} = \begin{pmatrix} T_1^{1^{-1}}[i_1] & T_1^{2^{-1}}[i_1] & \ldots & T_1^{\mathcal{Q}^{-1}}[i_1] \\ T_1^{1^{-1}}[i_2] & T_1^{2^{-1}}[i_2] & \ldots & T_1^{\mathcal{Q}^{-1}}[i_2] \\ \vdots & \vdots & \vdots & \vdots \\ T_1^{1^{-1}}[i_{\mathrm{N}}] & T_1^{2^{-1}}[i_{\mathrm{N}}] & \ldots & T_1^{\mathcal{Q}^{-1}}[i_{\mathrm{N}}] \end{pmatrix}. \tag{6.76}$$

Here, N denotes the number of sample points such that $i \in \{i_1, i_2, \ldots, i_{\mathrm{N}}\}$ and it is assumed that $\mathrm{N} > \mathcal{Q}$. Furthermore, the error should be greater than zero for the sake of ensuring an open tunnel, i.e. we have:

$$e(i) > 0, \;\; \forall i \in [0, 1]. \tag{6.77}$$

The resultant cost function, i.e. sum of the square of the errors, is given by [118]:

$$\mathcal{J}(\varrho_1, \ldots, \varrho_{\mathcal{Q}}) = \int_0^1 e(i)^2 di, \tag{6.78}$$

which may also be written as:

$$\mathcal{J}(\boldsymbol{\varrho}) = \mathbf{e}^T \mathbf{e}. \tag{6.79}$$

The overall process may be encapsulated as follows:

$$\boldsymbol{\varrho}_{opt} = \arg\min_{\boldsymbol{\varrho}} \mathcal{J}(\boldsymbol{\varrho}), \tag{6.80}$$

subject to Eq. (6.69) and (6.77), which is a convex optimization problem. The unconstrained optimal solution for Eq. (6.80) is found iteratively using steepest descent approach with a gradient of $\partial\mathcal{J}(\boldsymbol{\varrho})/\partial\boldsymbol{\varrho} = 2\mathbf{e}$, which is then projected onto the constraints defined in Eq. (6.69) and (6.77). Further details of this optimization algorithm can be found in [68, 118, 119].

6.8 Results and Discussions II

For the sake of demonstrating the curve matching capability of our proposed QIRCC, we designed a rate-1/9 concatenated code relying on the rate-1/3 EA inner code of Table 6.4, namely "PTO1REA" [95, 88], in conjunction with our proposed QIRCC, which constitutes the outer code. Since the entanglement consumption rate of "PTO1REA" is 2/3, the resultant code has an entanglement consumption

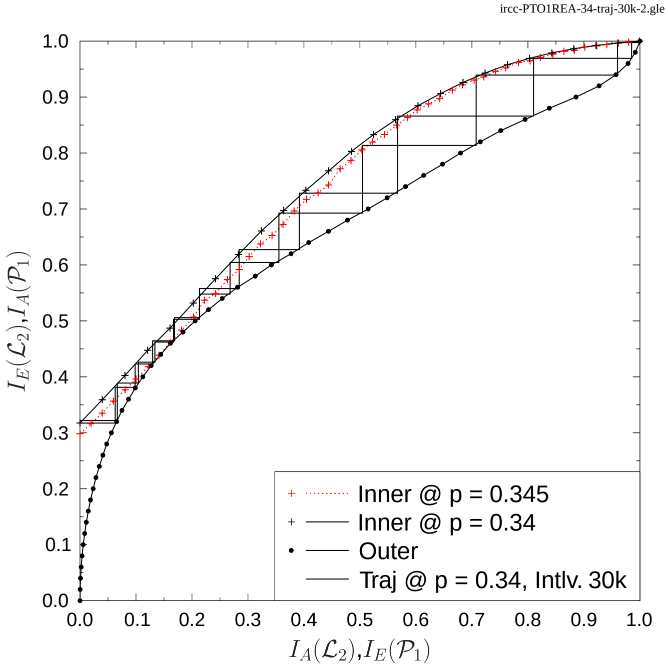


Figure 6.20: EXIT curves of the concatenated rate-1/9 system, with PTO1REA as the inner code and QIRCC as the outer, at $p = 0.345$ and $p = 0.34$. The parameters of the inner and outer code are summarized in Table 6.5, while the trajectories are plotted for a $30,000$-qubit long interleaver.

rate of 6/9, for which the corresponding noise limit is $p^* = 0.3779$ according to Eq. (6.3) [88]. Furthermore, since we intend to design a rate-1/9 system with a rate-1/3 inner code, we have $R_o = 1/3$. Hence, for a target coding rate of 1/3, we used the optimization algorithm discussed in Section 6.7 for the sake of finding the optimum weighting coefficients of Eq. (6.80) at the highest possible depolarizing probability $p = p^* - \epsilon$. It was found that we only need to invoke two subcodes out of the 10 possible subcodes, based on $\boldsymbol{\varrho} = [0\ 0\ 0\ 0\ 0.168\ 0.832\ 0\ 0\ 0\ 0]^T$, for attaining a marginally open tunnel, which occurs at $p = 0.345$, as shown in Figure 6.20 using the simulation parameters of Table 6.5. Hence, the resultant code has a convergence threshold of $p = 0.345$, which is only $\left[10 \times \log_{10}(\frac{0.345}{0.3779})\right] = 0.4$ dB from the noise limit of $p^* = 0.3779$. Figure 6.20 also shows two decoding trajectories at $p = 0.34$ for a $30,000$ qubit long interleaver. As gleaned from the figure, the decoding trajectories closely follow the EXIT curves reaching the $(1, 1)$ point of perfect convergence.

The corresponding WER performance curves recorded for our QIRCC-based optimized design, having the simulation parameters of Table 6.5, are seen in Figure 6.21, where the WER is reduced upon increasing the number of iterations from 1 through to 15. More explicitly, our code converges to a low WER for $p \leq 0.345$. Thus, this convergence threshold matches the one predicted using EXIT charts in Figure 6.20. More explicitly, since the EXIT chart tunnel closes for $p > 0.345$, the system fails to converge, if the depolarizing probability is increased beyond 0.345. Hence, the performance

Coding rate	$R = 1/9$
Entanglement consumption rate	$\mathtt{E} = 6/9$
Interleaver length	$N = 3,000$ qubits
Iterations	$\mathtt{I} = 15$
Inner QCC	
Coding rate	$R_i = 1/3$
Entanglement consumption rate	$\mathtt{E}_i = 2/3$
Memory	3
Seed transformation	$U_i = \{1355, 2847, 558, 2107, 3330, 739, 2009, 286, 473, 1669, 1979, 189\}_{10}$
Outer QIRCC	
Coding rate	$R_o = 1/3$
Entanglement consumption rate	$\mathtt{E}_o = 0$
Weighting coefficients	$\boldsymbol{\varrho} = [0\ 0\ 0\ 0\ 0.168\ 0.832\ 0\ 0\ 0\ 0]^T$

Table 6.5: Simulation parameters of the QIRCC-aided configuration. Inner code is the same as that in Table 6.4, while the QIRCC is used as the outer code, whose weighting coefficients are optimized with the aid of EXIT charts for the sake of approaching the Hashing bound.

does not improve upon increasing the number of iterations if the depolarizing probability exceeds the threshold. By contrast, when the depolarizing probability is below the threshold, the WER improves at each successive iteration. It should also be noted that the performance improves with diminishing returns at a higher number of iterations.

Figure 6.22 compares our QIRCC-based optimized design with the rate-1/9 “PTO1REA-PTO1R” configuration of [88]. An interleaver length of 3000 qubits was used. For the “PTO1REA-PTO1R” configuration, the turbo cliff region emerges around 0.31, which is within 0.9 dB of the noise limit. Therefore, our QIRCC-based design outperforms the “PTO1REA-PTO1R” configuration of [88]. More specifically, the “PTO1REA-PTO1R” configuration yields a WER of 10^{-3} at $p = 0.29$, while our design gives a WER of 10^{-3} at $p = 0.322$. Hence, our optimized design outperforms the ‘PTO1REA-PTO1R” configuration by about $\left[10 \times \log_{10}(\frac{0.29}{0.322})\right] = 0.5$ dB at a WER of 10^{-3}. It must be mentioned here that the “PTO1REA-PTO1R” configuration may have a lower error floor than our design for low value of p, yet our design exhibits a better performance in the turbo cliff region. We further compare our QIRCC-based optimized design with the exhaustive-search based optimized turbo code of Section 6.6.3 in Figure 6.22. Our QIRCC-based design does not provide any improvement in terms of its convergence threshold as compared to the optimized QTC of Section 6.6.3. However, our

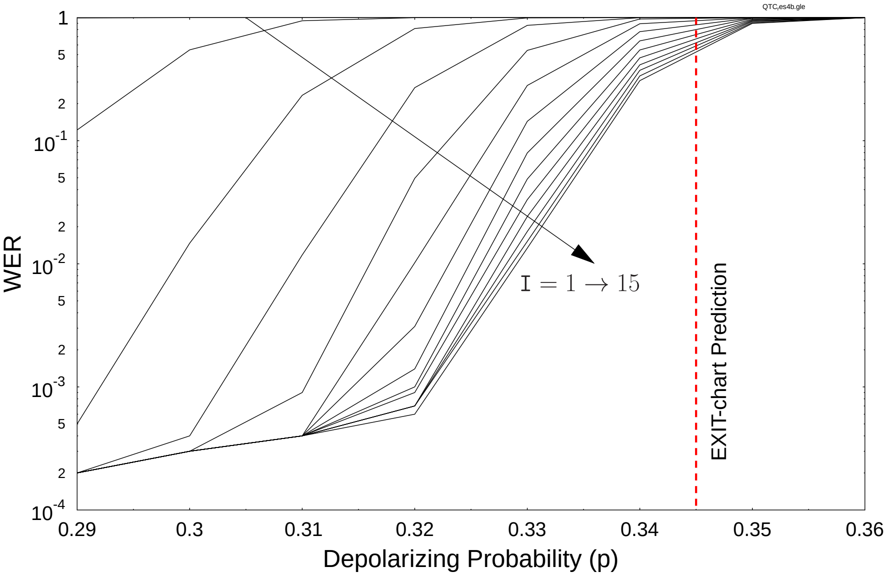


Figure 6.21: Achievable WER performance of our QIRCC-based optimized design, having the parameters of Table 6.5, as the number of iterations (I) is increased from 1 through 15.

QIRCC-based design has a much lower error rate associated with a lower error floor, as gleaned from Figure 6.22. Furthermore, the QIRCC-based design allows us to dispense with the exhaustive-search based optimization of Section 6.6.3. It must be pointed out here that we may accrue an improvement in the convergence threshold by increasing the number of QIRCC subcodes, which would facilitate better curve matching.

6.9 Summary and Conclusions

Powerful QECCs are required for stabilizing and protecting the fragile constituent qubits of both quantum computation as well as of communication systems against the undesirable decoherence. In line with the developments in the field of classical channel coding theory, this may be achieved by exploiting concatenated codes designs, which invoke iterative decoding. Therefore, in this chapter we have provided a slow-paced tutorial for designing Hashing bound approaching concatenated quantum codes using EXIT charts, which is based on the insights developed in Chapters 4 and 5.

We commenced our discourse by highlighting our design objectives in Section 6.2. In particular, we characterized the performance of an ideal code in terms of its channel depolarizing probability mitigating capacity, its coding rate and its entanglement consumption rate - all three of which are related to each other through the Hashing bound, or more specifically the Hashing region of Figure 6.1. We next presented the circuit based representation of QCCs in Section 6.3, which facilitates the degenerate iterative decoding of concatenated quantum codes. We also discussed the construction of

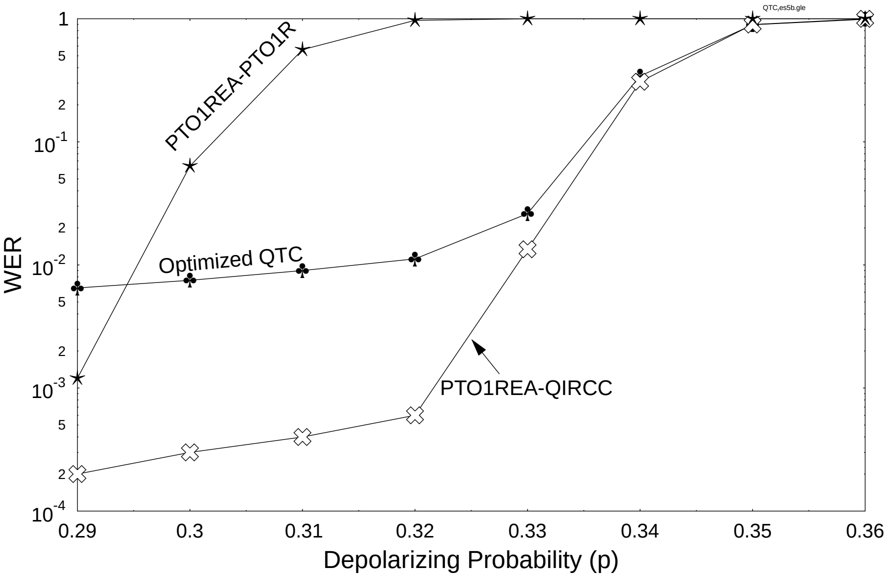


Figure 6.22: Comparison of the achievable WER performance of our QIRCC-based optimized QTC having the parameters of Table 6.5 with the "PRO1REA-PTO1R" configuration of [95, 88] (Table 6.4) and our exhaustive-search based optimized QTC of Section 6.6.3 (Table 6.3).

Clifford unitary encoder, which is completely specified by the Hadamard, phase and controlled-NOT gates of Eq. (6.12). We next presented our system model in Section 6.4.1, which relies on the circuit-based representation of a pair of concatenated QSCs. The degenerate iterative decoding approach, which is invoked in the system model of Section 6.4.1, is then detailed in Section 6.4.2. Unlike the classic syndrome-based MAP algorithm, which aims for finding the most likely error for a given syndrome, a degenerate MAP algorithm aims for finding the most likely error coset. More explicitly, the probabilities corresponding to degenerate errors are accumulated, as depicted in Eq. (6.49), in order to cater for the specific errors, which can be corrected by the same recovery operation. Finally, in Section 6.5, we extended the application of classical nonbinary EXIT charts to the circuit-based syndrome decoder of QTCs, to facilitate the EXIT-chart aided Hashing bound approaching design of QTCs. While the classical EXIT charts aim for modeling the *a-priori* information concerning the input bits of the inner encoder (and similarly the output bits of the outer encoder), the EXIT charts conceived for quantum codes have the objective of modeling the *a-priori* information concerning the corresponding error-sequence, i.e. the error-sequence related to the input qubits of the inner encoder (and similarly the error-sequence related to the output qubits of the outer encoder).

We evaluated the performance of our EXIT-chart based code design in Section 6.6. More specifically, we first established the accuracy of our EXIT chart approach in Section 6.6.1, where it was demonstrated that our EXIT-chart predictions of Figure 6.12 match the Monte-Carlo simulation results, recorded in Figure 6.13, to a reasonable degree. We next analyzed in Section 6.6.2 the behavior of both an unassisted (non-recursive) and an EA (recursive) inner QCC using EXIT charts. We demonstrated in Figure 6.14 that similar to their classical counterparts, the recursive inner quantum codes constitute families of QTCs having an unbounded minimum distance. In Section 6.6.3, we optimized the constituent inner and outer components of the QTC of Figure 6.6 using EXIT charts. In this context, our design guidelines for achieving a Hashing bound approaching performance may be summarized as follows:

- As discussed in the context of our design objectives in Section 6.2, we commence our design by determining the noise limit p^* for the desired code parameters, i.e the coding rate and the entanglement consumption rate of the resultant concatenated quantum code, using the capacity curves of Figure 6.1.

- We then proceed with the selection of the inner stabilizer code of Figure 6.6, which has to be both recursive as well as non-catastrophic, as argued in Section 6.4.1. Since the unassisted quantum codes cannot be simultaneously both recursive as well as non-catastrophic, we employ an EA code. Furthermore, the EA inner code of Figure 6.6 may be either derived from the family of known classical codes, as discussed in Chapter 4, or it may be constructed using the random Clifford operations, which were discussed in Section 6.3. At this point, the EXIT curves of Section 6.5 may be invoked for the sake of finding that specific inner code, which yields the largest area under its EXIT-curve at the noise limit p^*.

- Finally, we find the optimal non-catastrophic outer code of Figure 6.6, which gives the best EXIT-curve match to that of a carefully chosen inner code having the largest area under its EXIT curve. Our EXIT-chart aided design of Section 6.5 aimed for creating a narrow, but marginally open tunnel between the EXIT curves of the inner and outer decoders at the highest possible depolarizing probability, as demonstrated in the EXIT curves of Figure 6.15. The narrower the tunnel-area, the lower is the deviation from the Hashing bound, which may be quantified using Eq. (6.2).

Recall that the desired code structure may also be optimized on the basis of a range of conflicting design challenges, which were illustrated in Figure 6.2.

We demonstrated in Section 6.6.3 that in contrast to the distance spectra based QTCs of [88], whose convergence threshold is within 0.9 dB of the Hashing bound, the convergence threshold of our optimized QTC is within 0.3 dB of the noise limit, as evidenced in Figure 6.15. However, the resultant optimized QTC has a higher error floor and worse WER performance, which was recorded in Figure 6.16 and compared to the performance of [88] in Figure 6.17. Nevertheless, the error floor may be reduced upon increasing the interleaver length, which would in turn incur a longer delay. For example, upon increasing the interleaver length from $N = 1500$ to $N = 12,000$ in Figure 6.16, the WER floor was reduced from WER $= 10^{-2}$ to WER $= 2 \times 10^{-3}$, which is about an order of magnitude. It is important to mention here that we did not intend to carry out an exhaustive code search for finding the best code, but our aim was rather to find a design optimized in terms of improving the

Code Structure	Distance from Capacity		
	Convergence Threshold	Performance at **WER** $= 10^{-2}$	Performance at **WER** $= 10^{-3}$
QTC of [88]	0.9 dB	1.1 dB	1.15 dB
Exhaustive-search based optimized QTC	0.3 dB	0.8 dB	Error floor $< 10^{-3}$
QIRCC-based QTC	0.4 dB	0.6 dB	0.7 dB

Table 6.6: Comparison of the performance of the QTC of [88], i.e. PRO1REA-PTO1R, with our exhaustive-search optimized QTC of Section 6.6.3 and the PRO1REA-QIRCC of Section 6.8. All the three configurations rely on a rate-1/3 EA-QCC (recursive and non-catastrophic) as the inner code, while the outer code is an unassisted rate-1/3 QCC (non-catastrophic).

convergence threshold for the sake of demonstrating the explicit benefit of our EXIT-chart based design in terms of approaching the Hashing bound.

For the sake of further facilitating the Hashing bound approaching code design, we proposed the structure of QIRCC in Section 6.7, which constitutes the outer component of a concatenated quantum code. The proposed QIRCC allows us to dispense with the exhaustive code-search methods, since it can be dynamically adapted to match any given inner code using EXIT charts. We constructed a 10-subcode QIRCC and used it as an outer code in concatenation with the non-catastrophic and recursive EA-QCC of [95, 88]. We optimized our design using EXIT charts, as demonstrated in Figure 6.20, while the achievable performance was recorded in Figure 6.21. Finally, we compared the designs of Table 6.3, Table 6.4 and Table 6.5 in Figure 6.22. The corresponding results are summarized in Table 6.6.

In the spirit of designing iterative code structures, in this chapter we conceived Hashing bound approaching designs for concatenated quantum codes. Pursuing further the iterative code design, in the next chapter we will focus our efforts on the family of Quantum Low Density Parity Check (QLDPC) codes from the perspective of code design as well as on the associated iterative decoding algorithms.

Chapter 7

Quantum Low Density Parity Check Codes

7.1 Introduction

In Chapter 6, we conceived Hashing bound approaching designs for concatenated quantum codes. Pursuing further the quest for designing iterative code structures, in this chapter we focus on the Quantum Low Density Parity Check (QLDPC) codes. In particular, the astounding near-capacity performance of the classical Low Density Parity Check (LDPC) codes [69, 164, 165, 166, 167], despite an affordable decoding complexity, has inspired the community to design QLDPC codes. The sparseness of the QLDPC matrix is of particular interest in the quantum domain, because it requires only a small number of interactions per qubit during the error correction procedure. More specifically, each qubit is only involved in a small fraction of the stabilizer generators (or the rows of the QLDPC matrix). Consequently, the sparse nature of QLDPC matrix inherently limits the propagation of errors, hence facilitating ‘fault tolerant’ decoding.

QLDPC codes belong to the family of Quantum Stabilizer Codes (QSCs) [38, 39], which is a generalized formalism for designing quantum codes from any arbitrary classical binary and quaternary codes, as discussed in Chapter 4. However, this transfiguration from the classical to the quantum domain imposes a stringent symplectic criterion on the parent classical codes, which brings with it various design challenges. Against this backdrop, in this chapter we survey the evolution of QLDPC code designs, focusing on the various code constructions to conceive powerful QLDPC codes from the known families of classical LDPC codes. We also review the syndrome-based iterative decoding algorithms invoked for QLDPC codes. Finally, we contribute to these developments by conceiving a new code design and an improved decoding algorithm. More specifically, our novel contributions are as follows [1, 2]:

- We have proposed a radically new class of high-rate row-circulant Quasi Cyclic (QC) QLDPC codes. More specifically, our proposed unassisted non-dual-containing CSS QLDPC codes can be constructed from arbitrary row-circulant classical QC-LDPC matrices, which are known to generate efficient short and moderate length high-rate classical QC-LDPC codes.

- We have conceived a modified non-binary decoding algorithm for homogeneous Calderbank-Shor-Steane (CSS)-type QLDPC codes, which is capable of mitigating the impact of the unavoidable length-4 cycles..
- We demonstrate that the Uniformly-Reweighted Belief Propagation (URW-BP) technique of [168, 169] may also be invoked for further improving the attainable performance.

The rest of the chapter is organized as follows. We commence with a review of QLDPC code designs in Section 7.2, while a range of powerful decoding techniques are discussed in Section 7.3. We next propose our new code design in Section 7.4, whose performance is evaluated in Section 7.5. Finally, we present our proposed decoding algorithm in Section 7.6, while in Section 7.7 we detail the reweighted belief propagation. The corresponding simulation results are presented in Section 7.8, while Section 7.9 concludes our discourse.

7.2 Quantum LDPC Code Designs

Analogous to classical LDPC codes, which belong to the family of linear block codes discussed in Section 4.2, QLDPC codes are inherently stabilizer codes, which may be characterized using an equivalent classical Parity Check Matrix (PCM) H of Eq. (4.30). More specifically, an $[n, k]$ QLDPC code having a coding rate of $R_Q = k/n$ is equivalent to a $(2n, n+k)$ binary LDPC code having a coding rate of $R_c = (n+k)/2n$. We may divide the QLDPC codes into three main categories on the basis of the general global structure of the associated PCM H, namely Calderbank-Shor-Steane (CSS) codes, non-CSS codes and Entanglement-Assisted (EA) codes, as summarized in Figure 7.1. The CSS-type constructions may also be classified as dual-containing and as non-dual-containing codes. Let us now take a look at each of these categories individually.

7.2.1 Calderbank-Shor-Steane Codes

Ideally, any two classical binary LDPC codes, which meet the symplectic criterion, may be used for constructing a CSS-based QLDPC code. However, randomly choosing the constituent pair of classical codes is not feasible, because finding two sparse codes, which satisfy the stringent symplectic constraint, is highly unlikely. This motivated Postol [46] to conceive the first example of a CSS-based non-dual-containing QLDPC code from a small $(15, 7)$ finite geometry based classical LDPC code in 2001. More specifically, in Postol's code, the PCM of a finite geometry based cyclic classical LDPC code constitutes the H'_z of Eq. (4.35), while H'_x is derived from H'_z, so that the symplectic criterion is satisfied, i.e. we have $H'_z {H'_x}^T = 0$. Since both the constituent PCMs, i.e. H'_z and H'_x, are cyclic, this facilitates the implementation of the encoder. However, Postol did not develop a generalized method

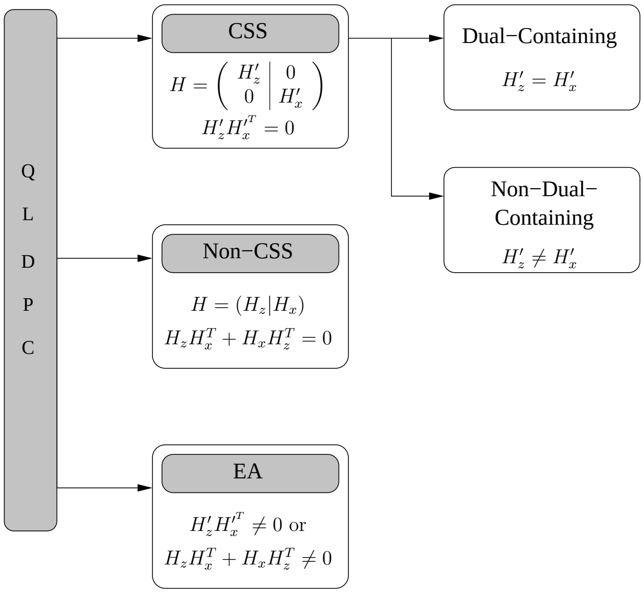


Figure 7.1: Classification of QLDPC codes.

for his proposed design, which could facilitate the construction of QLDPC codes from any arbitrary finite geometry based classical LDPC codes. This gap was filled by Mackay *et al.* in [47], where several systematic constructions were developed for the CSS-based QLDPC codes by restricting the designs to the dual-containing structure.

Before proceeding with the constructions of [47], let us take a look at the symplectic condition of Eq. (4.34) in the context of the dual-containing QLDPC codes. Recall from Section 4.3.2.1 that the symplectic criterion of Eq. (4.34) reduces to $H'_z H'^T_z = 0$ for the dual-containing QLDPC codes, which have $H'_x = H'^T_z$. This in turn implies that the PCM of a classical LDPC code may only be used for constructing a dual-containing QLDPC code if:

1. it has an even row weight; and
2. every pair of rows has an even number of overlapping 1's, which we may term as an 'even overlap'.

By contrast, good classical LDPC codes must have at most a single overlapping 1 between every pair of rows for the sake of avoiding length-4 cycles because short cycles of length-4 impair the performance of the associated decoding algorithm. Consequently, the 'even overlap' condition results in unavoidable cycles of length 4 in the resultant PCM, as depicted in Figure 7.2 for a random binary PCM H'_z given by[1]:

$$H'_z = \begin{pmatrix} 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 0 \end{pmatrix}. \tag{7.1}$$

[1] This is a random example for illustrating the impact of an even number of overlaps. The row weight must be even. Furthermore, the H'_z of Eq. (7.1) may not be a good classical code.

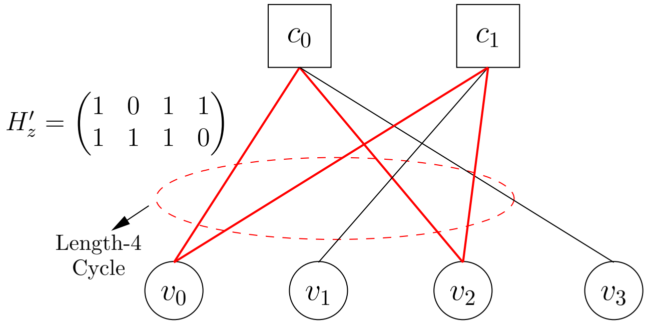

$H'_z = \begin{pmatrix} 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 0 \end{pmatrix}$
c_0
c_1
Length-4 Cycle
v_0
v_1
v_2
v_3

Figure 7.2: Tanner graph of H'_z. An 'even overlap' between the rows of H'_z results in a length-4 cycle.

Furthermore, the constraint $H'_z H'^T_z = 0$ also implies that the code-space of the underlying classical code must contain its dual. Hence, the resultant code contains codewords having a weight equal to the row weight ρ. Therefore, the minimum distance of the classical dual-containing code is upper-bounded by ρ. Surprisingly, this upper-bound does not exist for quantum codes due to the degenerate nature of quantum errors. Recall from Section 4.3.3 that the n-tuple channel error pattern acting on the codewords of a QSC, may be classified as:

1. **Detected Error Patterns:** These error patterns anti-commute with the stabilizers of the code, yielding a non-trivial syndrome.

2. **Harmful Undetected Error Patterns:** This class of error patterns commutes with the stabilizers. Consequently, these error patterns are harmful, because they map one valid codeword onto another; thus, corrupting the codeword without triggering a non-trivial syndrome. Harmful undetected error patterns are attributed to the small minimum distance of the code.

3. **Harmless Undetected Error Patterns:** This is a unique class of error patterns, which do not have a classical analogue. Similar to the 'harmful undetected error patterns', these error patterns also commute with the stabilizers, but they are harmless in the context of quantum codes. This is because these are the degenerate errors, which belong to the stabilizer group, and therefore do not corrupt the state of the valid codewords. More explicitly, for dual-containing CSS codes, the harmless undetected error patterns lie in the code-space of the dual code $C_1^{\perp}$, as depicted in Figure 7.3. It must be mentioned here that although the harmless undetected errors do not affect the minimum distance of the resultant quantum code, they lead to the 'symmetric degeneracy error' in the iterative decoding procedure of QLDPC codes, which will be discussed in Section 7.3.3.

Bicycle codes, which were proposed by Mackay *et al.* in [47], marked the first major breakthrough towards the realization of CSS-based dual-containing quantum LDPC codes. The proposed code design relies on a semi-random/semi-structured construction, which satisfies the dual-containing constraint by deliberately imposing a global structure on the constituent PCM. A bicycle code having a row

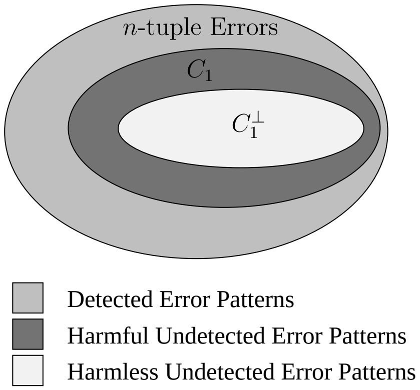

n-tuple Errors
C_1
$C_1^\perp$
Detected Error Patterns
Harmful Undetected Error Patterns
Harmless Undetected Error Patterns

Figure 7.3: Error pattern classification for dual-containing CSS codes.

weight of ρ, a block length of n and $(n-k)$ stabilizers is constructed using a random sparse $n/2 \times n/2$ cyclic matrix C_m, which has a row weight of $\rho/2$. The non-zero entries in C_m can be chosen either randomly or using a difference set satisfying the property that every difference (modulo $n/2$) occurs at most once in the set. This matrix C_m is then used for constructing a base matrix H_0, which is a concatenation of C_m and its transpose, i.e. we have:

$$H_0 = \left(C_m, C_m^T\right). \tag{7.2}$$

Consequently, H_0 is a dual-containing code satisfying the 'even overlap' constraint, because every overlap that occurs in C_m may also be found in C_m^T. Furthermore, since H_0 is an $n/2 \times n$ matrix, the resultant dual-containing quantum LDPC code has a coding rate $R_Q = 0$ (or equivalently $R_c = 1/2$). To achieve a non-zero coding rate, k rows of H_0 are discarded, so that the column weights of the resultant $(n-k) \times n$ PCM H_z' are as uniform as possible. This code design offers flexibility in choosing the code parameters, i.e. ρ, n and k. However, the minimum distance of the resultant code is upper-bounded by ρ. This is because the discarded rows of H_0 are all codewords of weight ρ, which are not contained in the dual, and therefore contribute to the harmful undetected error patterns.

Mackay *et al.* also proposed unicycle codes in [47], which are derived from perfect difference sets[2]. The perfect difference set property implies that all pairs of rows of the PCM must have a single

[2] A perfect difference set characterized on the additive group of size n has the unique property that every integer from 1 to $n-1$ may be expressed as a difference of two integers in the set (modulo n) in exactly one way. By contrast, in the plain difference sets, every difference occurs at most once, i.e. either it may not occur or will occur only once. For example, the set $\{1, 2, 4\}$ forms a perfect difference set for the group of size 7 because every integer from 1 to 6 can be expressed as the difference of two elements in the difference set, i.e. we have:

$$(1-2)\text{mod } 7 = 6, \quad (1-4)\text{mod } 7 = 4, \quad (2-1)\text{mod } 7 = 1,$$
$$(2-4)\text{mod } 7 = 5, \quad (4-1)\text{mod } 7 = 3, \quad (4-2)\text{mod } 7 = 2.$$

overlapping 1. Since we need an 'even overlap' to achieve a dual-containing structure, the PCM is extended by adding an extra column having all logical ones. Hence, every pair of rows in the resultant PCM have two overlapping 1's, which result in a single length-4 cycle between every pair of rows. Thus, an (n,k) PCM is transformed into a dual-containing $(n+1,k+1)$ PCM, which has a row weight of $(\rho+1)$ (where ρ is the row weight of the initial matrix and must be odd) and whose column weights are all ρ, except for the last 'all-one' column. Mackay *et al.* also suggested that the unique structure of unicycle codes may be exploited for avoiding the length-4 cycles during the decoding procedure [47]. More explicitly, a unicycle code may be viewed as a superposition of two codes, i.e. one having an 'all-zero' column at the end and the other having an 'all-one' column. For the sake of avoiding the short cycles, each of the two codes is decoded separately using the sum product algorithm [47]. If both decoders return a valid codeword, the codeword which has the maximum likelihood is chosen. Hence, an improved decoding procedure is conceived at the cost of an increased decoding complexity. Furthermore, the minimum distance of the unicycle codes constructed using difference sets is upper-bounded by the row weight, because the resultant code has codewords of weight ρ, which do not lie in the dual. Since the choice of n, k and ρ for perfect difference sets is limited, this design does not offer much flexibility in choosing the code parameters. By contrast, bicycle codes can be constructed from any arbitrary cyclic classical LDPC.

To extend the application of Mackay's unicycle codes to a wider range of code parameters, Aly [73] exploited the classical type-II Euclidean Geometry (EG) LDPC codes of [170]. Similar to the perfect difference sets, a classical type-II EG LDPC code having a PCM $H_{\text{EG-II}}$ has the unique characteristic that all pairs of rows have a single overlapping value of 1. Consequently, Aly suggested that the code characterized by an $(n-k)\times n$ matrix $H_{\text{EG-II}}$ may be converted into a dual-containing code in the following two ways:

1. If the row weight of $H_{\text{EG-II}}$ is odd, then similar to the unicycle codes, an 'all-one' column $\mathbf{1}$ is appended to $H_{\text{EG-II}}$, i.e. we have:
$$H'_z = (H_{\text{EG-II}} \mid \mathbf{1})\,. \tag{7.3}$$
2. If the row weight of $H_{\text{EG-II}}$ is even, then $\mathbf{1}$ is appended to $H_{\text{EG-II}}$ for the sake of ensuring an 'even overlap', while an identity matrix $\mathbf{I}$ of size $(n-k)\times(n-k)$ is appended to make the row weight even, i.e. we have:
$$H'_z = (H_{\text{EG-II}} \mid \mathbf{1} \mid \mathbf{I})\,. \tag{7.4}$$

The resultant codes offer beneficial high coding rates. However, they have an upper-bounded minimum distance of at least $(\gamma+1)$, where γ denotes the column weight.

Unicycle code construction was further explored by Djordjevic [74] for designing Quasi-Cyclic (QC) high-rate dual-containing QLDPC codes from the Balanced Incomplete Block Design (BIBD) based classical LDPC codes [171, 172], which have a minimum distance of at least $(\gamma+1)$, where γ denotes the column weight. More specifically, the BIBD[3] is characterized by the parameter λ. A BIBD-based

[3] BIBD(v, b, r, k, λ) distributes all the v elements (or points) of a set V into b subsets (or blocks) of size k such that,

LDPC code has exactly λ overlaps between every pair of rows. Since good classical LDPC codes must have at most a single row overlap, λ is set to 1 for designing classical LDPC codes with a girth of at least 6. Consequently, analogous to the perfect difference set based classical LDPC codes, each pair of rows has a single overlapping value of 1, which can be made even by imposing the unicycle code structure on the PCM. Djordjevic also designed dual-containing LDPC codes by using BIBDs associated with an even λ. Unfortunately, the even λ based QLDPC codes failed to outperform the unicycle based BIBD constructions [74].

Since all the aforementioned dual-containing constructions resulted in an upper-bounded minimum distance, the quest for the construction of unbounded QLDPC codes continued. Pursuing this objective, another non-trivial class of dual-containing QLDPC codes was proposed by Mackay *et al.* in [70], which was derived from Cayley graphs. These codes were further investigated by Couvreur *et al.* in [71, 72], where it was formally shown that the lower bound on the minimum distance of the resultant code is a logarithmic function of the code length, thus the minimum distance can be improved by extending the codeword (or block) length, albeit again, only logarithmically. However, this is achieved at the cost of an increased decoding complexity imposed by the escalating row weight, which also increases logarithmically with the block length. Furthermore, Cayley graph based designs may be viewed as a special class of the topological codes [173, 174, 175][4], which are already known to have growing minimum distances.

Let us recall that the dual-containing QLDPC codes have unavoidable short cycles, which impair the performance of the decoding algorithm. Hence, even if dual-containing QLDPC codes having an unbounded minimum distance are designed, they are unlikely to surpass the performance of their non-dual-containing counterparts. Therefore, in the midst of these activities, Lou *et al.* [75, 76] rekindled the interest in CSS-based non-dual-containing QLDPC codes by invoking the classical Low Density Generator Matrix (LDGM) codes for code construction. More specifically, since both the generator matrix and the PCM of an LDGM code are sparse, they can be used as the components of a CSS code. Let $\tilde{G}$ and $\tilde{H}$ be the generator matrix and PCM, respectively, of an (n, k) LDGM code. Then the resultant CSS code may be formulated as follows:

$$H = \begin{pmatrix} \tilde{H} & \mathbf{0} \\ \mathbf{0} & \tilde{G} \end{pmatrix}. \tag{7.5}$$

- each pair of elements occurs in exactly λ of the blocks,
- every element occurs in exactly r blocks, and
- the number of elements in each block k is small as compared to the size v of set V; thus, giving it the name "incomplete".

Let us consider a set V of seven numbers, which is given by $V = \{1, 2, 3, 4, 5, 6, 7\}$. Then, the blocks $\{1, 2, 4\}$, $\{2, 6, 5\}$, $\{3, 4, 6\}$, $\{4, 5, 7\}$, $\{1, 5, 6\}$, $\{2, 6, 7\}$ and $\{1, 3, 7\}$ constitute the BIBD(7,7,3,3,1) since there are 7 elements (v) in the set V which are distributed among 7 blocks (b), each element appears in 3 blocks (r), each block has 3 elements (k) and each pair of elements occur in 1 block (λ).

[4] Topological constructions are beyond the scope of this work.

Since $\tilde{H}$ is an $(n-k) \times n$ matrix, while $\tilde{G}$ is a $k \times n$ matrix, the resultant PCM H is an $n \times 2n$ matrix. Consequently, the corresponding QLDPC code has a coding rate of zero. Lou *et al.* [75, 76] suggested that this may be avoided by applying linear row operations both to $\tilde{G}$ as well as to $\tilde{H}$ for the sake of reducing their number of rows. Unfortunately, this row-reduction may in turn create short cycles in the resultant PCM. For the sake of avoiding the adverse impact of these short cycles, Lou *et al.* [75, 76] also conceived a modified Tanner graph, which requires code doping [176] for pushing the iterative decoding process towards convergence. Hence, an improved performance is achieved at the cost of an increased decoding complexity.

Unfortunately, the constituent codes of all the aforementioned CSS constructions, both those of the dual-containing as well as of the non-dual-containing codes, suffer from the presence of length-4 cycles. To dispense with these short cycles, Hagiwara *et al.* [77, 177] conceived a unique class of non-dual containing QC-QLDPC codes, which have a girth of at least 6. More specifically, let us consider a circulant matrix T having a size of $LP/2 \times LP/2$, $\rho = L/2$ and $\gamma = L$, which is given by [177]:

$$T = \begin{pmatrix} t_0 & t_1 & \ldots & t_{L/2-1} \\ t_{L/2-1} & t_0 & \ldots & t_{L/2-2} \\ \vdots & & & \vdots \\ t_1 & t_2 & \ldots & t_0 \end{pmatrix}, \tag{7.6}$$

where t_i denotes the index of the circulant permutation matrix[5] of size P and $t_i \in [P_\infty] := \{0, 1, \ldots, P-1\} \cup \{\infty\}$. Hagiwara *et al.* have shown that H'_z and H'_x derived from the matrix T of Eq. (7.6) satisfy the symplectic criterion, if they have the form:

$$H'_z = (T_1, T_2) \quad \text{and} \quad H'_x = \left(-T_2^T, -T_1^T\right). \tag{7.7}$$

Furthermore, since row deletion does not perturb the symplectic criterion, rows may be deleted from H'_z and H'_x in order to achieve the desired coding rate. For the sake of ensuring a girth of 6, Hagiwara *et al.* relied upon algebraic combinatorics for designing the constituent circulant matrices T_1 and T_2, so that all the rows of H'_z as well as of H'_x have at most a single overlap. The bicycle codes of [47] may be viewed as a special case of this construction, i.e. when $P = 1$ and $T_2 = T_1^T$. Unfortunately,

[5] A circulant permutation matrix $I(1)$ of size P is given by:

$$I(1) = \begin{pmatrix} 0 & 1 & 0 & \ldots & 0 \\ 0 & 0 & 1 & \ldots & 0 \\ 0 & 0 & 0 & \ldots & 0 \\ \vdots & & & & \vdots \\ 1 & 0 & 0 & \ldots & 0 \end{pmatrix}.$$

More explicitly, $I(1)$ is a $P \times P$ identity matrix shifted to the right by one position. Therefore, $I(x)$ may be defined as a $P \times P$ identity matrix shifted to the right by x positions, where x is known as the index of the permutation matrix. Moreover, $x = 0$ defines an unshifted identity matrix, while $x = \infty$ is specially used to denote a zero matrix of size $P \times P$.

the resultant codes failed to outperform MacKay's bicycle codes [47] and their minimum distance is upper-bounded by the row weight.

Among all the dual-containing codes discussed above, Mackay's bicycle construction [47] offers the best performance at an affordable decoding complexity. However, the resultant performance is still not on par with that of the classical LDPC codes. For example, the rate-1/4 bicycle code of [47], having $n = 19,014$, operates within about 5.5 dB of the Hashing limit at a Word Error Rate (WER) of 10^{-3}. Furthermore, all the aforementioned codes have an upper-bounded minimum distance except for the Cayley graph based designs. In the quest for increasing the minimum distance and hence to approach the capacity, Hagiwara *et al.* extended the QC design of [77, 177] to Spatially-Coupled (SC) codes in [80], which outperformed their corresponding 'non-coupled' counterparts at the cost of a small coding rate loss. However, the performance still remained relatively far from the capacity. More specifically, the SC QC-QLDPC of [80], having a coding rate of 0.49 and a length of $n = 1,81,000$, operates within about 3.8 dB of the Hashing limit at a WER of 10^{-3}. Kasai *et al.* further contributed to these developments by deriving non-binary QC-QLDPC codes in [78, 79] from the design of [77, 177]. The resultant codes outperformed their binary counterparts at the cost of an increased decoding complexity. A rate-1/2 code, having a length of $n = 20,560$ and a Galois field of GF(2^{10}), was shown to operate within about 1.9 dB of the Hashing limit at a WER of 10^{-3}. The SC codes were further investigated by Andriyanova *et al.* in [178], where the constituent codes were derived from the classical LDGM codes as in [75, 76]. Analogous to the EA quantum codes, Andriyanova *et al.* assumed that some qubits are transmitted over a noiseless channel. Consequently, the resultant rate-1/4 LDGM-based SC-QLDPC codes, having a length of $n = 76,800$, succeeded in operating within about 1.7 dB of the Hashing limit at a WER of 10^{-3}. The assumption of having noiseless qubits was later eliminated in [179], whereby these qubits were protected by the error reducing Quantum Turbo Code (QTC) of [179], which resulted in a modest coding rate loss and in a moderately increased complexity for the overall code. It was shown that the performance of the resultant rate-1/2 QTC-assisted LDGM-based SC-QLDPC code, having a length of $n = 8,21,760$, is within about 0.7 dB of the Hashing limit at a WER of 10^{-3}. Figure 7.4 compares the achievable performance of the aforementioned codes, namely 'bicycle' code of [47], 'SC QC-QLDPC' code of [80], 'non-binary QC-QLDPC' code of [78, 79], 'LDGM-based SC-QLDPC' code of [178] and the 'QTC-assisted LDGM-based SC-QLDPC' code of [179], at a WER of 10^{-3}, which is benchmarked against the Hashing bound.

All the main contributions pertaining to CSS-based QLDPC codes are summarized in Figure 7.5.

7.2.2 Non-CSS Codes

Non-CSS stabilizer codes have the potential of exploiting any redundancy more efficiently than their CSS-based counterparts. For example, a CSS-based block code requires a block length of 7 qubits to correct a single bit-flip or phase-flip [34], while only 5 qubits are required for a non-CSS block code [36]. Consequently, Camara *et al.* [48, 49] proposed the construction of non-CSS (also called unrestricted) QLDPC codes. In contrast to most of the aforementioned dual-containing constructions, which satisfy

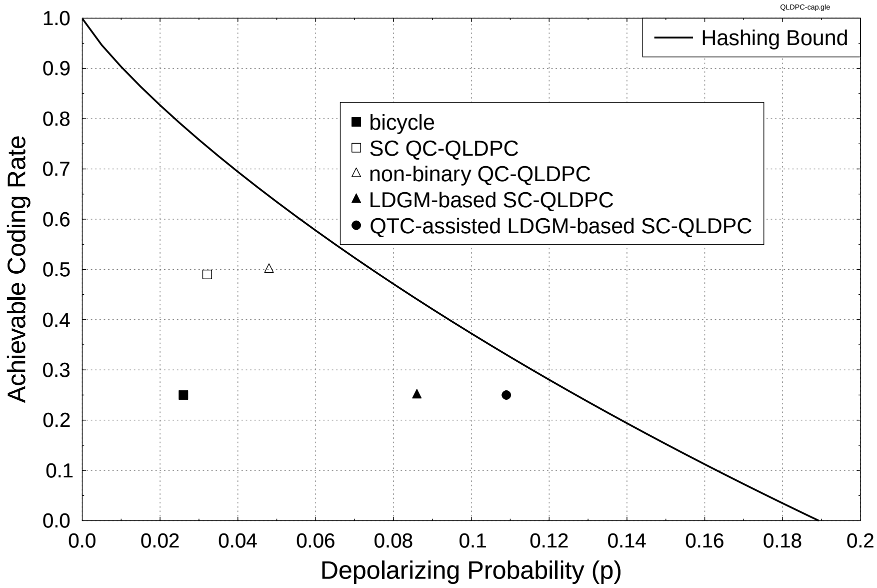
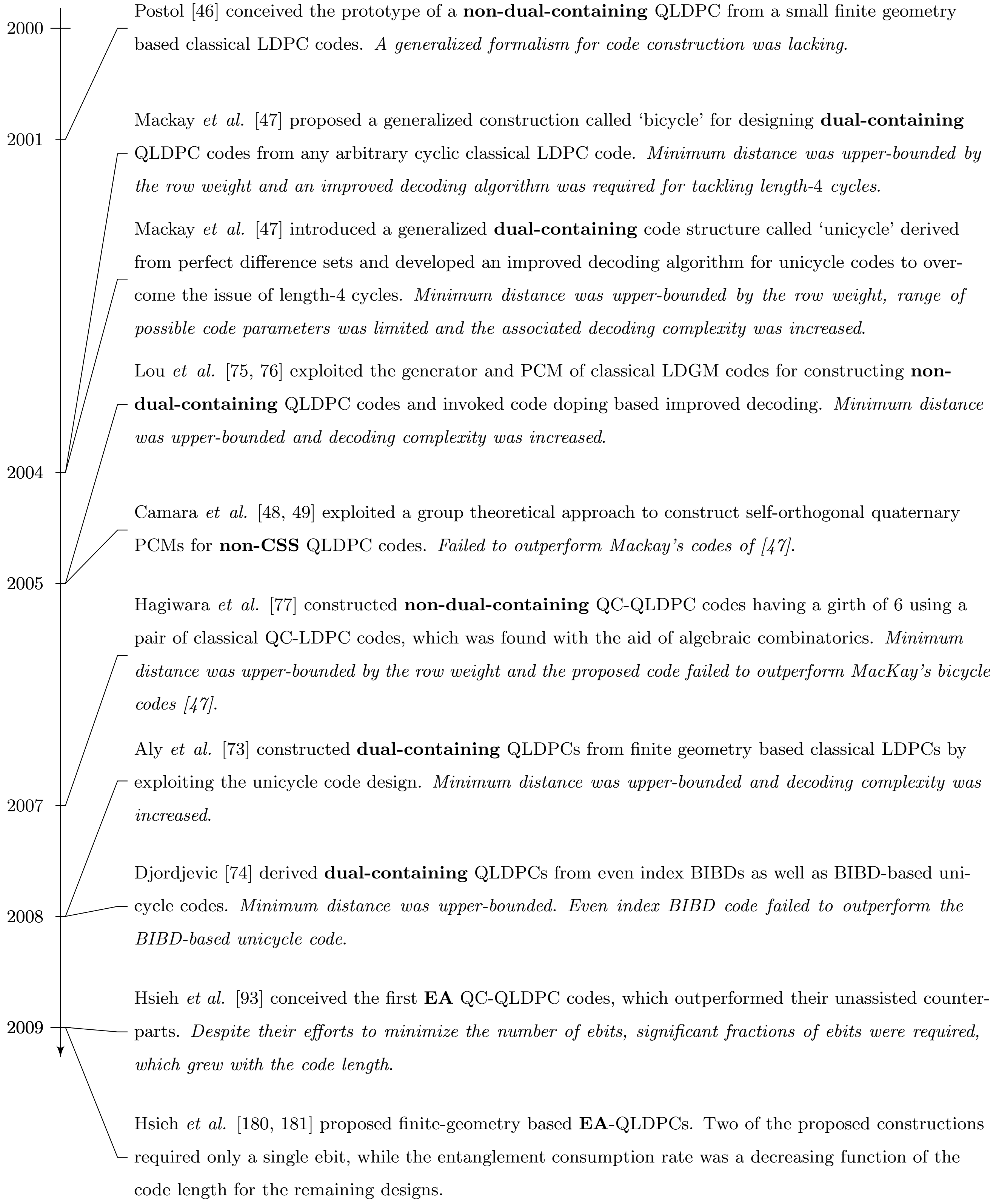


Figure 7.4: Achievable performance at a WER of 10^{-3} benchmarked against the Hashing bound for the 'bicycle' code ($R = 0.25$, $n = 19,014$) of [47], 'SC QC-QLDPC' code ($R = 0.49$, $n = 1,81,000$) of [80], 'non-binary QC-QLDPC' code ($R = 0.5$, $n = 20,560$, GF(2^{10})) of [78, 79], 'LDGM-based SC-QLDPC' code ($R = 0.25$, $n = 76,800$) of [178] and the 'QTC-assisted LDGM-based SC-QLDPC' code ($R = 0.25$, $n = 8,21,760$) of [179].

the symplectic criterion in their global code structure, the design conceived by Camara *et al.* aims at building the symplectic constraint into the local code structure. More specifically, since the PCM of a classical quaternary LDPC code can be mapped onto the generators of a QSC based on Eq. (4.39), Camara *et al.* developed a group theoretical approach for constructing self-orthogonal quaternary LDPC codes satisfying the symplectic criterion of Eq. (4.41). It was found that the Tanner graph of the resultant self-orthogonal quaternary PCM has cycles of length 4. However, these short cycles are imposed by the commutativity constraint. More specifically, every column of a quaternary PCM must contain at least two different non-zero entries, i.e Pauli-$\mathbf{X}$, Pauli-$\mathbf{Z}$, or Pauli-$\mathbf{Y}$, so that it can correct both phase-flips as well as bit-flips occurring on that qubit. On the other hand, any two rows of the PCM must have an even number of positions with different non-zero elements (or non-Identity Pauli operators). For example, let us consider a weight-2 column of a PCM, which is involved in two rows with a value of 1 and ω, respectively. Now to meet the commutativity constraint, these two rows must have another overlapping column having different non-zero entries; thus, creating cycles of length-4. Intuitively, these short cycles are also present in the PCM H of the CSS codes, when they are viewed in the quaternary domain. In fact, these cycles are excessive in the dual-containing CSS codes, which also have the additional cycles resulting from the dual-containing constraint[6]. The proposed non-CSS QLDPC codes of [48, 49] outperformed the bicycle codes in the waterfall region of their performance

[6] This is further discussed in Section 7.3.3.



Postol [46] conceived the prototype of a **non-dual-containing** QLDPC from a small finite geometry based classical LDPC codes. *A generalized formalism for code construction was lacking.*

Mackay *et al.* [47] proposed a generalized construction called 'bicycle' for designing **dual-containing** QLDPC codes from any arbitrary cyclic classical LDPC code. *Minimum distance was upper-bounded by the row weight and an improved decoding algorithm was required for tackling length-*4 *cycles.*

Mackay *et al.* [47] introduced a generalized **dual-containing** code structure called 'unicycle' derived from perfect difference sets and developed an improved decoding algorithm for unicycle codes to overcome the issue of length-4 cycles. *Minimum distance was upper-bounded by the row weight, range of possible code parameters was limited and the associated decoding complexity was increased.*

Lou *et al.* [75, 76] exploited the generator and PCM of classical LDGM codes for constructing **non-dual-containing** QLDPC codes and invoked code doping based improved decoding. *Minimum distance was upper-bounded and decoding complexity was increased.*

Camara *et al.* [48, 49] exploited a group theoretical approach to construct self-orthogonal quaternary PCMs for **non-CSS** QLDPC codes. *Failed to outperform Mackay's codes of [47].*

Hagiwara *et al.* [77] constructed **non-dual-containing** QC-QLDPC codes having a girth of 6 using a pair of classical QC-LDPC codes, which was found with the aid of algebraic combinatorics. *Minimum distance was upper-bounded by the row weight and the proposed code failed to outperform MacKay's bicycle codes [47].*

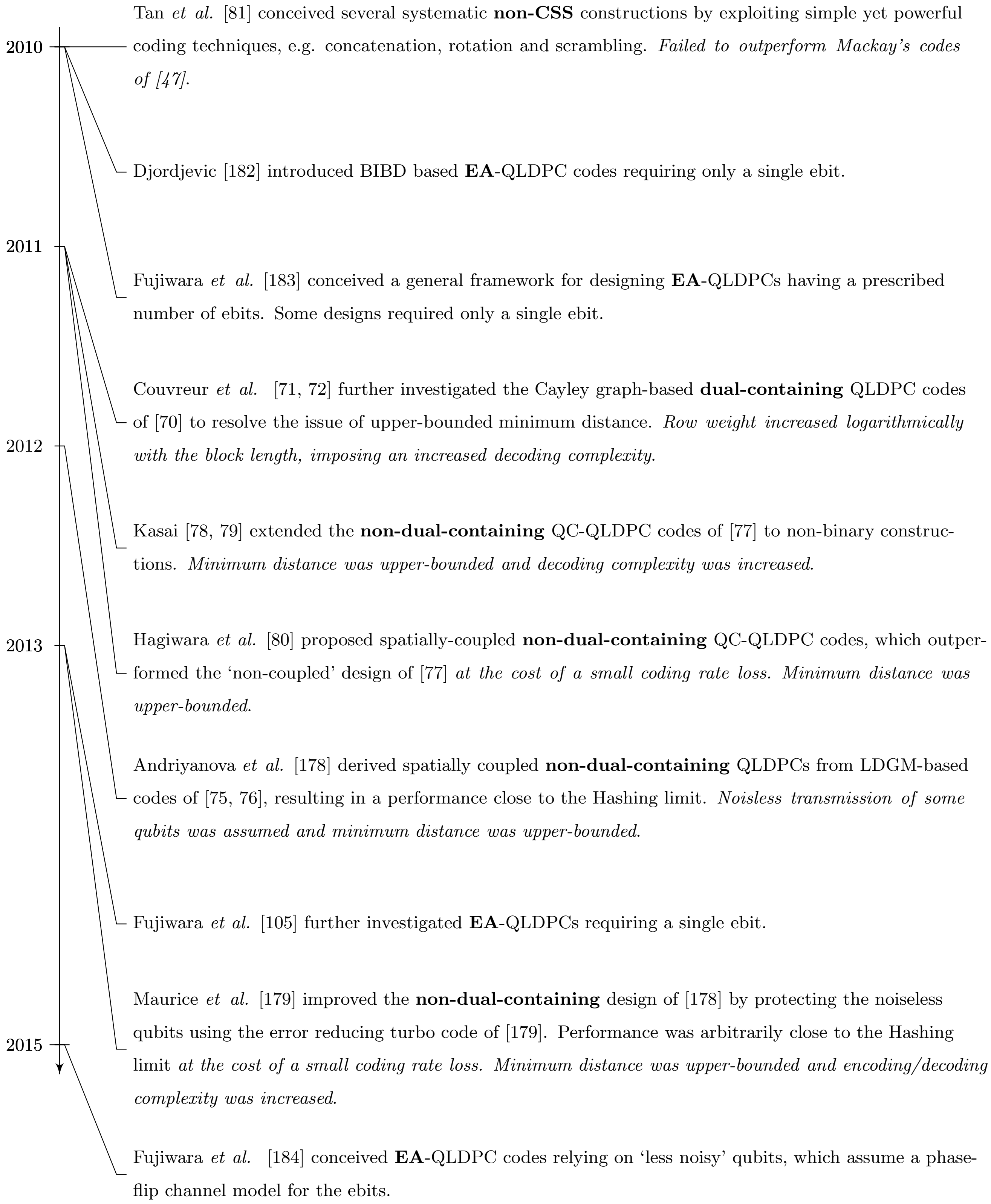
Aly *et al.* [73] constructed **dual-containing** QLDPCs from finite geometry based classical LDPCs by exploiting the unicycle code design. *Minimum distance was upper-bounded and decoding complexity was increased.*

Djordjevic [74] derived **dual-containing** QLDPCs from even index BIBDs as well as BIBD-based unicycle codes. *Minimum distance was upper-bounded. Even index BIBD code failed to outperform the BIBD-based unicycle code.*

Hsieh *et al.* [93] conceived the first **EA** QC-QLDPC codes, which outperformed their unassisted counterparts. *Despite their efforts to minimize the number of ebits, significant fractions of ebits were required, which grew with the code length.*

Hsieh *et al.* [180, 181] proposed finite-geometry based **EA**-QLDPCs. Two of the proposed constructions required only a single ebit, while the entanglement consumption rate was a decreasing function of the code length for the remaining designs.

Figure 7.5: (Continued on the next page)



Tan *et al.* [81] conceived several systematic **non-CSS** constructions by exploiting simple yet powerful coding techniques, e.g. concatenation, rotation and scrambling. *Failed to outperform Mackay's codes of [47].*

Djordjevic [182] introduced BIBD based **EA**-QLDPC codes requiring only a single ebit.

Fujiwara *et al.* [183] conceived a general framework for designing **EA**-QLDPCs having a prescribed number of ebits. Some designs required only a single ebit.

Couvreur *et al.* [71, 72] further investigated the Cayley graph-based **dual-containing** QLDPC codes of [70] to resolve the issue of upper-bounded minimum distance. *Row weight increased logarithmically with the block length, imposing an increased decoding complexity.*

Kasai [78, 79] extended the **non-dual-containing** QC-QLDPC codes of [77] to non-binary constructions. *Minimum distance was upper-bounded and decoding complexity was increased.*

Hagiwara *et al.* [80] proposed spatially-coupled **non-dual-containing** QC-QLDPC codes, which outperformed the 'non-coupled' design of [77] *at the cost of a small coding rate loss. Minimum distance was upper-bounded.*

Andriyanova *et al.* [178] derived spatially coupled **non-dual-containing** QLDPCs from LDGM-based codes of [75, 76], resulting in a performance close to the Hashing limit. *Noisless transmission of some qubits was assumed and minimum distance was upper-bounded.*

Fujiwara *et al.* [105] further investigated **EA**-QLDPCs requiring a single ebit.

Maurice *et al.* [179] improved the **non-dual-containing** design of [178] by protecting the noiseless qubits using the error reducing turbo code of [179]. Performance was arbitrarily close to the Hashing limit *at the cost of a small coding rate loss. Minimum distance was upper-bounded and encoding/decoding complexity was increased.*

Fujiwara *et al.* [184] conceived **EA**-QLDPC codes relying on 'less noisy' qubits, which assume a phase-flip channel model for the ebits.

Figure 7.5: Major contributions to the development of QLDPC codes. The 'code type' for each contribution is highlighted in **bold**, while the associated 'demerits' are marked in *italics*.

curve, while yielding a higher error floor due to their small minimum distance. It is expected that this non-CSS construction may have an unbounded minimum distance, thus yielding lower error floors, when the block length is sufficiently large. However, this was not explicitly proven in [48, 49].

Pursuing the same line of research, Tan *et al.* [81] were the first researchers to design the constituent PCMs H_z and H_x of a non-CSS code by invoking classical binary codes. More specifically, they conceived several systematic constructions for non-CSS QLDPC codes, which imposed both global as well as local structures on the underlying binary codes for the sake of satisfying the symplectic criterion. This is achieved by exploiting simple yet powerful coding techniques, which include concatenation, rotation and scrambling. The designed codes exhibit a better performance than the non-CSS codes of [48, 49]. However, they still failed to outperform Mackay's codes of [47]. In conclusion, the major milestones achieved in the domain of non-CSS QLDPC codes are summarized in Figure 7.5 .

7.2.3 Entanglement-Assisted QLDPC Codes

Efficient classical LDPC codes exist, which are known to approach the Shannon capacity for a large block size. For example, the optimized 1/2-rate classical LDPC code of [185] operates within 0.13 dB of the capacity limit for transmission over an Additive White Gaussian Noise (AWGN) channel at a Bit Error Rate (BER) of 10^{-6} using a code length of 10^{6}. More specifically, the turbo cliff of this LDPC code is merely 0.06 dB away from the Shannon capacity. This inspired researchers to achieve a comparable performance for QLDPCs. Unfortunately, the symplectic criterion, or more specifically the commutativity requirement of the stabilizers, limits the direct application of such efficient classical codes in the quantum domain. As discussed in Sections 7.2.1 and 7.2.2, only a limited class of classical codes, which conform to stringent local or global structural constraints, may be used as the constituents of a quantum code. This obstacle may be overcome by exploiting the EA quantum code designs of [90, 91, 92], which assist us in importing any classical code into the quantum domain. However, the pre-shared noiseless entangled qubits (ebits) of an EA code constitute a valuable resource, because maintaining a noiseless entangled state is not a trivial task. Consequently, a practically realizable code design should aim for minimizing the number of pre-shared noiseless ebits.

The first EA-QLDPC codes were conceived by Hsieh *et al.* in [93], whereby EA CSS-based QC-QLDPC codes were designed from their classical counterparts. Hsieh *et al.* chose the constituent circulant matrices of the classical QC code by ensuring that the number of ebits required is minimized. Despite their efforts, a significant number of these ebits was required, which grew with the code length. More importantly, these designs supported the conjecture that the high efficiency of EA codes should be attributed to the large fractions of pre-shared ebits. On a positive note, since the EA quantum codes of [93] shared the same attributes as the classical parent code, especially in terms of the girth and the minimum distance, these EA-QLDPC codes outperformed the state-of-the-art unassisted QLDPC codes. Working further in the direction of minimizing the number of pre-shared ebits, in [180, 181] Hsieh *et al.* conceived finite-geometry based EA-QLDPCs, whose 'entanglement consumption rate' decreases with the code length. Furthermore, two of these constructions required only a single ebit regardless of the code length; thus dispensing with the then prevailing apprehensions surrounding the family of EA-codes. It must be emphasized here that the proposed design does not impose any restrictions on the underlying finite geometry based classical LDPC codes of [170]. A more general framework conceived for designing the EA-QLDPCs, having a prescribed number of ebits, was presented in [183], which was derived from combinatorial design theory. Some of these designs required only a single ebit, despite having a high performance, a high coding rate and a low complexity. The necessary and sufficient conditions for designing single-ebit based EA-QLDPCs were further investigated in [105]. Moreover, BIBD based EA-QLDPC codes requiring only a single ebit were also identified in [182]. Recently, Fujiwara [186] introduced the notion of quantum codes relying on 'less noisy' (or 'reliable') qubits. More explicitly, unlike the EA formalism, which requires completely noiseless ebits, the framework of [186] assumes that these auxiliary qubits are subjected to a phase-flip channel, which is a more realistic noise model. In this spirit, Fujiwara *et al.* [184] conceived QLDPC codes relying on 'less noisy' qubits. The major contributions made in the domain of EA-QLDPC codes are summarized in Figure 7.5.

7.3 Iterative Decoding of Quantum LDPC Codes

Analogous to the classical LDPC codes, QLDPC codes invoke the classic Belief Propagation (BP) based decoding, also referred to as the Sum-Product Algorithm (SPA), which operates over the Tanner graph of the corresponding PCM. However, let us recall that qubits collapse upon measurement. Therefore, the syndrome-based version [187] of the classic codeword decoding has to be used for QLDPC codes. The underlying BP can be implemented both in the binary as well as in the quaternary domain, which are discussed in Sections 7.3.1 and 7.3.2, respectively.

7.3.1 Binary Decoding

A quantum depolarizing channel characterized by the depolarizing probability p is isomorphic to two independent Binary Symmetric Channels (BSCs) [47], i.e. one for phase-flips and the other for bit-flips, each having a cross-over probability of $2p/3$. More explicitly, based on the Pauli-to-binary isomorphism encapsulated in Eq. (4.29), a Pauli error $\mathcal{P} \in \mathcal{G}_n$ experienced by an n-qubit block transmitted over a depolarizing channel can be modeled by an effective error-vector P, which is a binary vector of length $2n$. The effective error P may be represented as $P = (P_z, P_x)$, where both P_z and P_x are n-bit long and represent $\mathbf{Z}$ and $\mathbf{X}$ errors, respectively. This implies that an $\mathbf{X}$ error imposed on the tth qubit will yield a 0 and a 1 at the tth and $(n+t)$th index of P, respectively. Similarly, a $\mathbf{Z}$ error imposed on the tth qubit will give a 1 and a 0 at the tth and $(n+t)$th index of P, respectively, while a $\mathbf{Y}$ error on the tth qubit will result in a 1 at both the tth as well as $(n+t)$th index of P. Since a depolarizing channel characterized by the probability p incurs $\mathbf{X}$, $\mathbf{Y}$ and $\mathbf{Z}$ errors with an equal probability of $p/3$,

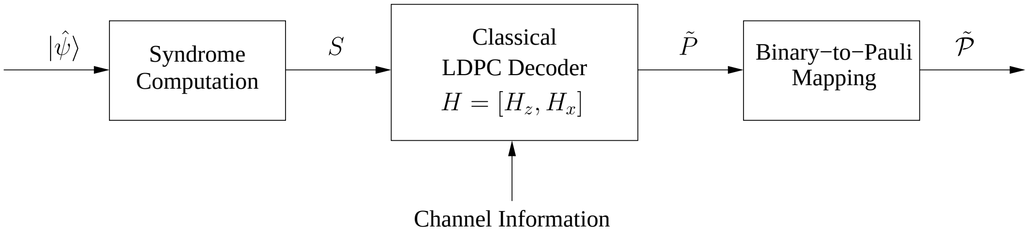


Figure 7.6: General schematic of a syndrome-based decoder for QLDPC codes.

the effective error-vector P reduces to two BSCs having a crossover probability of $2p/3$, where we have one channel for the **Z** errors and the other for the **X** errors.

Based on the aforementioned simplified notion, which ignores the correlation between the **X** and **Z** errors, QLDPC codes can be decoded by running the syndrome-based BP over the Tanner graph of the equivalent binary code having $H = (H_z|H_x)$ [81]. More explicitly, let S be the observed syndrome sequence, which is given by the symplectic product of H and P, as formulated below:

$$S = H \star P^T = H_z P_x^T + H_x P_z^T. \tag{7.8}$$

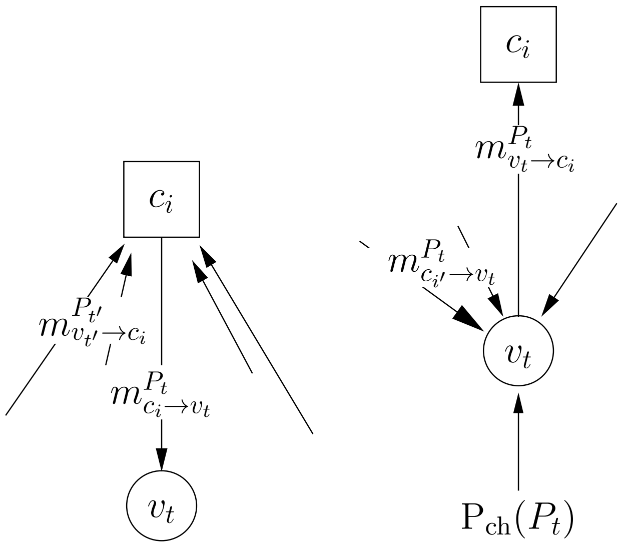
The observed syndrome S of Eq. (7.8) is fed to a classical syndrome-based LDPC decoder to estimate the most likely inflicted channel error $\tilde{P}$, as depicted in Figure 7.6. For an H of size $m \times 2n$, where we have $m = (n - k)$, the resultant estimated error vector $\tilde{P}$ is of length $2n$, whose first n bits are for the estimated phase errors $\tilde{P}_z$, while the other n bits indicate the estimated bit errors $\tilde{P}_x$. Finally, the $2n$-bit binary vector is mapped onto the n-qubit Pauli error $\tilde{\mathcal{P}}$ based on the mapping encapsulated in Eq. (4.29). More explicitly, the tth and $(n+t)$th value of $\tilde{P}$ are combined based on Eq. (4.29) to estimate the error inflicted on the tth qubit.

For CSS codes, we have $H_z = \begin{pmatrix} H'_z \\ \mathbf{0} \end{pmatrix}$ and $H_x = \begin{pmatrix} \mathbf{0} \\ H'_x \end{pmatrix}$. Consequently, the Tanner graph of the matrix H consists of two independent Tanner graphs corresponding to the matrices H'_z and H'_x. This in turn implies that **X** and **Z** errors can be decoded independently using the matrices H'_z and H'_x, respectively [47]. Hence, the QuBit Error Rate (QBER) of a CSS QLDPC code may be approximated by the sum of the BER of the two constituent classical codes. More explicitly, if p_e^x and p_e^z are the classical BERs for H'_z and H'_x, respectively, then the overall QBER is equivalent to $(p_e^x + p_e^z - p_e^x p_e^z) \approx (p_e^x + p_e^z)$, which reduces to $2p_e^z$ for a dual-containing CSS code having $H'_x = H'_z$.

For a binary $m \times 2n$ LDPC matrix H, the classical LDPC decoder of Figure 7.6 aims for finding the most likely error P of length $2n$ given the observed syndrome S, i.e. we have:

$$\tilde{P} = \underset{P \in (\mathbb{F}_2)^{2n}}{\arg\max}\, \mathrm{P}(P|S), \tag{7.9}$$

where $\mathrm{P}(P|S)$ is the probability of experiencing the error $P \in (\mathbb{F}_2)^{2n}$ imposed on the transmitted codewords, given that the syndrome of the received qubits $|\hat{\psi}\rangle$ is $S \in (\mathbb{F}_2)^m$. Unfortunately, Eq. (7.9)


c_i
$m^{P_{t'}}_{v_{t'}\rightarrow c_i}$
$m^{P_t}_{c_i\rightarrow v_t}$
v_t
c_i
$m^{P_t}_{v_t\rightarrow c_i}$
$m^{P_t}_{c_{i'}\rightarrow v_t}$
v_t
$\mathrm{P}_{\mathrm{ch}}(P_t)$

(a) Horizontal Message Exchange (b) Vertical Message Exchange

Figure 7.7: Belief Propagation (BP) algorithm. Check nodes and variable nodes are denoted by c_i and v_t, respectively.

defines an NP-complete problem [188]. A sub-optimal algorithm for solving Eq. (7.9) is constituted by the classic BP, which finds the element-wise optimum value rather than the global optimum. More explicitly, for $P = (P_0, P_1, \ldots, P_t, \ldots, P_{2n-1})$, BP finds P_t such that:

$$\tilde{P}_t = \arg\max_{P_t \in \mathbb{F}_2} \mathrm{P}(P_t|S), \tag{7.10}$$

where $\mathrm{P}(P_t|S)$ is the marginalized probability of the tth bit. The BP operates by exchanging messages over the Tanner graph of H having check nodes c_i for $i \in \{0, m-1\}$ and variable nodes v_t for $t \in \{0, 2n-1\}$. The messages sent by the ith check node c_i to the tth variable node are denoted by $m^{P_t}_{c_i\rightarrow v_t}$, while the messages directed from the tth variable node to the ith check node are given by $m^{P_t}_{v_t\rightarrow c_i}$, where P_t is the error imposed on the tth variable node. The overall syndrome-based message exchange procedure is summarized in Algorithm 1, which proceeds as follows [187]:

- **Initialization:** The algorithm begins by initializing the messages $m^{P_t}_{v_t\rightarrow c_i}$ according to the channel model $\mathrm{P}_{\mathrm{ch}}(P_t)$. For a BSC having a crossover probability of $2p/3$, we have:

$$\begin{aligned} m^{0}_{v_t\rightarrow c_i} &= 1 - 2p/3, \\ m^{1}_{v_t\rightarrow c_i} &= 2p/3. \end{aligned} \tag{7.11}$$

- **Horizontal message exchange:** Let $V(c_i)$ be the set of variable nodes connected to the check node c_i, i.e. $V(c_i) \equiv \{v_t : H_{it} = 1\}$, and $V(c_i)\backslash v_t$ be the set $V(c_i)$ excluding the variable node v_t. As depicted in Figure 7.7(a), in this step the algorithm runs through the rows of H (checks) and computes the message $m^{P_t}_{c_i \to v_t}$ for each $v_t \in V(c_i)$ and $P_t \in \mathbb{F}_2$. The message $m^{a}_{c_i \to v_t}$ represents the probability that the syndrome value observed for the check c_i is S_i given that the tth variable node has the error $(P_t = a)$, where $a \in \{0, 1\}$. This can be mathematically formulated as:

$$m^{a}_{c_i \to v_t} = K \sum_{P:P_t=a} \mathrm{P}(S_i|P) \prod_{v_{t'} \in V(c_i) \setminus v_t} m^{P_{t'}}_{v_{t'} \to c_i}, \tag{7.12}$$

where K is the normalization constant invoked for ensuring $\sum_{a \in \{0,1\}} m^{a}_{c_i \to v_t} = 1$, while $\mathrm{P}(S_i|P)$ is a binary function, which is equal to 1 only when the check c_i is satisfied, i.e. when the value of the check node c_i computed using the error vector P matches the measured syndrome value S_i, otherwise it is 0. Furthermore, according to Eq. (7.12), the messages $m^{a}_{c_i \to v_t}$ destined for the tth variable node do not take into account the messages flowing in the opposite direction along the same edge, i.e. $m^{a}_{v_t \to c_i}$. Consequently, $m^{a}_{c_i \to v_t}$ only contains the new information gleaned from the messages sent by the other variable nodes and it is therefore termed as being '*extrinsic*'. This ensures that the successive iterations of this iterative algorithm are independent.

- **Vertical message exchange:** Let $C(v_t)$ be the set of check nodes connected to the variable node v_t, i.e. $C(v_t) \equiv \{c_i : H_{it} = 1\}$, and $C(v_t) \setminus c_i$ be the set $C(v_t)$ excluding the check node c_i. As shown in Figure 7.7(b), for each column of H (hence called 'vertical'), the BP computes the message $m^{P_t}_{v_t \to c_i}$ for all $c_i \in V(v_t)$ and $P_t \in \mathbb{F}_2$. More explicitly, the messages $m^{a}_{v_t \to c_i}$ are computed by evaluating the product of the channel information $\mathrm{P}_{\mathrm{ch}}(P_t = a)$ and the messages $m^{a}_{c_{i'} \to v_t}$ flowing into the variable node v_t along all the edges connected to it, but excluding $m^{a}_{c_i \to v_t}$, which is received along the same edge. Hence, the extrinsic message is computed as:

$$m^{a}_{v_t \to c_i} = K \, \mathrm{P}_{\mathrm{ch}}(P_t = a) \prod_{c_{i'} \in C(v_t) \setminus c_i} m^{a}_{c_{i'} \to v_t}, \tag{7.13}$$

where K is the normalization constant, which ensures that $\sum_{a \in \{0,1\}} m^{a}_{v_t \to c_i} = 1$.

- **Element-wise marginal probability:** Finally, the element-wise marginal probability $\mathrm{P}(P_t|S)$ for $P_t \in \mathbb{F}_2$ is calculated as follows:

$$\mathrm{P}(P_t = a|S) = K \, \mathrm{P}_{\mathrm{ch}}(P_t = a) \prod_{c_i \in C(v_t)} m^{a}_{c_i \to v_t}, \tag{7.14}$$

which takes into account all the messages flowing into the variable node v_t.

- **Hard decision & syndrome check:** As previously portrayed in Eq. (7.10), a hard decision is made by finding the most likely error $\tilde{P}_t$, which maximizes the marginal probability computed in Eq. (7.14). Based on the estimated error vector $\tilde{P}$, the syndrome $\tilde{S} = H(\tilde{P}_x : \tilde{P}_z)^T$ is computed. If the syndrome $\tilde{S}$ of the estimated error $\tilde{P}$ is the same as the observed syndrome S, the process halts, indicating that the correct solution is found. Otherwise, the algorithm repeats itself from the horizontal message exchange step onwards. This iterative procedure continues, until either $\tilde{S} = S$ or the maximum number of iterations $\mathtt{I}_{\max}$ is reached.

Algorithm 1 Syndrome-based BP

```
Set $P_{\text{ch}}(0) \leftarrow (1 - 2p/3)$ and $P_{\text{ch}}(1) \leftarrow 2p/3$.
Initialize $m^a_{v_t \to c_i} \leftarrow P_{\text{ch}}(a)$, $\forall\ v_t$, $c_i \in C(v_t)$ and $a \in \{0, 1\}$.
for iter $\leftarrow 1$ to $\mathtt{I}_{\max}$ do
  for all $i \in \{0, (m-1)\}$, $v_t \in V(c_i)$ and $a \in \{0, 1\}$ do
    $m^a_{c_i \to v_t} \leftarrow k \sum_{P:P_t=a} \mathrm{P}(S_i|P) \prod_{v_{t'} \in V(c_i) \setminus v_t} m^{P_{t'}}_{v_{t'} \to c_i}$.
  end for
  for $t \leftarrow 0$ to $(2n-1)$ do
    for all $c_i \in C(v_t)$ and $a \in \{0, 1\}$ do
      $m^a_{v_t \to c_i} \leftarrow k\ \mathrm{P}_{\text{ch}}(P_t = a) \prod_{c_{i'} \in C(v_t) \setminus c_i} m^a_{c_{i'} \to v_t}$.
    end for
    for all $a \in \{0, 1\}$ do
      $\mathrm{P}(P_t = a|S) \leftarrow k\ \mathrm{P}_{\text{ch}}(P_t = a) \prod_{c_i \in C(v_t)} m^a_{c_i \to v_t}$.
    end for
    $\tilde{P}_t \leftarrow \arg\max_{P_t \in \mathbb{F}_2} \mathrm{P}(P_t|S)$.
  end for
  $\tilde{S} \leftarrow H(\tilde{P}_x : \tilde{P}_z)^T$.
  if $(\tilde{S} = S)$ then
    return $\tilde{P}$.
  end if
end for
```

7.3.2 Non-Binary Decoding

Based on the Pauli-to-GF(4) formalism of Eq. (4.39), QLDPC codes can be decoded by invoking the non-binary BP, which takes into account the correlation between the phase-flips and bit-flips. The syndrome-based non-binary BP is similar to the binary BP of Algorithm 1, with the following two major modifications:

- Non-binary BP exploits the depolarizing channel model, which does not ignore the correlation between the bit and phase errors. The equivalent 4-ary channel model has the following probability distribution:

$$\mathbf{P}_{\text{ch}}\left(\hat{P}_t = \hat{a}\right) = \begin{cases} 1-p, & \text{if } \hat{a} = 0 \\ p/3, & \text{if } \hat{a} \in \{1, \omega, \overline{\omega}\}, \end{cases} \tag{7.15}$$

  where we have $\hat{P} = \left(\hat{P}_0, \hat{P}_1, \ldots, \hat{P}_t, \ldots, \hat{P}_{n-1}\right)$ and $\hat{P}_t$ denotes the error inflicted on the tth qubit.

- The syndrome S_i, which was computed as $H_i(P_x : P_z)^T$ in the binary scenario, is now given by the trace inner product of $\hat{H}_i$ and $\hat{P}$ (see Eq. (4.41)):

$$S_i = \mathrm{Tr}(\hat{H}_i \cdot \hat{P}), \tag{7.16}$$

where $\hat{H}_i$ is the ith row of H in GF(4) and $i \in \{0, m-1\}$.

As compared to the binary BP, non-binary decoding imposes an increased complexity, specifically on the horizontal message exchange step. More explicitly, since the summation in Eq. (7.12) runs for all possible error sequences $\{\hat{P} : \hat{P}_t = \hat{a}\}$, which yield the syndrome S_i for the ith check node, the complexity increases both with the row weight as well as with the dimensionality of the Galois field. For classical non-binary LDPC codes, this increased complexity is alleviated by invoking the Fast Fourier Transform (FFT) based decoding of [189], which can be conveniently adapted to the syndrome-based decoding of QLDPC codes.

Based on the notion of the trace inner product of Eq. (4.41), Eq. (7.16) can be expanded as:

$$S_i = \mathrm{Tr}(\hat{S}_i) = \mathrm{Tr}\left(\sum_{t \in V(c_i)} \hat{H}_{it} \times \overline{\hat{P}}_t\right), \tag{7.17}$$

where we have $\hat{S}_i \in \{0, 1, \omega, \overline{\omega}\}$, which can also be expressed as:

$$\hat{S}_i = \hat{H}_{it} \times \overline{\hat{P}}_t + \sum_{t' \in V(c_i) \setminus v_t} \hat{H}_{it'} \times \overline{\hat{P}}_{t'}. \tag{7.18}$$

Unlike in the binary scenario, where we have $H_{it} \in \{0, 1\}$, here we have $\hat{H}_{it} \in \{1, \omega, \overline{\omega}\}$ in Eq. (7.18). Therefore, given the messages $m^{\hat{a}}_{c_i \to v_t}$ and $m^{\hat{a}}_{v_t \to c_i}$ exchanged between the check node c_i and the variable node v_t for $\hat{P} = \hat{a}$, we denote the equivalent messages for $(\hat{H}_{it} \times \overline{\hat{P}}_t)$ as $\check{m}^{\hat{a}_s}_{c_i \to v_t}$ and $\check{m}^{\hat{a}_s}_{v_t \to c_i}$, respectively, where we have $(\hat{H}_{it} \times \overline{\hat{a}}) = \hat{a}_s$. Based on this notation, we may infer from Eq. (7.17) and Eq. (7.18) that the Probability Density Function (PDF) of the horizontal message $\check{m}^{\hat{a}_s}_{c_i \to v_t}$ can be obtained by convolving the PDFs of the messages $\check{m}^{\hat{a}_s + \hat{S}_i}_{v_{t'} \to c_i}$ for $v_{t'} \in V(c_i) \setminus v_t$. We may further notice in Eq. (7.17) that for a given S_i, $\hat{S}_i$ can have two possible values. More explicitly, for GF(4), we have $\mathrm{Tr}(0) = \mathrm{Tr}(1) = 0$, while $\mathrm{Tr}(\omega) = \mathrm{Tr}(\overline{\omega}) = 1$. Consequently, for $S_i = \mathrm{Tr}(\hat{S}_i = 0) = \mathrm{Tr}(\hat{S}_i = 1) = 0$, we have:

$$\begin{aligned}
&PDF\{\check{m}^{0}_{c_i \to v_t}\} = PDF\{\check{m}^{1}_{c_i \to v_t}\} \\
&= \frac{1}{2}\left(\bigotimes_{v_{t'}} PDF\{\check{m}^{0}_{v_{t'} \to c_i}\} + \bigotimes_{v_{t'}} PDF\{\check{m}^{1}_{v_{t'} \to c_i}\}\right), \\
&PDF\{\check{m}^{\omega}_{c_i \to v_t}\} = PDF\{\check{m}^{\overline{\omega}}_{c_i \to v_t}\} \\
&= \frac{1}{2}\left(\bigotimes_{v_{t'}} PDF\{\check{m}^{\omega}_{v_{t'} \to c_i}\} + \bigotimes_{v_{t'}} PDF\{\check{m}^{\overline{\omega}}_{v_{t'} \to c_i}\}\right),
\end{aligned} \tag{7.19}$$

where $\bigotimes$ represents the convolution process and $v_{t'} \in V(c_i) \setminus v_t$. Similarly, for $S_i = \text{Tr}(\hat{S}_i = \omega) = \text{Tr}(\hat{S}_i = \overline{\omega}) = 1$, we have:

$$\begin{aligned}
&PDF\{\check{m}^0_{c_i \to v_t}\} = PDF\{\check{m}^1_{c_i \to v_t}\} \\
&= \frac{1}{2}\left(\bigotimes_{v_{t'}} PDF\{\check{m}^{\omega}_{v_{t'} \to c_i}\} + \bigotimes_{v_{t'}} PDF\{\check{m}^{\overline{\omega}}_{v_{t'} \to c_i}\}\right), \\
&PDF\{\check{m}^{\omega}_{c_i \to v_t}\} = PDF\{\check{m}^{\overline{\omega}}_{c_i \to v_t}\} \\
&= \frac{1}{2}\left(\bigotimes_{v_{t'}} PDF\{\check{m}^0_{v_{t'} \to c_i}\} + \bigotimes_{v_{t'}} PDF\{\check{m}^1_{v_{t'} \to c_i}\}\right). \qquad (7.20)
\end{aligned}$$

The complex convolution operation required in Eq. (7.19) and (7.20) can be efficiently implemented by multiplying the corresponding PDFs in the frequency domain with the aid of the FFT-based algorithm of [189].

7.3.3 Decoding Issues & Heuristic Methods for Improvement

Belief propagation invoked for decoding LDPC codes gives the exact solution only when the underlying Tanner graph is a tree. Nonetheless, it yields reasonably good approximations even in the presence of cycles, provided that the girth of the associated LDPC matrix is sufficiently large, at least 6. This has been proven by the capacity approaching classical LDPC codes, for example in [164, 165]. Unfortunately, short cycles of length 4 are unavoidable in the construction of QLDPC codes, which in turn impair the iterative decoding procedure.

The unavoidable cycles of length 4 found in QLDPC codes are the result of the commutativity property of the stabilizers. More explicitly, the constituent stabilizer generators of a stabilizer code must commute, i.e. they should have even number of places with different non-Identity Pauli operators. In other words, if an anti-commuting pair of Pauli operators acts on the tth variable node in a pair of stabilizer generators, then there should be another anti-commuting pair of Pauli operators acting on the t'th variable node in the same pair of generators for the sake of ensuring that the generators commute with each other. For example, the generators:

$$\begin{aligned}
g_0 &= \mathbf{XIYZ}, \\
g_1 &= \mathbf{ZYXI}, \qquad (7.21)
\end{aligned}$$

commute[7] because there are two pairs of anti-commuting Pauli operators acting on the first and third qubits, respectively. This in turn implies that the corresponding rows in the resultant PCM have even number of overlaps, which give rise to short cycles in the Tanner graph, as illustrated in Figure 7.8. Since here the key point is to have "different non-Identity operators", a possible option could be to

[7] This is just a random example to illustrate the concept of commutativity and the resulting short cycles. The generators g_0 and g_1 of this example may not constitute a good stabilizer code.

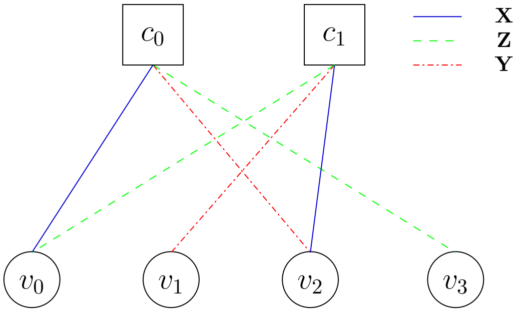


Figure 7.8: Tanner graph of a commuting pair of stabilizer generators, where c_0 and c_1 are the check nodes for the generators $g_0 = \mathbf{XIYZ}$ and $g_1 = \mathbf{ZYXI}$, respectively. The edges connected to the variable nodes v_0 and v_2 constitute a cycle of length 4.

assign only a single type of non-Identity operator to each variable node of the Tanner graph. If we only assign Pauli-**X** to the variable node v_t so that it does not anti-commute in any pair of generators, then we will be unable to detect both Pauli-**X** as well as Pauli-**Y** errors acting on v_t. This would yield an undesirable code, which has a minimum distance of one. We may conclude that:

1. each column of a QLDPC matrix must have at least two different non-Identity Pauli operators, and
2. every pair of rows must an have even number of places with different non-Identity Pauli operators.

Consequently, all CSS as well as non-CSS QLDPC constructions have a Tanner graph of girth-4. It is interesting to observe here that these short cycles may be avoided in the corresponding binary formalism. Let us consider the example given in Eq. (7.21), which can be expressed in the binary form as follows:

$$\begin{aligned} g_0 &\rightarrow \left(\begin{array}{cccc|cccc} 1 & 0 & 1 & 0 & 0 & 0 & 1 & 1 \end{array} \right), \\ g_1 &\rightarrow \left(\begin{array}{cccc|cccc} 0 & 1 & 1 & 0 & 1 & 1 & 0 & 0 \end{array} \right). \end{aligned} \tag{7.22}$$

Since these binary generators only have a single overlapping 1, the length 4 cycle no longer exits. However, let us recall from Section 7.3.2 that binary decoding ignores the correlation between the **X** and **Z** errors, which degrades the performance. Hence, a compromise must be struck between these two conflicting aspects.

The issue of short cycles is more pronounced in both the dual-containing QLDPC codes as well as in the EA CSS QLDPC codes having $H'_x = H'_z$. We may call them homogeneous CSS codes, since identical PCMs are used for correcting bit-flips and phase-flips. Let the resultant $m \times n$ PCM in GF(4) be $\hat{H}$ as follows:

$$\hat{H} = \begin{pmatrix} \omega H'_z \\ H'_z \end{pmatrix}. \tag{7.23}$$

Code Type	Unavoidable Short Cycles	
	Binary Formalism	GF(4) Formalism
Dual-containing CSS (Homogeneous)	✓	✔
Non-dual-containing CSS	✗	✓
Non-CSS	✗	✓
Homogeneous CSS EA	✗	✔
All other EA	✗	✗

Table 7.1: Unavoidable short cycles in various code structures *(✓ = present, ✗ = absent, ✔ = numerous cycles present).*

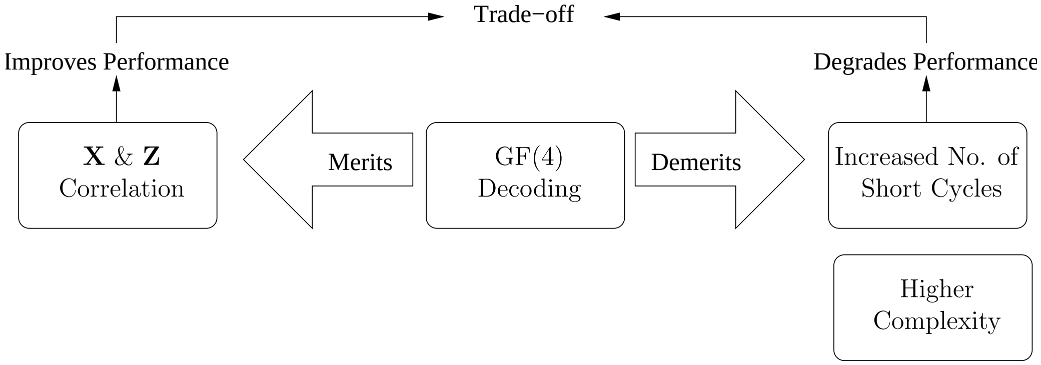


Figure 7.9: Merits and demerits of GF(4) decoding as compared to binary decoding.

Consequently, the ith and $(i+m/2)$th rows completely overlap, resulting in numerous cycles of length 4. Furthermore, the dual-containing code construction also has the additional short cycles within the matrix H'_z, as discussed in Section 7.2.1, which exist even in the binary formalism. Table 7.1 summarizes the presence of unavoidable short cycles in various code structures, while Figure 7.9 captures the merits and demerits of GF(4) decoding as compared to its binary counterpart.

Degeneracy is another unique aspect, which distinguishes a quantum code from a classical one. Let us recall from Section 4.3.3 that errors, which differ only by the stabilizer group, have the same impact on the transmitted codewords and can therefore be corrected by the same recovery operation. This in turn improves the performance of quantum codes. Unfortunately, the iterative decoding invoked for QLDPC codes does not take into account this degeneracy. More explicitly, rather than finding the most likely error, as in Eq. (7.9), the decoding algorithm should find the most likely error coset by summing the probabilities of all degenerate errors [85, 190]. Furthermore, QLDPC codes are highly degenerate as compared to the other families of quantum codes. This is because the generators of a QLDPC code are sparse in nature. Consequently, it has many low-weight degenerate errors, which dominate the probability of the error coset. It is therefore more likely that the most probable error $\tilde{P}$ of Eq. (7.9) may not coincide with the most probable error coset for QLDPC codes. However, rather than exploiting the benefits of high degeneracy associated with sparse codes, the marginalized iterative decoding invoked for QLDPC codes is impaired by degeneracy[85, 190]. This is because degenerate errors of equal weight have the same marginalized probability distribution, which can be attributed to the symmetry of the probability distribution of the channel depicted in Eq. (7.15).

Let us review the case study given in [85]. Consider a 2-qubit stabilizer code having the generators **XX** and **ZZ**. Assume furthermore that **IX** is the channel error encountered during transmission over a depolarizing channel, whose PDF is given in Eq. (7.15). The resultant syndrome is $S = (0\ 1)$ and the corresponding set of degenerate errors is $\{\mathbf{XI}, \mathbf{IX}, \mathbf{YZ}, \mathbf{ZY}\}$. Consequently, the marginalized conditional probability of the error on each of the two qubits is given by:

$$\mathrm{P}\left(\hat{P}_t = \hat{a}|S\right) = \begin{cases} 1-p, & \text{if } \hat{a} = 0 \\ p/3, & \text{if } \hat{a} \in \{1, \omega, \overline{\omega}\}, \end{cases} \tag{7.24}$$

where $t = \{0, 1\}$. Hence, the marginalized probability is identical for both the qubits. This symmetry forces the decoder to detect the same error on both the qubits. However, none of the associated errors, i.e. $\{\mathbf{XI}, \mathbf{IX}, \mathbf{YZ}, \mathbf{ZY}\}$, exhibit this symmetry, hence leading to the 'symmetric degeneracy error' concept of [85]. Moreover, since the channel profile of Eq. (7.15) is biased towards the Identity operator, the probability of 'no-error' dominates at low noise levels.

Poulin and Chung investigated various heuristic methods in [85] to break the symmetry exhibited by the marginalized probabilities of Eq. (7.24). Among the investigated methods, "random perturbation" provides the best performance. It aims for breaking the degenerate symmetry by randomly perturbing the channel PDF of Eq. (7.15) for the qubits involved in the frustrated checks[8], thus putting an end to the decoding impasse. Random perturbation begins with the standard non-binary BP, which gives the estimated channel error $\hat{\tilde{P}}$. If the syndrome computed for $\hat{\tilde{P}}$ is not the same as the observed channel syndrome S, the channel probabilities of all variable nodes v_t connected to a randomly chosen frustrated check c_i are perturbed (up to a normalization) as follows:

$$\begin{aligned} &\mathbf{P}_{\mathrm{ch}}(\hat{P}_t = 0) \rightarrow \mathbf{P}_{\mathrm{ch}}(\hat{P}_t = 0), \\ &\mathbf{P}_{\mathrm{ch}}(\hat{P}_t = 1) \rightarrow (1+\delta_1)\mathbf{P}_{\mathrm{ch}}(\hat{P}_t = 1), \\ &\mathbf{P}_{\mathrm{ch}}(\hat{P}_t = \omega) \rightarrow (1+\delta_\omega)\mathbf{P}_{\mathrm{ch}}(\hat{P}_t = \omega), \\ &\mathbf{P}_{\mathrm{ch}}(\hat{P}_t = \overline{\omega}) \rightarrow (1+\delta_{\overline{\omega}})\mathbf{P}_{\mathrm{ch}}(\hat{P}_t = \overline{\omega}), \end{aligned} \tag{7.25}$$

where δ_1, δ_ω and $\delta_{\overline{\omega}}$ are random variables in the range $[0, \delta]$. Non-binary BP is re-run with these modified channel probabilities for T_{pert} iterations and $\hat{\tilde{P}}$ is estimated again. If all the check nodes are satisfied now, the process terminates. Otherwise, the channel probabilities perturbed in Eq. (7.25) are restored and the process is repeated with another randomly chosen frustrated check.

[8] Check nodes for which the computed syndrome does not match the observed syndrome are known as frustrated checks [85].

Another heuristic method of alleviating the symmetric degeneracy problem was conceived in [86], which exploits an enhanced feedback procedure. More specifically, Wang *et al.* [86] proposed an "enhanced feedback" strategy for perturbing the channel probabilities similar to the random perturbation, but this perturbation is based both on the stabilizer generators involved in the frustrated checks as well as on the channel model. Similar to the random perturbation method, the enhanced feedback algorithm randomly selects a frustrated check c_i. It also selects a variable node v_t connected to c_i. Let $\tilde{S}_i$ be the value of the ith check node for the estimated error $\tilde{\hat{P}}$, while S_i be the ith observed channel syndrome. The channel probability for v_t is then perturbed as follows:

- If $\tilde{S}_i = 0$ and $S_i = 1$, then:

$$\mathbf{P}_{\text{ch}}\left(\hat{P}_t = \hat{a}\right) = \begin{cases} p/2, & \text{if } \hat{a} = 0 \text{ or } \hat{H}_{it}, \\ (1-p)/2, & \text{otherwise.} \end{cases} \tag{7.26}$$

- If $\tilde{S}_i = 1$ and $S_i = 0$, then:

$$\mathbf{P}_{\text{ch}}\left(\hat{P}_t = \hat{a}\right) = \begin{cases} (1-p)/2, & \text{if } \hat{a} = 0 \text{ or } \hat{H}_{it}, \\ p/2, & \text{otherwise.} \end{cases} \tag{7.27}$$

The perturbed values are fed to the standard non-binary BP decoder, which provides a new estimate of the channel error. The perturbation process is repeated, until all the checks are satisfied or the maximum number of feedbacks n_a is reached. Since these perturbations are more reliable than random perturbations, this method outperforms the random perturbation based heuristic method of [47].

7.4 High-Rate QLDPC Codes from Row-Circulant Classical LDPCs

Classical row-circulant QC-LDPC matrices are known to generate efficient short and moderate length high-rate QC-LDPC codes, which outperform the corresponding randomly constructed codes. More specifically, short and moderate length random (unstructured) LDPC matrices with a high coding rate have numerous cycles of length 4, which may be avoided in the structured configurations. The BIBD and Cyclic Difference Family (CDF) based code structures of [191, 192, 171, 172, 193] are particularly significant in this respect, which have at least a girth of 6.

Let us have a look at the BIBD constructions proposed by Bose [194], which constitute the row-circulant classical QC-LDPCs. Recall that a BIBD, having the parameters (v, b, r, k, λ), divides all the v elements of a set V into b blocks of size k such that, each pair of elements occurs in exactly λ of the blocks, every element occurs in exactly r blocks, and the number of elements in each block k is small as compared to the size v of set V. These parameters are summarized in Table 7.2. Based on this notation, Bose proposed the following BIBD [194] constructions, which are suitable for the row-circulant QC-LDPCs [172, 195].

Variable	Description
V	Finite set.
v	Number of elements of V.
b	Number of blocks.
r	Number of blocks containing a given element.
k	Number of elements of a block.
λ	Number of blocks containing a given pair of elements.

Table 7.2: BIBD parameters.

1. **Type-I Bose BIBDs:** Given that $\mathtt{t}$ is a positive integer such that $(12\mathtt{t}+1)$ is a power of a prime, then there exists a prime Galois field GF$(12\mathtt{t}+1)$ having elements from 0 to $12\mathtt{t}$. The elements of GF$(12\mathtt{t}+1)$ constitute the finite set V of the BIBD. Furthermore, let α be the primitive element of GF$(12\mathtt{t}+1)$, which satisfies the following condition:

$$\alpha^{4\mathtt{t}} - 1 = \alpha^{\mathtt{c}}, \tag{7.28}$$

where $\mathtt{c}$ is an integer in the range $0 < \mathtt{c} < 12\mathtt{t}+1$. The legitimate values of $\mathtt{t}$ along with the corresponding primitive elements are listed in Table 7.3 [195, p. 526]. For these values of $\mathtt{t}$ and α, Bose [194] proposed that there exists a BIBD having the parameters $v = (12\mathtt{t}+1)$, $b = \mathtt{t}(12\mathtt{t}+1)$, $r = 4\mathtt{t}$, $k = 4$ and $\lambda = 1$, whose $\mathtt{t}$ base blocks are given by:

$$B_i = \{0, \alpha^{2i}, \alpha^{2i+4\mathtt{t}}, \alpha^{2i+8\mathtt{t}}\}, \tag{7.29}$$

for $0 \leq i < \mathtt{t}$. We can further create $(12\mathtt{t}+1)$ blocks for block B_i by adding each element of the Galois field to each element of the base block, hence resulting in a total of $\mathtt{t}(12\mathtt{t}+1)$ blocks. The incidence matrix of this BIBD is a $(12\mathtt{t}+1) \times \mathtt{t}(12\mathtt{t}+1)$ matrix, which is formed by $\mathtt{t}$ submatrices as follows:

$$H_{\text{BIBD}} = \left(H_0, H_1, \ldots, H_{\mathtt{t}-1}\right), \tag{7.30}$$

where the ith submatrix H_i is a $(12\mathtt{t}+1) \times (12\mathtt{t}+1)$ circulant matrix. More specifically, H_i is the incidence matrix corresponding to the ith base block constructed by adding each element of the GF$(12\mathtt{t}+1)$ to the elements of the ith base block. Furthermore, the row weight and column weight of each submatrix is 4. Therefore, the matrix H_{BIBD} has a row weight of $4\mathtt{t}$ and a column weight of 4. Since the incidence matrix of Eq. (7.30) has the required properties of a QC-LDPC matrix, a subarray of H_{BIBD} can be used for constructing a classical QC-LDPC code. For $0 < m < \mathtt{t}$, the PCM of the resulting code is given by [172]:

$$H = \left(H_0, H_1, \ldots, H_{m-1}\right), \tag{7.31}$$

$\mathtt{t}$	GF(q)	$(\alpha, \mathtt{c})$
1	GF(13)	$(2, 1)$
6	GF(73)	$(5, 33)$
8	GF(97)	$(5, 27)$
9	GF(109)	$(6, 71)$
15	GF(181)	$(2, 13)$
19	GF(229)	$(6, 199)$
20	GF(241)	$(7, 191)$
23	GF(277)	$(5, 209)$
28	GF(337)	$(10, 129)$
34	GF(409)	$(21, 9)$
35	GF(421)	$(2, 167)$
38	GF(457)	$(13, 387)$
45	GF(541)	$(2, 7)$
59	GF(709)	$(2, 381)$
61	GF(733)	$(6, 145)$

Table 7.3: Legitimate values of $\mathtt{t}$ for GF$(12\mathtt{t}+1)$ along with the corresponding primitive elements satisfying Eq. (7.28) [195, p. 526].

which is a $(12\mathtt{t}+1) \times m(12\mathtt{t}+1)$ QC matrix having a row weight of $4m$ and a column weight of 4. The minimum distance of the resulting LDPC code is at least 5 and the coding rate is approximately $(m-1)/m$ [172].

2. **Type-II Bose BIBDs:** Let $\mathtt{t}$ be a positive integer such that $(20\mathtt{t}+1)$ is a power of a prime, then there exists a prime Galois field GF$(20\mathtt{t}+1)$ having elements from 0 to $20\mathtt{t}$. Analogous to the type-I design, the elements of GF$(20\mathtt{t}+1)$ constitute the finite set V of the BIBD. However, in contrast to Eq. (7.28), the primitive element α of GF$(20\mathtt{t}+1)$ has to satisfy the condition:

$$\alpha^{4\mathtt{t}} + 1 = \alpha^{\mathtt{c}}, \tag{7.32}$$

where $\mathtt{c}$ is an integer in the range $0 < \mathtt{c} < 20\mathtt{t}+1$. The legitimate values of $\mathtt{t}$ along with the corresponding primitive elements are listed in Table 7.4 [195, p. 532]. For these values of $\mathtt{t}$ and α, Bose [194] proposed that we can construct a BIBD having the parameters $v = (20\mathtt{t}+1)$, $b = \mathtt{t}(20\mathtt{t}+1)$, $r = 5\mathtt{t}$, $k = 5$ and $\lambda = 1$, whose $\mathtt{t}$ base blocks are given by:

$$B_i = \{\alpha^{2i}, \alpha^{2i+4\mathtt{t}}, \alpha^{2i+8\mathtt{t}}, \alpha^{2i+12\mathtt{t}}, \alpha^{2i+16\mathtt{t}}\}, \tag{7.33}$$

for $0 \leq i < \mathtt{t}$. Similar to the type-I design, $(12\mathtt{t}+1)$ blocks can be constructed for each base block B_i by adding each element of the Galois field to each element of the base block. Hence, there is a total of $\mathtt{t}(20\mathtt{t}+1)$ blocks. The incidence matrix of the resulting BIBD is a $(20\mathtt{t}+1) \times \mathtt{t}(20\mathtt{t}+1)$

t	GF(q)	$(\alpha, \mathtt{c})$
2	GF(41)	$(6, 3)$
3	GF(61)	$(2, 23)$
12	GF(241)	$(7, 197)$
14	GF(281)	$(3, 173)$
21	GF(421)	$(2, 227)$
30	GF(601)	$(7, 79)$
32	GF(641)	$(3, 631)$
33	GF(661)	$(2, 657)$
35	GF(701)	$(2, 533)$
41	GF(821)	$(2, 713)$

Table 7.4: Legitimate values of $\mathtt{t}$ for $(20\mathtt{t} + 1)$ along with the corresponding primitive elements satisfying Eq. (7.32) [195, p. 532].

matrix, which is formed by $\mathtt{t}$ submatrices as previously shown in Eq. (7.30). For the type-II design, the row weight and column weight of each submatrix H_i is 5. Therefore, the matrix H_{BIBD} has a row weight of $5\mathtt{t}$ and a column weight of 5. Again, according to Eq. (7.31), we can construct a QC-LDPC of size $(20\mathtt{t} + 1) \times m(20\mathtt{t} + 1)$, which has a row weight of $5m$ and a column weight of 5 [172].

In [74], Djordjevic proposed that BIBDs having an odd row weight and $\lambda = 1$ can be transformed into a dual-containing QLDPC code by appending an all-ones column to the original PCM, while in [182] Djordjevic conceived the EA BIBD-based QLDPC codes, which required a single e-bit.

In contrast to these developments, we conceive a radically new class of codes for constructing unassisted non-dual-containing CSS QLDPC codes from arbitrary row-circulant classical QC-LDPC matrices. This construction brings with it the following obvious benefits:

- Pre-shared ebits are not required.
- Since the constructed codes are non-dual-containing, they do not suffer from the excessive short cycles, which are a characteristic of the dual-containing designs as well as the homogeneous CSS EA codes (Table 7.1).

Particularly, we apply our proposed construction to the family of BIBD-based classical codes for evaluating the resulting performance.

Let us consider the row-circulant QC-LDPC matrix of Eq. (7.31), assuming that it consists of an even number of square circulant submatrices. Inspired by the QC-QLDPC codes of [77, 177], we propose that if we formulate H'_z and H'_x as follows:

$$\begin{aligned} H'_z &= H, \\ H'_x &= \left(H^T_{\frac{m}{2}}, H^T_{\frac{m}{2}+1}, \ldots, H^T_{m-1}, H^T_0, H^T_1, \ldots, H^T_{\frac{m}{2}-1}\right), \end{aligned} \tag{7.34}$$

where when m is even, the resulting CSS code satisfies the symplectic criterion $H'_z H'^T_x = 0$. This may be readily proved as shown below:

$$\begin{aligned} H'_z H'^T_x &= \left(H_0, H_1, \ldots, H_{\frac{m}{2}-1}, H_{\frac{m}{2}}, H_{\frac{m}{2}+1}, \ldots, H_{m-1}\right) \begin{pmatrix} H_{\frac{m}{2}} \\ H_{\frac{m}{2}+1} \\ \vdots \\ H_{m-1} \\ H_0 \\ H_1 \\ \vdots \\ H_{\frac{m}{2}-1} \end{pmatrix} \\ &= H_0 H_{\frac{m}{2}} + H_1 H_{\frac{m}{2}+1} + \cdots + H_{\frac{m}{2}-1} H_{m-1} \\ &+ H_{\frac{m}{2}} H_0 + H_{\frac{m}{2}+1} H_1 + \cdots + H_{m-1} H_{\frac{m}{2}-1} \ . \end{aligned} \tag{7.35}$$

Since the multiplication of circulant matrices is commutative, the two parts of Eq. (7.35), i.e. $(H_0 H_{\frac{m}{2}} + H_1 H_{\frac{m}{2}+1} + \cdots + H_{\frac{m}{2}} H_m)$ and $(H_{\frac{m}{2}} H_0 + H_{\frac{m}{2}+1} H_1 + \cdots + H_{m-1} H_{\frac{m}{2}-1})$ are equal. Hence, the modulo 2 addition of Eq. (7.35) yields 0; thus, satisfying the symplectic criterion. Furthermore, the resulting quantum coding rate is $(m-2)/2$. We may also notice here that our proposed code structure may be viewed as a special case of Eq. (7.7) for $m = 2$, , given further that T_1 and T_2 of Eq. (7.7) are binary circulant matrices, i.e. $t_i \in \{0, 1\}$ in Eq. (7.6).

Let us now have a look at the girth of the resulting QC-QLDPC code. The constituent $l \times l$ circulant submatrix H_i of Eq. (7.31), which has a row weight and a column weight of γ, is completely characterized by the polynomial $h_i(x) = x^{d_{i,0}} + x^{d_{i,1}} + \cdots + x^{d_{i,\gamma-1}}$, where $d_{i,j}$ denotes the column index of the jth non-zero entry in the first row of H_i. For example, if the first row of H_i has a 1 at index 0, 5 and 8, then the polynomial is given by $1 + x^5 + x^8$. The PCM H has a girth of at least 6 if every difference $(d_{i,j_1} - d_{i,j_2})$ modulo l, for $0 \leq i \leq (m-1)$ and $0 \leq j_1, j_2 \leq (\gamma - 1)$ is a unique integer between 0 and $(l-1)$. Furthermore, the polynomial transpose is defined as $h_i(x)^T = x^{l-d_{i,1}} + x^{l-d_{i,2}} + \cdots + x^{l-d_{i,\gamma-1}}$, which would yield the same differences as $h_i(x)$. Hence, since in Eq. (7.34) we are taking the transpose of all the sub-matrices H_i and just permuting their location, the differences $(d_{i,j_1} - d_{i,j_2})$ for H'_z and H'_x are the same and both have the same girth.

7.5 Results and Discussions I

To evaluate the performance of our proposed design, we considered BIBD$(12\mathtt{t}+1, \mathtt{t}(12\mathtt{t}+1), 4\mathtt{t}, 4, 1)$. We further arbitrarily chose $\mathtt{t} = 15$ from Table 7.3, whose primitive root is $\alpha = 2$. Consequently, the

BIBD Parameters	
Galois field	$\mathrm{GF}(12\mathtt{t}+1)$
$\mathtt{t}$	15
α	2
B_i	$\{0, 2^{2i}, 2^{2i+4\mathtt{t}}, 2^{2i+8\mathtt{t}}\}$
m	14
LDPC Parameters	
Coded qubits	$n = 2534$
Information qubits	$k = 2172$
E-bits	$c = 0$
Row weight	56
Column Weight	4
QLDPC Decoder	
Standard decoding iterations	$\mathtt{I}_{\max} = 100$

Table 7.5: Simulation parameters.

resulting $\mathtt{t}$ base blocks are give by:

$$B_i = \{0, 2^{2i}, 2^{2i+4\mathtt{t}}, 2^{2i+8\mathtt{t}}\}, \tag{7.36}$$

for $0 \leq i < \mathtt{t}$. Since our design requires m to be even, we chose $m = 14$, which would yield a $[2534, 2172]$ QLDPC code having a coding rate of 0.857. Furthermore, the row weight of the resulting QLDPC code is 56, while its column weight is 4. These parameters are summarized in Table 7.5. We further compare our design with an equivalent bicyle QLDPC code, whose PCM is given by:

$$H'_x = H'_z = [H_0, H_1, \ldots, H_{\frac{m}{2}-1}, H_0^T, H_1^T, \ldots, H_{\frac{m}{2}-1}^T], \tag{7.37}$$

and a comparable EA-QLDPC, which requires a single e-bit and has:

$$H'_x = H'_z = [H_0, H_1, \ldots, H_{m-1}]. \tag{7.38}$$

Figure 7.10 compares the performance of our designed QLDPC code (labeled 'Proposed'), the bicycle code of Eq. (7.37) (labeled 'BiC') and the EA-QLDPC code of Eq. (7.38) (labeled 'EA') for both the binary as well as the non-binary decoding. We invoked a maximum of 100 iterations. Each decoding algorithm iterates until either a valid error is found or the maximum number of iterations

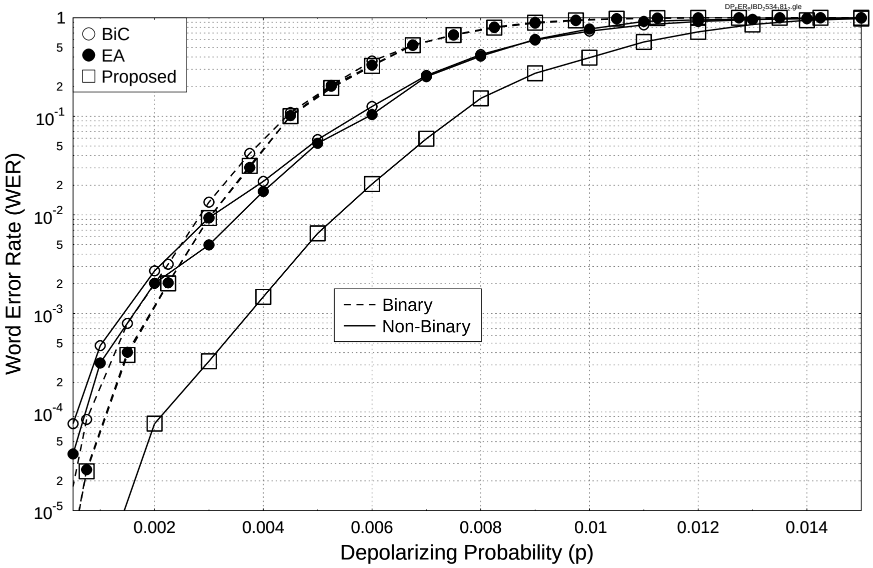


Figure 7.10: Comparison of the achievable WER performance of our proposed $[2534, 2172]$ QLDPC code (labeled 'Proposed')) with the bicycle code of Eq. (7.37) (labeled 'BiC') and the EA-QLDPC code of Eq. (7.38) (labeled 'EA'), using the simulation parameters of Table 7.5.

is reached. Furthermore, the WER metric here counts the detected as well as the undetected block errors. We may observe in Figure 7.10 that the performance of our designed QLDPC is exactly the same as that of the EA-QLDPC code for binary decoding, while that of the bicycle QLDPC code is slightly worse, which is due to the presence of length-4 cycles. By contrast, when non-binary decoding is invoked, then the performance of our proposed design improves significantly as compared to both the bicycle code as well as the EA-QLDPC. More specifically, the bicycle code achieves a WER of 10^{-4} at $p = 0.00055$, which increases to $p = 0.0007$ when EA-QLDPC is used. By contrast, our construction exhibits a WER of 10^{-4} at $p = 0.00215$, which is almost a three times increase as compared to EA-QLDPC. As mentioned in Section 7.4, unlike the bicycle codes and EA-QLDPC, which have numerous short cycles in the GF(4) formalism, our design has only a few short cycles in the GF(4) formalism. Consequently, it outperforms the comparable bicycle and EA counterparts. It is also pertinent to notice in Figure 7.10 that the performance of the non-binary decoder is not always better than that of the binary decoder. As previously discussed in Figure 7.9, a non-binary decoder tends to improve the performance because it takes into account the correlation between the bit and phase errors. However, the GF(4) formalism increases the number of short cycles, which degrades the performance. Due to these conflicting attributes, a non-binary decoder may not always outperform its binary counterpart, as observed Figure 7.10 in the low-noise regime.

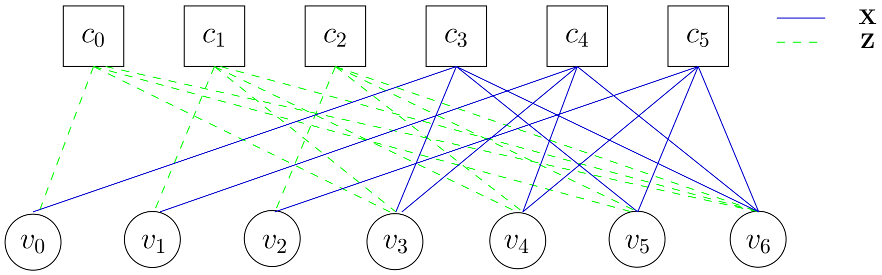


Figure 7.11: Tanner graph of the 7-qubit Steane code.

7.6 Modified Non-Binary Decoding

Let us recall from Section 7.3.3 that homogeneous CSS-type QLDPC codes, which include both the dual-containing construction as well as the EA-QLDPC codes, have an excessive number of short cycles. The ith and $(i+m/2)$th rows of the associated PCM $\hat{H}$ are related by a multiple of ω, i.e. we have $\hat{H}_i = \omega \hat{H}_{i+m/2}$, as seen in Eq. (7.23). For example, consider the 7-qubit Steane code [35], which is derived from the $(7,4)$ Hamming code. The PCM of a classical $(7,4)$ Hamming code is given by:

$$H'_z = \begin{pmatrix} 1 & 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 \end{pmatrix}. \tag{7.39}$$

Consequently, according to Eq. (7.23), the corresponding PCM of the 7-qubit Steane code is:

$$\hat{H} = \begin{pmatrix} \omega & 0 & 0 & \omega & 0 & \omega & \omega \\ 0 & \omega & 0 & \omega & \omega & 0 & \omega \\ 0 & 0 & \omega & 0 & \omega & \omega & \omega \\ 1 & 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 \end{pmatrix}, \tag{7.40}$$

whose Tanner graph is plotted in Figure 7.11. As gleaned from Figure 7.11, cycles of length 4 exist between all the variable nodes connected to the checks c_i and c_{i+3}. The dual-containing nature of Steane code also results in some additional short cycles. However, here we focus our attention only on the cycles resulting from the homogeneous CSS structure. To alleviate the impact of these short cycles, we propose a modified Tanner graph, which amalgamates the check nodes c_i and $c_{i+m/2}$ into a supernode, thereby eliminating the cycles. The resultant modified Tanner graph is given in Figure 7.12. Based on the modified Tanner graph of Figure 7.12, the horizontal messages exchanged between the supernodes $(c_i, c_{i+m/2})$ and the variable nodes v_t aim for satisfying both the checks c_i and $c_{i+m/2}$ simultaneously. Therefore, we have to modify Eq. (7.19) and (7.20) of the non-binary BP accordingly.

Since we have $\hat{H}_i = \omega \hat{H}_{i+m/2}$, the non-binary syndromes $\hat{S}_i$ and $\hat{S}_{i+m/2}$ are also related similarly, i.e. we have $\hat{S}_i = \omega \hat{S}_{i+m/2}$. Based on this relation, Table 7.6 enlists the possible values of $\hat{S}_{i+m/2}$

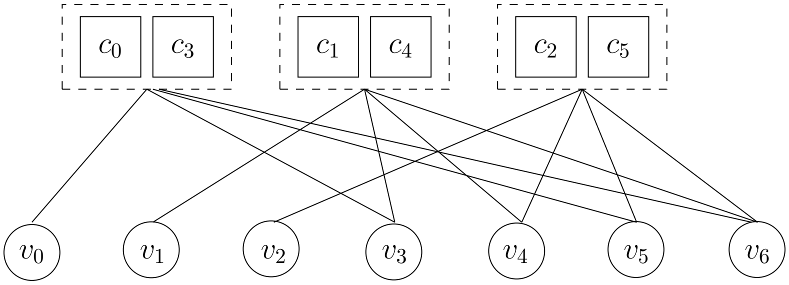


Figure 7.12: Modified Tanner graph of 7-qubit Steane code. Check nodes c_i and $c_{i+m/2}$ are combined to form a supernode.

$\hat{S}_i$	$\hat{S}_{i+m/2}$	S_i	$S_{i+m/2}$
0	0	0	0
1	$\overline{\omega}$	0	1
ω	1	1	0
$\overline{\omega}$	ω	1	1

Table 7.6: List of all possible values of $\hat{S}_i$ and the corresponding values of $\hat{S}_{i+m/2}$ and the binary syndromes $S_i = \text{Tr}(\hat{S}_i)$ and $S_{i+m/2} = \text{Tr}(\hat{S}_{i+m/2})$.

for all the possible values of $\hat{S}_i$ along with the corresponding binary syndromes $S_i = \text{Tr}(\hat{S}_i)$ and $S_{i+m/2} = \text{Tr}(\hat{S}_{i+m/2})$. As gleaned from Table 7.6, for each value of S_i (or $S_{i+m/2}$), there are two possible values of $\hat{S}_i$ (or $\hat{S}_{i+m/2}$). Recall from Section 7.3.2 that this is because $\text{Tr}(0) = \text{Tr}(1) = 0$, while $\text{Tr}(\omega) = \text{Tr}(\overline{\omega}) = 1$. On the other hand, for every pair of $(S_i, S_{i+m/2})$, there is a unique value of $\hat{S}_i$ and $\hat{S}_{i+m/2}$. Consequently, for the supernode $C_i = (c_i, c_{i+m/2})$, the PDFs of Eq. (7.19) and (7.20) may be modified as follows:

- If the observed channel syndromes are $(S_i, S_{i+m/2}) = (0, 0)$, then:

$$PDF\{\check{m}^{\hat{a}_s}_{C_i \to v_t}\} = \bigotimes_{v_{t'}} PDF\{\check{m}^{\hat{a}_s}_{v_{t'} \to C_i}\}. \tag{7.41}$$

- If the observed channel syndromes obey $(S_i, S_{i+m/2}) = (0, 1)$, then we have:

$$PDF\{\check{m}^{\hat{a}_s}_{C_i \to v_t}\} = \bigotimes_{v_{t'}} PDF\{\check{m}^{\hat{a}_s+1}_{v_{t'} \to C_i}\}. \tag{7.42}$$

- If the observed channel syndromes satisfy $(S_i, S_{i+m/2}) = (1, 0)$, then we arrive at:

$$PDF\{\check{m}^{\hat{a}_s}_{C_i \to v_t}\} = \bigotimes_{v_{t'}} PDF\{\check{m}^{\hat{a}_s+\omega}_{v_{t'} \to C_i}\}. \tag{7.43}$$

- If the observed channel syndromes are $(S_i, S_{i+m/2}) = (1,1)$, then:

$$PDF\{\check{m}^{\hat{a}_s}_{C_i \to v_t}\} = \bigotimes_{v_{t'}} PDF\{\check{m}^{\hat{a}_s+\overline{\omega}}_{v_{t'} \to C_i}\}. \tag{7.44}$$

Here $\hat{a}_s = (\hat{H}_{it} \times \overline{\hat{a}})$ for $\hat{a} \in \{0, 1, \omega, \overline{\omega}\}$.

Hence, Eq. (7.41) to (7.44) ensure that both the constituent check nodes c_i and $c_{i+m/2}$ of the supernode C_i are satisfied simultaneously. This is achieved without any additional complexity overhead. In fact, our proposed method requires less computations than the standard non-binary BP, because the number of check nodes is reduced to half.

Let us consider the Steane code of Eq. (7.40) for explaining the decoding procedure. Assume that when the 7-qubit codeword is transmitted over a quantum depolarizing channel having a depolarizing probability of $p = 0.26$, an **X** error is inflicted on the first qubit, i.e. we have $\mathcal{P} =$ **XIIIIII**. Using Eq. (7.16), the observed syndrome may be computed as:

$$S = \text{Tr}\left(\begin{pmatrix} \omega & 0 & 0 & \omega & 0 & \omega & \omega \\ 0 & \omega & 0 & \omega & \omega & 0 & \omega \\ 0 & 0 & \omega & 0 & \omega & \omega & \omega \\ 1 & 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 \end{pmatrix} \cdot \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}\right) = \text{Tr}\begin{pmatrix} \omega \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}. \tag{7.45}$$

We first run the standard non-binary BP on the Tanner graph of Figure 7.11 for estimating the channel error. The non-binary BP algorithm proceeds as follows:

- **Initialization:** The messages $m^{\hat{a}}_{v_t \to c_i}$, which are sent from the variable nodes $v_t \in \{v_0, v_1, \ldots, v_6\}$ to the check nodes $c_i \in \{c_0, c_1, \ldots, c_5\}$ for $\hat{a} \in \{0, 1, \omega, \overline{\omega}\}$, are initialized according to the channel depolarizing probability of $p = 0.26$, i.e. we have:

$$m^{\hat{a}}_{v_t \to c_i} = \begin{cases} 0.74, & \text{if } \hat{a} = 0 \\ 0.0867, & \text{if } \hat{a} \in \{1, \omega, \overline{\omega}\}. \end{cases} \tag{7.46}$$

- **Horizontal message exchange:** The horizontal messages $m^{\hat{a}}_{c_i \to v_t}$ equivalent to Eq. (7.12), which are sent from the check nodes c_i to the variable nodes v_t, may be computed using the FFT-based algorithm of [189]. The algorithm is briefly outlined below:

  1. **PDF of $\check{m}^{\hat{a}_s}_{v_t \to c_i}$:** Recall from Section 7.3.2 that we have:

  $$\hat{a}_s = \hat{H}_{it} \times \overline{\hat{a}}. \tag{7.47}$$

  Consequently, the PDF of $\check{m}^{\hat{a}_s}_{v_t \to c_i}$ can be obtained by permuting the corresponding PDF of $m^{\hat{a}}_{v_t \to c_i}$ according to the value of $\hat{H}_{it}$ using Eq. (7.47). Let us consider the PDF of the message $m^{\hat{a}}_{v_0\to c_0}$, which is equivalent to $(m^{0}_{v_0\to c_0}, m^{1}_{v_0\to c_0}, m^{\omega}_{v_0\to c_0}, m^{\overline{\omega}}_{v_0\to c_0})$. The corresponding entry in $\hat{H}$ is $\hat{H}_{00} = \omega$. Hence, using Eq. (7.47) and Table 4.7, we get $\hat{a}_s = (0, \omega, 1, \overline{\omega})$ for $\hat{a} = (0, 1, \omega, \overline{\omega})$. This implies that the PDF of $\check{m}^{\hat{a}_s}_{v_0\to c_0}$ is equivalent to $(m^{0}_{v_0\to c_0}, m^{\omega}_{v_0\to c_0}, m^{1}_{v_0\to c_0}, m^{\overline{\omega}}_{v_0\to c_0})$. For the $\hat{H}$ of Eq. (7.40), we may generalize the computation of $\check{m}^{\hat{a}_s}_{v_t\to c_i}$ as follows:

$$PDF\{\check{m}^{\hat{a}_s}_{v_t\to c_i}\} = \begin{cases} (m^{0}_{v_t\to c_i}, m^{\omega}_{v_t\to c_i}, m^{1}_{v_t\to c_i}, m^{\overline{\omega}}_{v_t\to c_i}), & \text{if } c_i \in \{c_0, c_1, c_2\} \\ (m^{0}_{v_t\to c_i}, m^{1}_{v_t\to c_i}, m^{\omega}_{v_t\to c_i}, m^{\overline{\omega}}_{v_t\to c_i}), & \text{if } c_i \in \{c_3, c_4, c_5\}. \end{cases} \tag{7.48}$$

Furthermore, for the initial PDF of Eq. (7.46), Eq. (7.48) reduces to:

$$\check{m}^{\hat{a}_s}_{v_t\to c_i} = \begin{cases} 0.74, & \text{if } \hat{a}_s = 0 \\ 0.0867, & \text{if } \hat{a}_s \in \{1, \omega, \overline{\omega}\}, \end{cases} \tag{7.49}$$

for $c_i \in \{c_0, c_1, \ldots, c_5\}$.

2. **FFT of the PDF of $\check{m}^{\hat{a}_s}_{v_t\to c_i}$:** Recall from Section 7.3.2 that the convolution operation required in Eq. (7.19) and Eq. (7.20) is equivalent to the multiplication of the corresponding PDFs in the frequency domain. The FFT of the PDF of Eq. (7.49) can be computed using the FFT matrix as follows:

$$\mathcal{F}\{\check{m}^{\hat{a}_s}_{v_t\to c_i}\} = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{pmatrix} \cdot \begin{pmatrix} \check{m}^{0}_{v_t\to c_i} \\ \check{m}^{1}_{v_t\to c_i} \\ \check{m}^{\omega}_{v_t\to c_i} \\ \check{m}^{\overline{\omega}}_{v_t\to c_i} \end{pmatrix}, \tag{7.50}$$

where $\mathcal{F}$ denotes the FFT operation. Hence, the FFT of the PDF of Eq. (7.49) is equivalent to:

$$\mathcal{F}\{\check{m}^{\hat{a}_s}_{v_t\to c_i}\} = \begin{pmatrix} 1 \\ 0.6533 \\ 0.6533 \\ 0.6533 \end{pmatrix}. \tag{7.51}$$

3. **Convolution of PDFs:** The convolution operations of Eq. (7.19) and Eq. (7.20), which are invoked for computing the horizontal messages related to the variable node v_t, can be carried out using FFT as follows:

$$\bigotimes_{v_{t'}} PDF\{\check{m}^{\hat{a}_s}_{v_{t'}\to c_i}\} \equiv \mathcal{F}^{-1}\left\{\prod_{v_{t'}} \mathcal{F}\{\check{m}^{\hat{a}_s}_{v_{t'}\to c_i}\}\right\}, \tag{7.52}$$

where $\mathcal{F}^{-1}$ denotes the Inverse FFT (IFFT) operation and $v_{t'} \in V(c_i) \setminus v_t$. Given the $\hat{H}$ of Eq. (7.40) and the FFT of Eq. (7.51), we get:

$$\prod_{v_{t'}} \mathcal{F}\{\check{m}^{\hat{a}_s}_{v_{t'}\to c_i}\} \equiv \begin{pmatrix} 1 \\ 0.2788 \\ 0.2788 \\ 0.2788 \end{pmatrix}. \tag{7.53}$$

Then, the inverse FFT of Eq. (7.53) is computed by multiplying with the FFT matrix, which is the same as that in Eq. (7.50). More explicitly, we have:

$$\mathcal{F}^{-1}\left\{\prod_{v_{t'}}\mathcal{F}\{\check{m}^{\hat{a}_s}_{v_{t'}\to c_i}\}\right\} = \begin{pmatrix}1 & 1 & 1 & 1\\ 1 & -1 & 1 & -1\\ 1 & 1 & -1 & -1\\ 1 & -1 & -1 & 1\end{pmatrix} \cdot \begin{pmatrix}1\\ 0.2788\\ 0.2788\\ 0.2788\end{pmatrix} = \begin{pmatrix}1.8364\\ 0.7212\\ 0.7212\\ 0.7212\end{pmatrix}. \tag{7.54}$$

Finally, the PDF of Eq. (7.54) is normalized to yield the output of Eq. (7.52), i.e. we get:

$$\bigotimes_{v_{t'}} PDF\{\check{m}^{\hat{a}_s}_{v_{t'}\to c_i}\} \equiv \begin{pmatrix}0.4591\\ 0.1803\\ 0.1803\\ 0.1803\end{pmatrix}. \tag{7.55}$$

4. **PDF of $\check{m}^{\hat{a}_s}_{c_i\to v_t}$:** The PDF of the messages $\check{m}^{\hat{a}_s}_{c_i\to v_t}$ may be computed using Eq. (7.19) or Eq. (7.20) depending on the value of the syndrome observed, which was computed in Eq. (7.45). Since the syndrome of Eq. (7.45) is 1 for the first check node c_0, we use Eq. (7.20) for computing the PDF of the messages emerging from the check node c_0. Therefore, we get:

$$\check{m}^{\hat{a}_s}_{c_0\to v_t} = \begin{pmatrix}0.1803\\ 0.1803\\ 0.3197\\ 0.3197\end{pmatrix}. \tag{7.56}$$

   Furthermore, the syndrome of Eq. (7.45) has a value of 0 for all other check nodes. Therefore, we use Eq. (7.19) for $c_i \neq c_0$, which yields the following PDF:

$$\check{m}^{\hat{a}_s}_{c_i\to v_t} = \begin{pmatrix}0.3197\\ 0.3197\\ 0.1803\\ 0.1803\end{pmatrix}. \tag{7.57}$$

5. **PDF of $m^{\hat{a}}_{c_i\to v_t}$:** For the sake of retrieving the messages $m^{\hat{a}}_{c_i\to v_t}$ from the PDF of $\check{m}^{\hat{a}_s}_{c_i\to v_t}$, the resultant PDFs of Eq. (7.56) and Eq. (7.57) have to permuted as we did in Step 1. More specifically, the permutation operation, which is required for this step, is the reverse of the permutation operation carried out in Step 1. Let us consider the check nodes $c_i \in \{c_0, c_1, c_2\}$, for which the non-zero values of $\hat{H}_{it}$ are always equal to ω (or equivalently all the branches emerging from these check nodes in the Tanner graph of Figure 7.11 are labeled with the Pauli-**Z** operator). Furthermore, recall from Step 1 that $\hat{a}_s = (0, \omega, 1, \overline{\omega})$ for $\hat{a} = (0, 1, \omega, \overline{\omega})$, when $\hat{H}_{it} = \omega$. This implies that the PDF of $m^{\hat{a}}_{c_i\to v_t}$ is equivalent to $(\check{m}^{0}_{c_i\to v_t}, \check{m}^{\omega}_{c_i\to v_t}, \check{m}^{1}_{c_i\to v_t}, \check{m}^{\overline{\omega}}_{c_i\to v_t})$, for $c_i \in \{c_0, c_1, c_2\}$. For all other check nodes, the PDF

of $m^{\hat{a}}_{c_i \to v_t}$ is same as that of $\check{m}^{\hat{a}_s}_{c_i \to v_t}$, because $\hat{H}_{it} = 1$. Therefore, the resultant PDFs are as follows:

$$m^{\hat{a}}_{c_i \to v_t} = \begin{pmatrix} 0.1803 \\ 0.3197 \\ 0.1803 \\ 0.3197 \end{pmatrix}, \tag{7.58}$$

for $c_i = c_0$, while we have:

$$m^{\hat{a}}_{c_i \to v_t} = \begin{pmatrix} 0.3197 \\ 0.1803 \\ 0.3197 \\ 0.1803 \end{pmatrix}, \tag{7.59}$$

for $c_i \in \{c_1, c_2\}$, and we have:

$$m^{\hat{a}}_{c_i \to v_t} = \begin{pmatrix} 0.3197 \\ 0.3197 \\ 0.1803 \\ 0.1803 \end{pmatrix}, \tag{7.60}$$

for the remaining check nodes $c_i \in \{c_3, c_4, c_5\}$.

- **Vertical message exchange:** We next compute the vertical messages $m^{\hat{a}}_{v_t \to c_i}$ using Eq. (7.13). For example, consider the message $m^{\hat{a}}_{v_0 \to c_0}$, which is destined from the variable node v_0 to the check node c_0. Since the variable node v_0 is only connected to c_0 and c_3 in the Tanner graph of Figure 7.11, the message $m^{\hat{a}}_{v_0 \to c_0}$ may be computed as:

$$m^{\hat{a}}_{v_0 \to c_0} = K \, \mathrm{P}_{\mathrm{ch}}(P_0 = \hat{a}) \times m^{\hat{a}}_{c_3 \to v_0} = \begin{pmatrix} 0.8005 \\ 0.0937 \\ 0.0529 \\ 0.0529 \end{pmatrix}. \tag{7.61}$$

- **Element-wise marginal probability:** The element-wise marginal probabilities of the error on the variable node v_t, given the observed syndrome S, may be computed using Eq. (7.14). Let us consider again the variable node v_0, which is connected to check nodes c_0 and c_3. Consequently, the resultant marginal distribution of the error P_t inflicted on the variable node v_t may be computed as:

$$\mathrm{P}(P_0 = \hat{a}|S) = K \, \mathrm{P}_{\mathrm{ch}}(P_0 = \hat{a}) \times m^{\hat{a}}_{c_0 \to v_0} \times m^{\hat{a}}_{c_3 \to v_0} = \begin{pmatrix} 0.7189 \\ 0.1493 \\ 0.0475 \\ 0.0842 \end{pmatrix}. \tag{7.62}$$

The process is repeated for all the variable nodes and the resultant marginalized probabilities are tabulated in Table 7.7.

t	$\hat{a}=0$	$\hat{a}=1$	$\hat{a}=\omega$	$\hat{a}=\overline{\omega}$	$\tilde{P}_t$
0	0.7189	0.1493	0.0475	0.0842	0
1	0.8552	0.0565	0.0565	0.03185	0
2	0.8552	0.0565	0.0565	0.03185	0
3	0.8392	0.0983	0.0313	0.0313	0
4	0.9205	0.0343	0.0343	0.0109	0
5	0.8392	0.0983	0.0313	0.0313	0
6	0.9100	0.0601	0.0191	0.0108	0

Table 7.7: Marginal probability $\mathrm{P}(P_t = \hat{a}|S)$ after the first iteration, when the standard non-binary BP decoding algorithm is invoked over the Tanner graph of the 7-qubit Steane code for transmission through a depolarizing channel having $p = 0.26$, which inflicts an $\mathbf{X}$ error on the first qubit, i.e. we have $\mathcal{P} = \mathbf{XIIIIII}$.

- **Hard decision & syndrome check:** Finally, a hard decision is made for the sake of finding the most likely error $\tilde{P}_t$, which maximizes the marginal probability computed in the previous step. The resultant values of $\tilde{P}_t$ are listed in the last column of Table 7.7. More specifically, the probability of 'no-error' dominates for all the variable nodes. The syndrome corresponding to the resultant estimated error $\tilde{P}_t$ does not match the observed syndrome S of Eq. (7.45). Hence, the algorithm repeats itself from the horizontal message exchange step.

Figure 7.13(a) plots the resultant marginal probability $\mathrm{P}(P_t = \hat{a}|S)$ for the first qubit, as the iterations proceed. As gleaned from Figure 7.13(a), the standard decoding algorithm fails to converge. We next invoke our modified non-binary BP algorithm for the sake of analyzing the impact of our proposed algorithm. Recall from Figure 7.12 that the check nodes c_i and c_{i+3} are amalgamated into a single supernode C_i. The corresponding observed syndrome values of Eq. (7.45) are also amalgamated, which yields $(S_0, S_3) = (1, 0)$, $(S_1, S_4) = (0, 0)$ and $(S_2, S_5) = (0, 0)$. Consequently, the modified BP differs from the standard non-binary in the Step 4 of the 'horizontal message exchange', since it takes into account the amalgamated supernodes, rather than the individual check nodes. Using Eq. (7.41) to Eq. (7.44), Step 4 of the 'horizontal message exchange' may be carried out as follows:

- **PDF of $\check{m}^{\hat{a}_s}_{C_i\to v_t}$:** Since the syndrome observed for the supernode C_0 is $(S_0, S_3) = (1, 0)$, we use Eq. (7.43) for computing the PDF of the messages emerging from this supernode. Consequently, we arrive at:

$$\check{m}^{\hat{a}_s}_{C_0\to v_t} = \bigotimes_{v_{t'}} PDF\{\check{m}^{\hat{a}_s+\omega}_{v_{t'}\to C_0}\} = \begin{pmatrix} 0.1803 \\ 0.1803 \\ 0.4591 \\ 0.1803 \end{pmatrix}. \tag{7.63}$$

t	$\hat{a}=0$	$\hat{a}=1$	$\hat{a}=\omega$	$\hat{a}=\overline{\omega}$	$\tilde{P}_t$
0	0.1946	0.6525	0.0764	0.0764	0
1	0.8788	0.0404	0.0404	0.0404	0
2	0.8788	0.0404	0.0404	0.0404	0
3	0.8271	0.0969	0.0380	0.0380	0
4	0.9486	0.0171	0.0171	0.0171	0
5	0.8271	0.0969	0.0380	0.0380	0
6	0.9241	0.0425	0.0167	0.0167	0

Table 7.8: Marginal probability $\mathrm{P}(P_t = \hat{a}|S)$ after the first iteration, when the modified non-binary BP decoding algorithm is invoked over the Tanner graph of the 7-qubit Steane code for transmission through a depolarizing channel having $p = 0.26$, which inflicts an **X** error on the first qubit, i.e. we have $\mathcal{P} = \mathbf{XIIIIII}$.

Furthermore, since the syndrome is $(S_i, S_{i+3}) = (0, 0)$ for all other supernodes, we use Eq. (7.41) for computing the corresponding PDFs. Hence, we get:

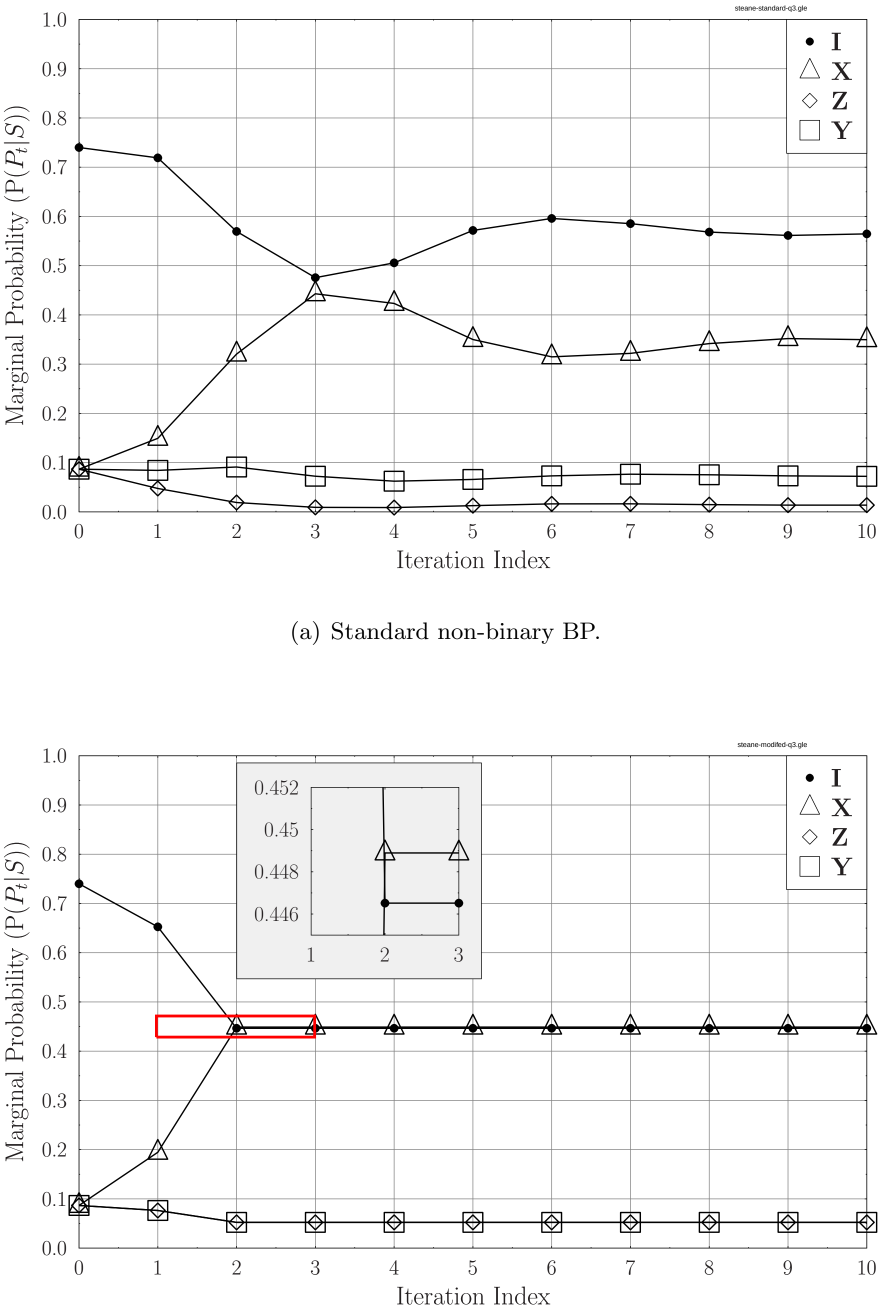
$$\check{m}^{\hat{a}_s}_{C_i \to v_t} = \bigotimes_{v_{t'}} PDF\{\check{m}^{\hat{a}_s}_{v_{t'} \to C_i}\} = \begin{pmatrix} 0.4591 \\ 0.1803 \\ 0.1803 \\ 0.1803 \end{pmatrix}, \tag{7.64}$$

for $C_i \in \{C_1, C_2\}$.

The rest of the decoding algorithm is same as the standard non-binary BP, except that we only have three supernodes in the modified Tanner graph of Figure 7.12 in contrast to the six check nodes of Figure 7.11. The resultant marginalized probabilities are tabulated in Table 7.8, while Figure 7.13(b) plots the marginal probability for the first qubit, as the iterations proceed. We may observe in Figure 7.13(b) that our modified BP algorithm converges to the correct estimate in two iterations. On the other hand, the standard non-binary decoder initially tends to converge towards the correct solution. However, it starts diverging from the correct estimate after the third iteration, because the probability values become highly correlated (or over-confident) due to the presence of numerous short cycles.

7.7 Reweighted BP for Graphs Exhibiting Cycles

Belief propagation is capable of providing a reasonably good approximation to the optimization problem of Eq. (7.10), provided that the underlying Tanner graph has a sufficiently high girth. However,



(a) Standard non-binary BP.

(b) Modified non-binary BP.

Figure 7.13: Evolution of the marginal probability for the first qubit of the 7-qubit Steane code for transmission through a depolarizing channel having $p = 0.26$, which inflicts an **X** error on the first qubit, i.e. we have $\mathcal{P} = \mathbf{XIIIIII}$. Standard BP fails to converge, while our modified BP converges to the correct solution in two iterations.

it is not guaranteed to converge or may converge onto an incorrect solution in the presence of cycles [196, 197]. Furthermore, it may require a large number of iterations for achieving convergence, especially in the high noise regime, thereby imposing a higher complexity. These shortcomings of the classic BP algorithm are primarily due to the fact that the BP messages become dependent with time when short cycles exist in the Tanner graph. Alternatively, we may refer to the messages as being 'over-confident' or 'over-estimated'. To alleviate the impact of this over-confidence, Wainwright *et al.* [196] conceived the Tree-ReWeighted Belief Propagation (TRW-BP) method for pair-wise interactions, which improves the convergence of the classic BP by reweighting the edges of the underlying graph with their Edge Appearance Probabilities (EAP)[9]. The TRW-BP algorithm was extended to higher-order interactions in [168, 169], whereby EAPs were replaced by the Factor Appearance Probabilities (FAPs) of the nodes[10]. Based on this extended TRW-BP, Wymeersch *et al.* re-formulated the vertical message exchange step of the classic BP (Eq. (7.13)) as [168, 169]:

$$m^a_{v_t \to c_i} = K\ \mathrm{P}_{\mathrm{ch}}(P_t = a) \left(m^a_{c_i \to v_t}\right)^{\rho_i - 1} \prod_{c_{i'} \in C(v_t) \backslash c_i} \left(m^a_{c_{i'} \to v_t}\right)^{\rho_{i'}}, \tag{7.65}$$

where ρ_i is the FAP of the ith check node. Similarly, the computation of the element-wise marginal probability (Eq. (7.14)) was modified as:

$$\mathrm{P}(P_t = a|S) = K\ \mathrm{P}_{\mathrm{ch}}(P_t = a) \prod_{c_i \in C(v_t)} \left(m^a_{c_i \to v_t}\right)^{\rho_i}. \tag{7.66}$$

Both Eq. (7.65) and (7.66) reduces to the classic BP for $\rho_i = 1\ \forall i$.

The TRW-BP technique requires the optimization of ρ_i for all nodes. To reduce this optimzation task, Wymeersch *et al.* [168, 169] also proposed the URW-BP, which invokes a uniform FAP value for all the nodes, where we have $\rho_i = \rho\ \forall i$. Various other variations of TRW-BP have been investigated in [198, 199, 200, 201] for classical binary LDPC codes, which demonstrate that the TRW-BP effectively improves the convergence of binary LDPC codes, when the number of iterations is not too high. Inspired by these results, in Section 7.8 we also analyze the impact of URW-BP on the non-binary decoding of quantum LDPC codes, which are known to have unavoidable short cycles.

7.8 Results and Discussions II

7.8.1 Modified Non-Binary Decoding

For the sake of quantifying the attainable performance gain of our modified non-binary BP of Section 7.6, in this section we compare its performance in conjunction with the decoding algorithms of Section 7.3. Our first system of Table 7.9 relies on Mackay's 1/2-rate $[800, 400]$ bicycle code having a row weight of 30. The corresponding WER performance recorded for varying channel depolarizing probabilities is plotted in Figure 7.14, where we have considered the following decoders:

[9] EAP of an edge represents the probability of appearance of that edge in a randomly chosen spanning tree.

[10] FAP denotes the appearance probability of a node in the collection of trees [168, 169].

QLDPC Matrix	
Code Construction	Mackay's bicycle code
Coded qubits	$n = 800$
Information qubits	$k = 400$
E-bits	$c = 0$
Row weight	30
QLDPC Decoder	
Standard decoding iterations	$\mathtt{I}_{\max} = 90$
Perturbation iterations	$T_{\text{pert}=40}$
Random perturbation strength	$\delta = 0.1$
Maximum no. of feedbacks	$n_a = 40$

Table 7.9: System I - Simulation parameters.

1. **Binary:** the binary BP decoding algorithm of Section 7.3.1,
2. **Standard Non-Binary:** the non-binary BP decoding algorithm of Section 7.3.2,
3. **Random Perturbation:** the random perturbation technique [85] of Section 7.3.3,
4. **Enhanced Feedback:** the enhanced feedback method [86] of Section 7.3.3,
5. **Modified Non-Binary:** our modified non-binary BP of Section 7.6,
6. **Modified & Enhanced Feedback:** our modified non-binary BP of Section 7.6 amalgamated with the enhanced feedback method [86] of Section 7.3.3.

For all the decoding schemes, we have used a maximum of $\mathtt{I}_{\max} = 90$ iterations. Furthermore, for both the 'Random Perturbation' as well as for the 'Enhanced Feedback', we set $T_{\text{pert}} = 40$, while the random perturbation strength was set to $\delta = 0.1$ and the maximum number of feedbacks to $n_a = 40$ for the 'Enhanced Feedback'[11]. These simulation parameters are tabulated in Table 7.9. Each decoding algorithm iterates until either a valid error is found or the maximum number of iterations is reached. Furthermore, the WER metric here counts both the detected as well as the undetected block errors.

We may observe in Figure 7.14 that the 'Binary' decoder exhibits the worse performance. Using the 'Binary' decoder, we achieve a WER of 10^{-4} at a channel depolarizing probability of $p = 0.0085$, which increases to $p = 0.01075$ with the 'Standard Non-Binary' decoder. This is equivalent to a

[11] We have used the decoding parameters of [86].

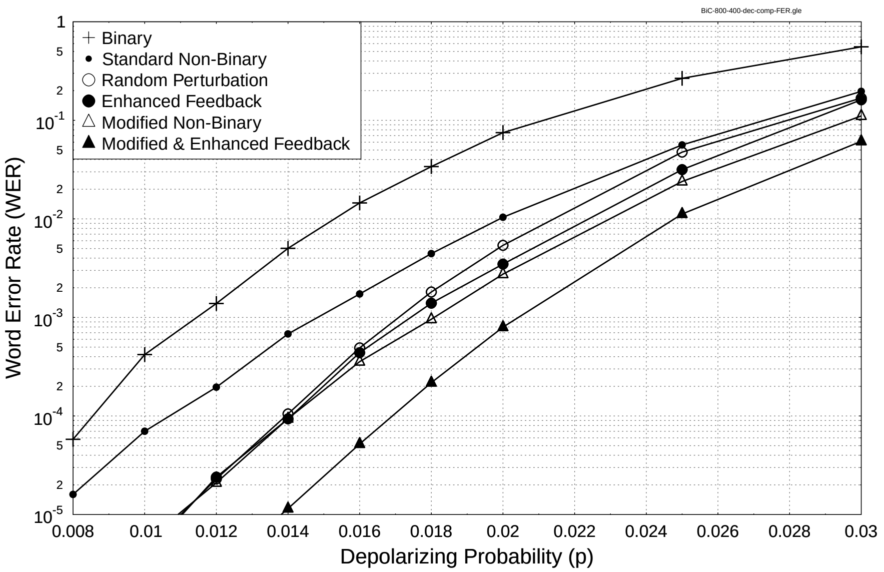


Figure 7.14: Achievable WER performance comparison of the modified BP with the existing decoding schemes, using the simulation parameters of Table 7.9.

$(\frac{(0.01075-0.0085)}{0.0085} \times 100) = 26\%$ depolarizing probability increase that the decoder can cope with. Furthermore, the 'Random Perturbation', the 'Enhanced Feedback' and the 'Modified Non-Binary' decoders have a similar performance at low noise levels, increasing the tolerable depolarizing probability to $p = 0.014$ at a WER of 10^{-4}, which corresponds to a $(\frac{(0.014-0.01075)}{0.01075} \times 100) = 30\%$ increase of p at WER $= 10^{-4}$ with respect to the 'Standard Non-Binary' decoder. Furthermore, with the 'Modified & Enhanced Feedback' configuration, the tolerable depolarizing probability increases to $p = 0.017$ at a WER of 10^{-4}, which is equivalent to about $(\frac{(0.017-0.014)}{0.014} \times 100) = 21\%$ increase with respect to $p = 0.014$. Table 7.10[12] summarizes these results.

The performance of our 'Modified Non-Binary' BP at a WER of 10^{-4} is similar to that of the heuristic methods, namely 'Random Perturbation' and 'Enhanced Feedback'. However, the 'Modified Non-Binary' technique imposes a lower decoding complexity in terms of the average number of decoding iterations, which is evidenced in Figure 7.15. Consequently, our 'Modified Non-Binary' BP converges faster than the existing decoding schemes. In particular, in the high-noise regime, our 'Modified Non-Binary' decoder outperforms both the 'Random Perturbation' and the 'Enhanced Feedback' in terms of its WER performance recorded in Figure 7.14 as well as in terms of the average number of iterations seen in Figure 7.15. As compared to the 'Standard Non-Binary' decoder, the 'Modified Non-Binary' algorithm always yields a lower WER and invokes on average less decoding iterations. We may observe furthermore in Figure 7.15 that the amalgamated 'Modified & Enhanced Feedback' invokes less iterations as compared to the 'Enhanced Feedback', while the performance of the former

[12] 'With respect to' is abbreviated as 'w.r.t.' in Table 7.10.

Dec. No.	**Decoding Method**	p	**Improvement**
1	Binary	0.0085	-
2	Standard Non-Binary	0.01075	26% w.r.t. Dec. 1
3	Random Perturbation	0.014	30% w.r.t. Dec. 2
4	Enhanced Feedback	0.014	30% w.r.t. Dec. 2
5	Modified Non-Binary	0.014	30% w.r.t. Dec. 2
6	Modified & Enhanced Feedback	0.017	21% w.r.t. Dec. 5

Table 7.10: Achievable depolarizing probability (p) at a WER of 10^{-4}, based on Figure 7.14.

is also superior in terms of the WER curve of Figure 7.14. This is again due to the fact that the modified BP of Section 7.6 facilitates faster convergence as compared to the standard non-binary decoding. More specifically, in the region of interest, i.e. for $p \leq 0.017$ corresponding to the desired WER of $\leq 10^{-4}$, the combination of the enhanced feedback method with our modified BP, namely 'Modified & Enhanced Feedback', imposes almost the same complexity as that imposed by the 'Modified Non-Binary' BP, when used on its own. However, the former exhibits a much lower WER than the latter. We compare furthermore the performance of all the decoding schemes at a depolarizing probability of $p = 0.016$ in Table 7.11.

Let us now compare the performance of the different decoding schemes in the context of our second system of Table 7.12, relying on the homogeneous EA-QLDPC code of [86] having $n = 816$, $k = 404$ and $e = 404$, which is derived from the Mackay's classical $(816, 408)$ LDPC, having a row weight of 6 and a column weight of 3. For all the decoding schemes, we have used a maximum of $\mathtt{I}_{\text{max}} = 90$ iterations. Furthermore, for both the the 'Random Perturbation' as well as for the 'Enhanced Feedback' methods, we set $T_{\text{pert}} = 40$, while the random perturbation strength was set to $\delta = 0.1$ and the maximum number of feedbacks $n_a = 81$ was used for the 'Enhanced Feedback' decoder. These simulation parameters are summarized in Table 7.12. The resultant WER performance curves are compared in Figure 7.16, while the average number of decoding iterations invoked for varying channel depolarizing probabilities are compared in Figure 7.17. As observed from Figure 7.16, the 'Binary' decoder achieves a WER of 10^{-4} at $p = 0.057$, which increases to $p = 0.069$ when the 'Standard Non-Binary' decoder is invoked. Consequently, the 'Standard Non-Binary' increases the tolerable depolarizing probability by $(\frac{(0.069-0.057)}{0.057} \times 100) = 21\%$ as compared to the 'binary' decoder. This is further increased to $p = 0.076$ in conjunction with the 'Random Perturbation', which corresponds to about $(\frac{(0.076-0.069)}{0.069} \times 100) = 10\%$ increase and to $p = 0.082$ for the 'Enhanced Feedback', which represents a $(\frac{(0.082-0.069)}{0.069} \times 100) = 19\%$ increase. By contrast, our 'Modified Non-Binary' BP exhibits a WER of 10^{-4} around $p = 0.085$, which corresponds to a $(\frac{(0.085-0.069)}{0.069} \times 100) = 23\%$

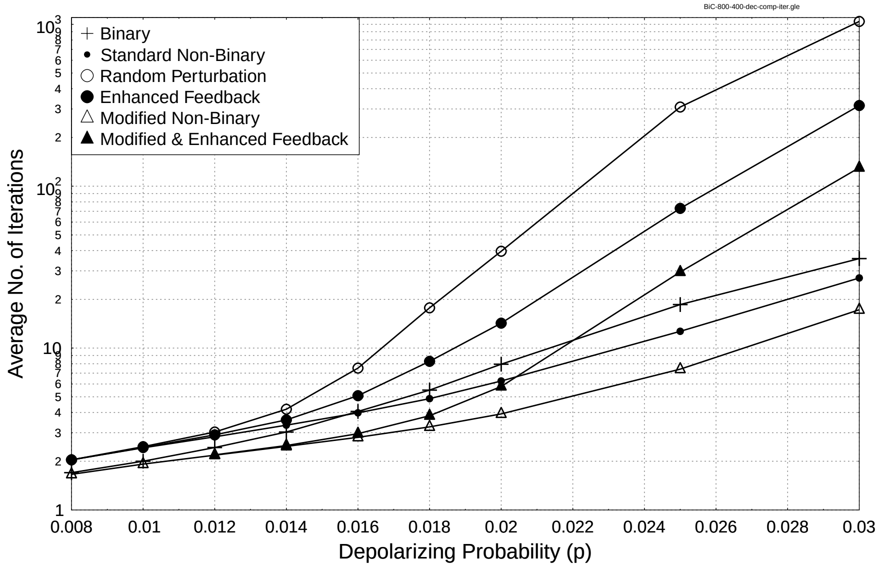


Figure 7.15: Comparison of the average number of decoding iterations invoked by the modified BP and the existing decoding schemes using the simulation parameters of Table 7.9.

Dec. No.	**Decoding Method**	**WER**	$\mathtt{I}_{\mathbf{avg}}$
1	Binary	1.5710^{-2}	3.98
2	Standard Non-Binary	1.4710^{-3}	4.28
3	Random Perturbation	4.5710^{-4}	7.52
4	Enhanced Feedback	4.4710^{-4}	5.08
5	Modified Non-Binary	3.6710^{-4}	2.81
6	Modified & Enhanced Feedback	5.2710^{-5}	2.96

Table 7.11: Performance comparison in terms of the achievable WER and the average number of decoding iterations ($\mathtt{I}_{\text{avg}}$) invoked at a depolarizing probability of $p = 0.016$, based on Figure 7.14 and Figure 7.15.

QLDPC Matrix	
Code Construction	Homogeneous EA-QLDPC
Coded qubits	$n = 816$
Information qubits	$k = 404$
E-bits	$c = 404$
Row weight	6
Column weight	3
QLDPC Decoder	
Standard decoding iterations	$\mathtt{I}_{\max} = 90$
Perturbation iterations	$T_{\text{pert}} = 40$
Random perturbation strength	$\delta = 0.1$
Maximum no. of feedbacks	$n_a = 81$

Table 7.12: System II - Simulation parameters.

increase as compared to the 'Standard Non-Binary' decoder. Using the heuristic enhanced feedback approach with our modified BP, namely 'Modified & Enhanced Feedback' provides a further increase to $p = 0.0945$, which represents a $(\frac{(0.0945-0.085)}{0.085} \times 100) = 11\%$ increase. These results are tabulated in Table 7.13. In terms of the average number of decoding iterations, our 'Modified Non-Binary' BP always outperforms both the 'Standard Non-Binary' decoder as well as the 'Random perturbation' and the 'Enhanced Feedback' solutions, as depicted in Figure 7.17.

7.8.2 Uniformly-Reweighted BP

Since bicycle codes exhibit numerous short cycles, we use our first system of Table 7.9 for the analysis of the URW-BP of Section 7.7, which is combined with our modified non-binary decoder of Section 7.6. More precisely, we amalgamate the horizontal exchange step of our modified non-binary BP with the vertical exchange step of the URW-BP.

We commence by heuristically determining the optimum value ρ of the FAP, which varies with both the channel depolarizing probability as well as with the maximum number of decoding iterations. Figure 7.18 shows the impact of ρ on the WER performance at varying channel depolarizing probabilities p for $\mathtt{I}_{\max} = 10$, 20 and 90 iterations. We may observe in Figure 7.18 that the WER varies with the value of ρ, attaining a minimum value at the optimum ρ. This optimum ρ is different for each p value, tending to move towards $\rho = 1$ as the value of p increases or as the maximum affordable number of iterations increases. The resultant values of ρ optimized for different channel depolarizing

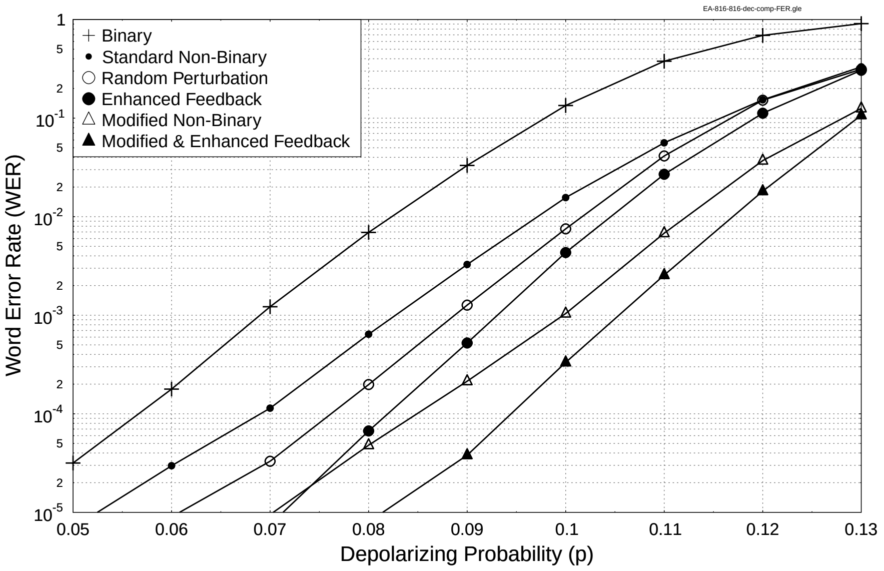


Figure 7.16: Achievable WER performance comparison of the modified BP with the existing decoding schemes, using the simulation parameters of Table 7.12.

Dec. No.	**Decoding Method**	p	**Improvement**
1	Binary	0.057	-
2	Standard Non-Binary	0.069	21% w.r.t. Dec. 1
3	Random Perturbation	0.076	10% w.r.t. Dec. 2
4	Enhanced Feedback	0.082	19% w.r.t. Dec. 2
5	Modified Non-Binary	0.085	23% w.r.t. Dec. 2
6	Modified & Enhanced Feedback	0.0945	11% w.r.t. Dec. 5

Table 7.13: Achievable depolarizing probability (p) at a WER of 10^{-4}, based on Figure 7.16.

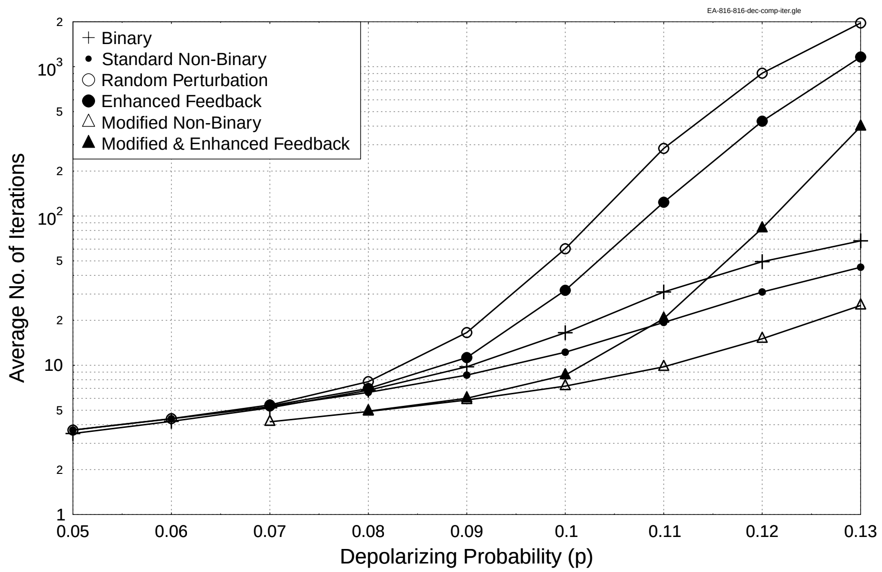


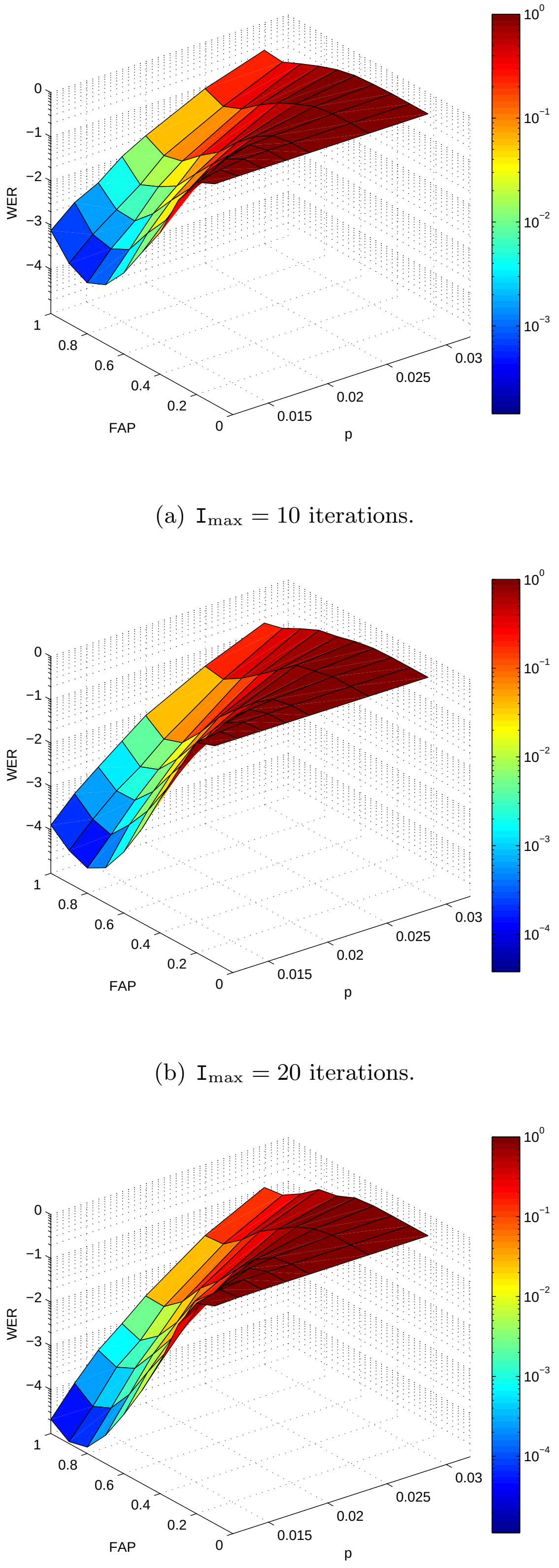
Figure 7.17: Comparison of the average number of decoding iterations invoked by the modified BP and the existing decoding schemes using the simulation parameters of Table 7.12.

$\mathtt{I}_{\max}$	Optimized ρ for different values of p						
	0.012	**0.014**	**0.016**	**0.018**	**0.02**	**0.025**	**0.03**
10	0.8	0.8	0.8	0.9	0.9	0.9	0.9
20	0.8	0.8	0.9	0.9	0.9	0.9	1
90	0.9	0.9	0.9	0.9	0.9	1	1

Table 7.14: Optimized ρ for different values of p and maximum number of iterations $\mathtt{I}_{\max}$ for System I of Table 7.9, based on the performance curves of Figure 7.18.

probabilities p and for different maximum number of iterations are summarized in Table 7.14.

To quantify the performance gain achieved with the aid of the URW-BP, we compare the performance of the optimized URW-BP to our modified non-binary BP in Figure 7.19 for $\mathtt{I}_{\max} = 10$, 20 and 90 iteration. Here the optimized URW-BP is based on the best values of ρ listed in Table 7.14. The performance curves of Figure 7.19 reveal that the improvement in WER is lower for higher values of p as well as for larger values of the maximum number of affordable iterations. For example, when a maximum of $\mathtt{I}_{\max} = 10$ decoding iterations are invoked at a WER of 10^{-3}, the URW-BP scheme increases $p = 0.0125$ to $p = 0.0155$, which is around a $(\frac{(0.0155-0.0125)}{0.0125} \times 100) = 24\%$ increase. By contrast, for a maximum of $\mathtt{I}_{\max} = 20$ iterations, URW-BP increases from $p = 0.015$ to $p = 0.017$ at a WER of 10^{-3}. This is equivalent to an increase of $(\frac{(0.017-0.015)}{0.015} \times 100) = 13\%$. Furthermore, at an even higher maximum number of iterations of $\mathtt{I}_{\max} = 90$, URW-BP achieves a WER of 10^{-3} at $p = 0.0185$,



(a) $\mathtt{I}_{\max} = 10$ iterations.

(b) $\mathtt{I}_{\max} = 20$ iterations.

(c) $\mathtt{I}_{\max} = 90$ iterations.

Figure 7.18: URW-BP optimization: impact of varying FAP values on the WER performance at various channel depolarizing probabilities p. URW-BP is amalgamated with the modified non-binary decoder and the performance is analyzed for the System I of Table 7.9.

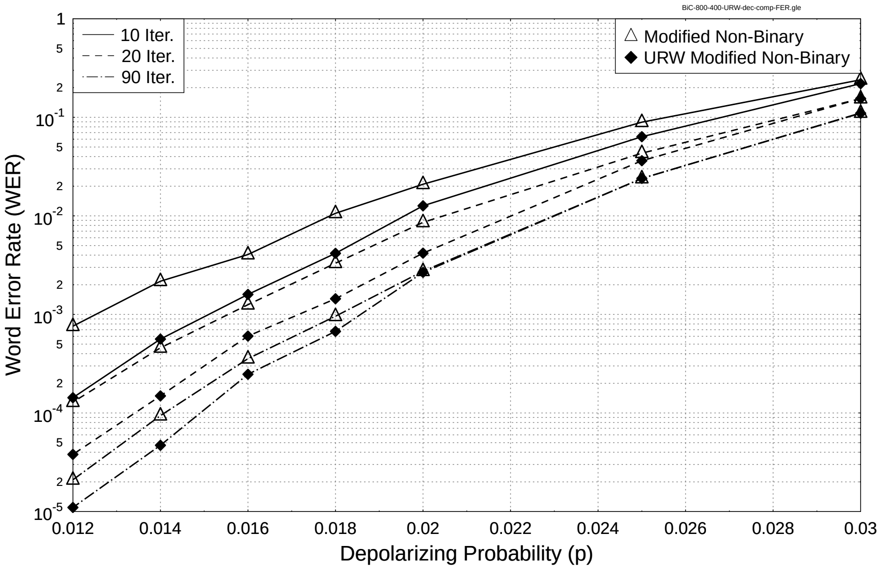


Figure 7.19: Achievable WER performance of the URW-BP, having the best values of ρ listed in Table 7.14, compared with the 'Modified Non-Binary', when used on its own. Performance is evaluated for the System I of Table 7.9 using $\mathtt{I}_{\text{max}} = 10$, 20 and 90 iterations.

which is only a $(\frac{(0.0185-0.018)}{0.018} \times 100) = 3\%$ increase as compared to the modified non-binary algorithm. These values are summarized in Table 7.15. Hence, the notion of reweighting the message probabilities is more beneficial at low depolarizing probabilities and for smaller values of the maximum affordable number of iterations. This is because at higher depolarizing probabilities (and similarly larger values of the maximum number of iterations), the messages are highly correlated.

	p at a WER of 10^{-3}		
$\mathtt{I}_{\text{max}}$	**Modified BP**	**URW-BP**	**Increase**
10	0.0125	0.0155	24%
20	0.015	0.017	13%
90	0.018	00185	3%

Table 7.15: Performance comparison of URW-BP with the modified BP for System I of Table 7.9, based on Figure 7.19.

7.9 Summary and Conclusions

Classical LDPC codes are known to exhibit a near-capacity performance at an affordable decoding complexity. This has spurred considerable interest in the design of QLDPC codes over the recent years. The sparseness of LDPC matrix is also particularly important for quantum codes because it facilitates fault tolerant decoding.

QLDPC codes may be constructed from the classical binary as well as quaternary codes, as discussed in the review of QLDPC construction methods in Section 7.2. The design guidelines for constructing QLDPC codes may be summarized as follows:

- An $[n, k]$ QLDPC code, having a coding rate of $R_Q = k/n$, may be constructed from a classical $(2n, n + k)$ binary LDPC code, having a coding rate of $R_c = (n + k)/2n$, if the associated PCM H satisfies the stringent symplectic criterion.
- Ideally, the rows of the PCM H should have at most a single overlapping value of 1 (or non-zero value in the GF(4) formalism)) for the sake of avoiding length-4 cycles in the Tanner graph, which degrade the performance of the iterative decoding algorithm. Unfortunately, the symplectic criterion requires 'even overlaps' between the rows of H, thus resulting in unavoidable length-4 cycles. A major design challenge is therefore to construct good QLDPC codes in the wake of the unavoidable length-4 cycles.
- We may exploit four main global structures of the PCM H for designing QLDPC codes, namely dual-containing CSS, non-dual-containing CSS, non-CSS and entanglement-assisted solutions of Figure 7.1. The design challenges associated with each of these structures are summarized below:
  - **Dual-containing CSS (Section 7.2.1):** Mackay's bicycle codes are so far the best amongst the dual-containing CSS codes, but their performance is not on par with the classical LDPC codes. This is because this construction suffers the most from having short cycles, which exist both in the binary as well as in the GF(4) formalism.
  - **Non-dual-containing CSS codes (Section 7.2.1):** It is difficult to find a pair of sparse binary PCMs satisfying the symplectic criterion, which constitute good QLDPC codes. At the time of writing, only the SC QC-LDPC codes and the non-binary QC-QLDPC codes are known to perform close to the Hashing bound. But this comes either at the cost of pre-shared noiseless ebits or at an increased complexity.
  - **Non-CSS codes (Section 7.2.2):** Ideally, non-CSS constructions are preferred over the CSS codes because they exploit the redundant qubits more efficiently. However, finding good non-CSS QLDPC codes remains an open challenge at the time of writing.
  - **EA codes (Section 7.2.3):** Entanglement-assistance may aid in achieving a performance comparable to that of the classical LDPC codes. However again, this requires pre-shared ebits, which constitute a valuable resource gleaned at the cost of a transmission overhead. Therefore, efforts must be made to minimize the number of required ebits.

- Additionally, it is desirable that the resulting QLDPC code has the following attributes:
  - A structured PCM, for example a cyclic or quasi-cyclic structure, for facilitating its implementation; and
  - An unbounded minimum distance or at least a sufficiently high minimum distance for long block lengths.

We further discussed in Section 7.3 that QLDPC codes may be decoded using syndrome-based BP either in the binary domain or in the non-binary domain. Besides the obvious lower complexity of the binary decoding, the two main differences between these decoding regimes are:

- In contrast to the binary decoding of Section 7.3.1, which assumes that the bit-flips and phase-flips are independent, non-binary decoding of Section 7.3.2 takes into account the correlation between them, which improves their performance.
- The number of length-4 cycles is higher in the non-binary formalism of the PCM as compared to the binary one. This tends to degrade the performance of the non-binary decoder.

Hence, we have a pair of conflicting attributes.

From the perspective of decoding, the challenges discussed in Section 7.3.3 may be summarized as follows:

- **Degeneracy:** Quantum codes are inherently degenerate in nature. This may improve the associated decoding performance if the decoder takes this degeneracy into account. Unfortunately, the BP algorithm does not exploit this degeneracy. In fact, since BP is based on marginalized probabilities, the presence of degenerate errors impairs its performance.
- **Short cycles:** Unavoidable length-4 cycles found in QLDPC codes degrade the performance of BP. This gets even worse for the homogeneous CSS codes, when they are decoded in the non-binary domain.

Heuristic methods, namely random perturbation and enhanced feedback, are known to mitigate both these issues to some extent. However, this is achieved at the cost of an increased decoding complexity.

In the spirit to construct non-dual-containing CSS QLDPC codes, we conceived a formalism in Section 7.4 for constructing high-rate row-circulant QC-QLDPC codes from arbitrary row-circulant classical LDPC matrices. Since our design is merely based on the transpose and column permutation operations, the characteristics of the underlying classical LDPC matrix are not compromised. In particular, we applied our formalism to the BIBD-based classical LDPCs for evaluating its performance in Section 7.5. It was demonstrated in Figure 7.10 that our designed $[2534, 2172]$ QLDPC code is capable of tolerating three times higher depolarizing probability at a WER of 10^{-4} as compared to its EA (homogeneous) counterpart and about four times higher depolarizing probability, when compared

Dec. No.	**Decoding Method**	p **at WER** $= 10^{-4}$	**Improvement in** p	$\mathrm{I}_{\mathbf{avg}}$
System I of Table 7.9 (Figure 7.14 and Figure 7.15)				
1	Binary	0.0085	-	1.7
2	Standard Non-Binary	0.01075	26% w.r.t. Dec. 1	2.6
3	Random Perturbation	0.014	30% w.r.t. Dec. 2	4.19
4	Enhanced Feedback	0.014	30% w.r.t. Dec. 2	3.6
5	Modified Non-Binary	0.014	30% w.r.t. Dec. 2	2.47
6	Modified & Enhanced Feedback	0.017	21% w.r.t. Dec. 5	3.4
System II of Table 7.12 (Figure 7.16 and Figure 7.17)				
1	Binary	0.057	-	4
2	Standard Non-Binary	0.069	21% w.r.t. Dec. 1	5
3	Random Perturbation	0.076	10% w.r.t. Dec. 2	7.5
4	Enhanced Feedback	0.082	19% w.r.t. Dec. 2	7
5	Modified Non-Binary	0.085	23% w.r.t. Dec. 2	5.5
6	Modified & Enhanced Feedback	0.0945	11% w.r.t. Dec. 5	7

Table 7.16: Summary of the simulation results of Section 7.8.1 at a WER of 10^{-4}.

to an equivalent dual-containing bicycle code in the GF(4) formalism. This is because both the homogeneous EA-QLDPC codes as well as the dual-containing QLDPCs have numerous short cycles in the GF(4) formalism, which impair the performance of the decoding algorithm.

In Section 7.6, we conceived a modified non-binary decoding algorithm for homogeneous CSS-type QLDPC codes, which successfully mitigated the problems imposed by unavoidable length-4 cycles. In Section 7.8.1, we demonstrated in Figure 7.14 to Figure 7.17 that our modified decoder exhibits a superior WER performance, despite its lower decoding complexity, when compared to the state-of-the-art decoding techniques. We also amalgamated our improved decoding algorithm with the heuristic methods for attaining additional performance gains. Results are summarized in Table 7.16.

In Section 7.7, we laid out the reweighted BP algorithm, which is known to alleviate the issue of short cycles in classical LDPC codes. In Section 7.8.2, we amalgamated our modified algorithm of Section 7.6 with the URW-BP of Section 7.7 for evaluating the performance. It was demonstrated in Figure 7.19 that URW-BP can be exploited for counteracting the issues of short-cycles, particularly when the maximum number of decoding iterations is small. More specifically, the results summarized in Table 7.15 reveal that URW-BP improves the tolerable depolarizing probability by 24% at a WER of 10^{-3}, when $\mathtt{I}_{\max} = 10$ decoding iterations are invoked, while the improvement decreases to 13% at $\mathtt{I}_{\max} = 20$. Increasing the maximum number of affordable decoding iterations to $\mathtt{I}_{\max} = 90$ reduces the performance gain to 3%.

Chapter 8

Conclusions and Future Directions

In this concluding chapter, we will summarize our conclusions in Section 8.1, while a range of potential future research directions will be discussed in Section 8.2.

8.1 Summary and Conclusions

Quantum Error Correction Codes (QECCs) are required for stabilizing and protecting fragile qubits against the undesirable effects of quantum decoherence. For classical information transmission over a quantum channel, we may alternatively exploit the family of classical error correction codes for counteracting the deleterious impact of decoherence. However, for realizing reliable quantum information transmission through a quantum-based communication system as well as for quantum computing systems, QECCs are indispensable. Against this background, in this thesis, we aimed to:

- Design classical error correction schemes for reliable transmission of classical information over an absolutely secure quantum channel;
- Design QECCs for absolutely secure quantum-based communication and quantum computation systems.

In light of these objectives, we progressed through this thesis as follows:

- **Chapter 1:** We commenced our discourse in Section 1.1 by laying out the motivation for designing error correction schemes for both quantum communication and computing systems, while the historical background of QECCs was presented in Section 1.2. We then proceeded with the outline of the thesis in Section 1.3. Finally, the novel contributions of the thesis were highlighted in Section 1.4.

- **Chapter 2:** In Chapter 2, we provided a preliminary introduction to quantum information theory. More specifically, we detailed the difference between classical bits and qubits in Section 2.2, while the notion of an N-qubit quantum system was presented in Section 2.3. We then introduced the no-cloning theorem and the notion of quantum entanglement in Sections 2.4 and 2.5, respectively. In Section 2.6, various quantum unitary operators were discussed, while the Pauli group was defined in Section 2.7. Finally, we discussed various quantum channel models in Section 2.8.

- **Chapter 3:** In Chapter 3, EXtrinsic Information Transfer (EXIT) Chart-aided near-capacity classical code designs were presented for reliable transmission of classical information over a secure quantum channel. In particular, we focused our attention on the entanglement-assisted transmission of classical information over a depolarizing channel, which was achieved with the aid of the SuperDense (SD) coding protocol.

  Explicitly, in Section 3.2, we briefly reviewed the SD protocol, which maps the classical bits onto qubits for transmission over a quantum channel. More specifically, the 2-qubit SD (2SD) protocol was presented in Section 3.2.1, which transmits 2 classical bits per channel use (cbits/use) with the aid of a single pre-shared entangled qubit. The 2SD protocol of Section 3.2.1 was further generalized to its N-qubit SD (NSD) counterpart in Section 3.2.2, which facilitates the transmission of N classical bits by sending only $(N-1)$ qubits over the noisy quantum channel, while one qubit is pre-shared with the receiver before actual transmission takes place. More explicitly, we considered the $N = 3$ scenario, resulting in a transmission rate of 1.5 cbits/use. The corresponding quantum circuits designed for the 2SD and 3SD protocols were given in Figure 3.2 and Figure 3.3, respectively. The associated Entanglement-Assisted Classical Capacity (EACC) was derived for the generalized NSD transmission in Section 3.3, which was specifically characterized for the 2SD and 3SD schemes in Eq. (3.17) and (3.19), respectively.

  Section 3.4 proposed a radically new amalgamated classical-quantum code structure, where a classical IRregular Convolutional Code (IRCC) was concatenated with a classical symbol-based recursive Unity Rate Code (URC) and a quantum-based SD mapper through a bit interleaver. More explicitly, an IRCC constitutes the outer code of the proposed code design, while the URC and SD scheme were combined to form a single inner component. The resultant system was referred to as a 'bit-based IRCC-URC-SD' arrangement, whose schematic was given in Figure 3.4. Furthermore, iterative decoding was invoked for exchanging extrinsic information between the inner (URC-SD) and outer (IRCC) decoders. We next presented our EXIT-chart aided near-capacity design criterion in Section 3.5 for optimizing the weighting coefficients of the IRCC. The design guidelines conceived for our IRCC-URC-SD structure are summarized as follows:

  - Using the bit-based capacity curves of Figure 3.5, the noise limit p^*, which may be defined as the maximum tolerable channel depolarizing probability, is determined for the desired classical information transmission rate in cbits/use and for the required SD protocol .

- The inner decoder EXIT curve of the amalgamated twin-component inner (URC-SD) decoder is computed at a channel depolarizing probability of $p = (p^* - \epsilon)$ using the binary EXIT chart generation approach of Section 3.5.1. Here ϵ is a small number, which characterizes the distance from the capacity p^*.
- Finally, as discussed in Section 3.5.2, the weighting coefficients of the IRCC subcodes, having an overall coding rate of R_o, are optimized for the sake of ensuring that a marginally open tunnel exists between the EXIT curves of the outer and inner decoder at the highest possible depolarizing probability, i.e at the lowest value of ϵ. This in turn guarantees that the system has a near-capacity performance.

The performance of the bit-based IRCC-URC-SD design invoked for our 2SD and 3SD schemes was evaluated in Section 3.6.1 and 3.6.2, respectively, which was benchmarked against the bit-based EACC of Figure 3.5. It was demonstrated that the actual convergence threshold determined using the Bit Error Rate (BER) performance curves of Figure 3.8 and Figure 3.12 was the same as the EXIT chart predictions of Figure 3.7 and Figure 3.11, respectively. Furthermore, the BER performance improved upon increasing the number of iterations, as long as the depolarizing probability was lower than the convergence threshold, which was quantified in Figure 3.9 and Figure 3.13 in terms of the distance from capacity (dB) at BER of 10^{-4}. However, the performance only improved with diminishing returns at a higher number of iterations. In particular, the convergence threshold of the designed systems of Figure 3.7 and Figure 3.11 was within 0.4 dB of the achievable noise limit for both 2SD as well as for 3SD schemes. More specifically, the normalized capacity loss was only 0.062 and 0.031 cbits/use from the corresponding 2-qubit and 3-qubit noise limits, respectively. The attainable performance was also benchmarked against that of classical turbo codes in Figure 3.14, which is summarized in Table 8.1.

The bit-based IRCC-URC-SD code structure of Figure 3.4 incurred a capacity loss due to the symbol-to-bit conversion, which was quantified in Figure 3.5. To alleviate this issue, a symbol-based CC-URC-SD scheme was conceived in Section 3.7, where a symbol-based classical Convolutional Code (CC) was concatenated with an integrated URC-SD through a symbol interleaver. The proposed CC-URC-SD scheme was optimized with the aid of non-binary EXIT charts. More specifically, we optimized the system in Section 3.8 by exhaustively searching through all the possible generator polynomials in Figure 3.16 for finding the specific CC whose EXIT curve yields a marginally open tunnel at the highest possible depolarizing probability. The resultant optimal memory-2, memory-3 and memory-4 CCs of Figure 3.20 were found to have the octally represented generator polynomials of $(g_1, g_2) = (7, 5)_8$, $(g_1, g_2) = (17, 15)_8$ and $(g_1, g_2) = (31, 36)_8$, respectively. The simulation results of Figure 3.19 demonstrated that the symbol-based CC-URC-2SD provides a significant BER performance improvement, despite its lower encoding/decoding complexity than that of the bit-based IRCC-URC-2SD. Quantitatively, we observed in Figure 3.19 that after 2 iterations our proposed symbol-based CC-URC-2SD design incorporating a memory-4 CC outperformed the bit-based IRCC-URC-2SD scheme of Figure 3.4 by 3.7 dB at a BER of 10^{-4}. Even the memory-2 and memory-3 designs were found

Code Structure	$\mathtt{I}_{\infty}$	**Distance from Capacity at BER** $= 10^{-4}$	**Reference Figures**
IRCC-URC-2SD	32	0.6 dB	Figure 3.8 and Figure 3.9
IRCC-URC-3SD	32	0.7 dB	Figure 3.12 and Figure 3.13
TC-2SD	16	1.9 dB	Figure 3.14
TC-3SD	16	2.2 dB	Figure 3.14

Table 8.1: Comparison of the decoding complexity and the achievable performance of the TC-SD and IRCC-URC-SD schemes. Decoding complexity is quantified in terms of the number of iterations required for achieving a near-perfect convergence, which is denoted as $\mathtt{I}_{\infty}$, while the achievable performance is measured in terms of the distance from the bit-based capacity at a BER of 10^{-4}.

to outperform the bit-based IRCC-URC-2SD in Figure 3.21. Furthermore, the performance of the bit-based IRCC-URC-2SD and the symbol-based CC-URC-2SD was comparable, once perfect convergence was achieved. Nonetheless, the symbol-based designs imposed a lower decoding complexity, because they require less iterations, as quantified in Table 8.2.

- **Chapter 4:** In Chapter 3, we designed near-capacity classical-quantum coding schemes, which exploited redundancy in the classical domain. Therefore, the designs given in Chapter 3 are only suitable for a quantum-based communication system, which transmits classical information. By contrast, we have to resort to QECCs for the transmission of quantum information and for quantum computing systems. In this spirit, the rest of the thesis is focused on the design of QECCs.

  Unlike a classical bit, a qubit cannot be copied and it collapses to a classical bit upon measurement. Furthermore, while bit errors are the only type of errors experienced during transmission over a classical channel, a quantum channel may inflict both bit-flips as well as phase-flips. Therefore, it is not feasible to directly map classical codes onto their quantum counterparts. Nevertheless, quantum codes may be designed from the known classical codes by exploiting the underlying quantum-to-classical isomorphism, which This transition from the classical to the quantum domain was presented in Chapter 4. The discussion commenced in Section 4.2, where the classical linear block codes were reviewed with a particular emphasis on the 3-bit repetition code. The stabilizer formalism, which facilitates the design of quantum codes from the classical ones, was then discussed in Section 4.3. It was demonstrated with the aid of the 3-qubit bit-flip repetition code and 3-qubit phase-flip repetition code that:

Code Structure	$\mathtt{I}_\infty$	**Distance from Capacity at BER** $= 10^{-4}$	**Reference Figures**
IRCC-URC-2SD	32	1.2 dB	Figure 3.8 and Figure 3.9
CC(2,1,2)-URC-2SD	20	1 dB	Figure 3.17 and Figure 3.18
CC(2,1,3)-URC-2SD	20	1 dB	Figure 3.21
CC(2,1,4)-URC-2SD	20	1 dB	Figure 3.21

Table 8.2: Comparison of the decoding complexity and the achievable performance of the bit-based IRCC-URC-2SD and the symbol-based CC-URC-2SD schemes. The decoding complexity is quantified in terms of the number of iterations required for achieving a near-perfect convergence, which is denoted as $\mathtt{I}_\infty$, while the achievable performance is measured in terms of the distance from the symbol-based capacity at a BER of 10^{-4}.

- The copying operation of classical codes is equivalent to copying the basis states of the qubit, which can be achieved via quantum entanglement;
- Measurement of a qubit may be circumvented by observing the channel errors without observing the actual quantum information by invoking the classical syndrome decoding techniques;
- Phase-flips may be corrected by copying in the Hadamard basis.

This quantum to classical transition was summarized in Figure 4.6.

Section 4.3.2 detailed the equivalence between the quantum and classical Parity Check Matrices (PCMs), focusing specifically on the Pauli-to-binary isomorphism in Section 4.3.2.1, while the Pauli-to-quaternary isomorphism was discussed in Section 4.3.2.2. The underlying mapping was also summarized in Table 4.11, which facilitates the design of quantum codes from arbitrary classical binary codes, if they meet the symplectic criterion, and from arbitrary classical quaternary codes, if they satisfy the Hermitian inner product. This in turn led to various structures of Quantum Stabilizer Codes (QSCs), namely to the dual-containing Calderbank-Shor-Steane (CSS) as well as to the non-dual-containing CSS and non-CSS structures, which are summarized in Table 8.3. Furthermore, it was discussed in Section 4.3.3 that QSCs are degenerate in nature. Consequently, errors, which only differ by the generator of the stabilizer group have the same impact on the transmitted qubit and can be corrected by the same recovery operation. This further leads to the class of harmless undetected error patterns in Figure 4.9, which only exist in the quantum domain.

Code Type	**Parity Check Matrix**	**Criteria**	**Design Examples**
Dual-containing CSS	$\left(\begin{array}{ccc} H'_z & \vert & \mathbf{0} \\ \mathbf{0} & \vert & H'_z \end{array}\right)$	$H'_z H_z'^T = 0$	Steane code of Eq. (4.36) and QCC of Eq. (4.65)
Non-dual-containing CSS	$\left(\begin{array}{ccc} H'_z & \vert & \mathbf{0} \\ \mathbf{0} & \vert & H'_x \end{array}\right)$	$H'_z \neq H'_x$ and $H'_z H_x'^T = 0$	-
Non-CSS	$(H_z \vert H_x)$	$H_z H_x^T + H_x H_z^T = 0$	Hamming code of Eq. (4.51) and QCC of Eq. (4.68)
EA	$\left(\begin{array}{ccc} H'_z & \vert & \mathbf{0} \\ \mathbf{0} & \vert & H'_z \end{array}\right)$ and $(H_z \vert H_x)$	None	EA-QSC of Eq. (4.77)

Table 8.3: Summary of the stabilizer code structures of Chapter 4.

Section 4.4 extended the discussion of Section 4.3.2 to the family of convolutional codes, where both CSS-type and non-CSS type Quantum Convolutional Codes (QCCs) were designed from the known classical CCs. Section 4.5 then presented the Entanglement-Assisted (EA) stabilizer formalism. This formalism allows any classical code to be used as a quantum code, even if it does not meet the symplectic or Hermitian inner product criterion. However, this is achieved with the aid of pre-shared entanglement, which is indeed a valuable resource gleaned at the cost of a transmission overhead. Therefore, efforts must be made to minimize the number of required entangled qubits.

The various stabilizer code structures discussed in this chapter are summarized in Table 8.3.

- **Chapter 5:** The stabilizer codes of Chapter 4 invoke the classical syndrome decoding, which runs over the equivalent classical code derived on the basis of the underlying quantum-to-binary or quantum-to-quaternary isomorphism. Therefore, in Chapter 5, we focused our efforts on the classical syndrome decoding techniques.

  The notion of syndrome decoding derives its essence from the Look-Up Table (LUT) based syndrome decoding of classical linear block codes, which was discussed in Section 5.2. More explicitly, in contrast to the conventionally used codeword decoding, which aims for finding the most likely codeword, syndrome decoding aims for identifying the most likely channel error vector. It was demonstrated in Section 5.2 that in the context of linear block codes, LUT-based syndrome decoding substantially reduces the storage requirements of the standard array-based codeword decoding. In Section 5.3, the concept of syndrome decoding was extended to trellis-based decoding. In particular, Section 5.3.1 discussed the construction of syndrome-based trellis of linear block codes, while Section 5.3.2 presented the syndrome-based trellis of convolutional codes. It was pointed out that since every path in the error trellis of Figure 5.6 corresponds to a path in the codeword trellis of Figure 5.7, these representations are exactly equivalent. Furthermore, each path of the error trellis is a legitimate error sequence for a given observed syndrome, while each path of a codeword trellis is a valid codeword. Consequently, the former is used for syndrome decoding, while the latter is exploited for codeword decoding.

The error trellis-based syndrome decoding is of particular significance, because the state probabilities of an error trellis depend on the channel errors, rather than on the coded sequence. Consequently, in the low noise regime, the syndrome decoder is more likely to encounter a zero-state due to having predominantly error-free transmissions. This unique feature of error trellis-based syndrome decoding led to the discussion on Block Syndrome Decoding (BSD) in Section 5.4. Section 5.4.1 presented the general BSD formalism, while Section 5.4.2 conceived a reduced-complexity BSD for Turbo Trellis Coded Modulation (TTCM), hence referred to as 'BSD-TTCM', whose schematic was given in Figure 5.10. The proposed BSD-TTCM divides the received frame into error-free and erroneous sub-blocks based on the syndrome of the received frame. Only the erroneous blocks are subsequently decoded, thereby reducing the decoding complexity. This yields a reduction in the decoding complexity in the higher Signal-to-Noise Ratio (SNR) region as well as for higher indexed decoding iterations. Furthermore, the performance of BSD-TTCM of Figure 5.10 depends on the design parameter L_{min}, which is the minimum number of consecutive zero syndromes after which the sub-block is deemed to be error-free. Section 5.5 evaluated the performance of the BSD-TTCM of Figure 5.10 for transmission over a classical Additive White Gaussian Noise (AWGN) channel as well as an uncorrelated Rayleigh fading channel. The results of Figure 5.12 to Figure 5.17 obtained for the TTCM of Table 5.1 are summarized in Table 8.4. Here, the design parameter L_{min} was heuristically optimized for ensuring that BSD-TTCM yields the same BER performance as the conventional full-complexity decoder TTCM decoder, as evidenced in Figure 5.12 and Figure 5.15 for transmission over an AWGN and a Rayleigh fading channel, respectively. Section 5.5.3 further evaluated the performance of BSD-TTCM for a short frame of only 500 symbols. The BSD-TTCM scheme was found to outperform the frequently used high-SNR early termination technique in Figure 5.20, despite the short frame length.

- **Chapter 6:** Based on the insights developed in Chapter 4 as well as Chapter 5 and inspired by the EXIT-chart aided near-capacity classical code designs of Chapter 3, we conceived Hashing bound approaching concatenated QECCs using EXIT charts in Chapter 6.

  Section 6.2 presented our code design objectives, where the achievable capacity region, or more precisely the Hashing region, was characterized in Figure 6.1, which depends on the coding rate, on the channel depolarizing probability as well as on the entanglement consumption rate. We then proceeded with the circuit based representation of QCCs in Section 6.3, which facilitates the degenerate iterative decoding of concatenated quantum codes. We also discussed the construction of the Clifford unitary encoder, which is completely specified by the Hadamard as well as by the phase and the controlled-NOT gates of Eq. (6.12). This was further explained by constructing the equivalent binary Clifford encoder for a 3-qubit bit-flip repetition code using

Channel	Percentage of No-Decoding	Equivalent No.of Iterations	Comparison with Early Termination
AWGN	At least 20% and 45% reduction in the 5th and 6th iterations, respectively.	At least 17% deduction.	At least a reduction of 0.5 iteration.
Rayleigh	At least 20% and 30% reduction in the 5th and 6th iterations, respectively.	At least 12% deduction.	At least a reduction of 0.5 iteration.

Table 8.4: Decoding complexity reduction of BSD-TTCM (summarized from the simulation results of Figure 5.12 to Figure 5.17, which were obtained for the TTCM of Table 5.1). Decoding complexity is quantified in terms of the percentage of no-decoding (second column) and the equivalent number of iterations (third column). Complexity reduction is further compared with high-SNR early termination technique (fourth column).

the encoding circuit of Figure 6.4. Finally, Section 6.4 detailed the structure and decoding of concatenated quantum codes, which may also be referred to as Quantum Turbo Codes (QTCs). More specifically, the schematic of a quantum communication system relying on a pair of concatenated QSCs was presented in Section 6.4.1, while the associated degenerate iterative decoding was detailed in Section 6.4.2. In particular, it was highlighted in Section 6.4.2 that unlike the conventional Maximum *A-Posteriori* (MAP) decoder, which yields the most likely error for a given syndrome, a degenerate MAP decoder aims for finding the most likely error coset by summing the probabilities over the set of degenerate errors (Eq. (6.49)).

Section 6.5 conceived the EXIT charts of the concatenated quantum code structure of Figure 6.6. More specifically, the classical non-binary EXIT chart generation technique was extended to the circuit-based syndrome decoder of QTCs in Figure 6.9 and Figure 6.10 for the inner and outer decoders, respectively. While the classical EXIT charts aim for modeling the *a-priori* information concerning the input bits of the inner encoder (and similarly the output bits of the outer encoder), the EXIT chart conceived for quantum codes models the *a-priori* information concerning the corresponding error-sequence, i.e. the error-sequence related to the input qubits of the inner encoder (and similarly the error-sequence related to the output qubits of the outer encoder). Section 6.6 then evaluated the accuracy of the EXIT charts of Section 6.5. It was demonstrated in Section 6.6.1 that the convergence threshold predicted by the EXIT-chart of Figure 6.12 is the same as the one determined using the Monte-Carlo simulation results of Figure 6.13. In Section 6.6.2, we analyzed the inner decoder EXIT curves of recursive (unassisted) and non-recursive (entanglement-assisted) QCCs in Figure 6.14 for demonstrating that the EXIT curve of a recursive inner QCC reaches the $(1, 1)$-point of perfect convergence. This in turn facilitates the design of families of QTCs having an unbounded minimum distance. Section 6.6.3 presented the inner and outer components of our optimized QTC in Figure 6.15, which were found using an EXIT-chart aided exhaustive search. Our design guidelines for achieving a Hashing bound approaching performance may be summarized as follows:

- **Design Criterion:** Determine the noise limit p^* for the desired code parameters, i.e the coding rate and the entanglement consumption rate of the resultant QTC.
- **Selection of Inner Component:** Select a recursive and non-catastrophic inner EA-QCC, which can be either derived from the family of known classical codes, as discussed in Chapter 4, or it can be constructed using random Clifford operations, which were discussed in Section 6.3. At this point, the EXIT chart technique investigated in Section 6.5 is invoked for the sake of finding that specific QCC, which yields the largest area under its inner decoder EXIT-curve at the noise limit p^*.
- **Selection of Outer Component:** Find a non-catastrophic outer QCC, whose EXIT curve gives the best EXIT-curve match with the inner decoder EXIT curve of the chosen inner code. Our EXIT-chart aided design of Section 6.5 aims for creating a narrow, but marginally open tunnel between the EXIT curves of the inner and outer decoders at the highest possible depolarizing probability, as demonstrated in Figure 6.15. The narrower the tunnel-area, the lower is the deviation from the Hashing bound, which is quantified using Eq. (6.2).

Based on the EXIT chart of Figure 6.15 and the achievable performance of Figure 6.16, it was demonstrated in Section 6.6.3 that the convergence threshold of our optimized QTC is 0.6 dB closer to the Hashing limit as compared to the distance spectra based QTCs of [88]. However, this is achieved at the cost of a higher Word Error Rate (WER) floor, which may be reduced upon increasing the interleaver length, as evidenced in Figure 6.16, hence imposing a longer delay.

Section 6.7 proposed the structure of a 10-subcode Quantum IRregular Convolutional Code (QIRCC) for further facilitating the Hashing bound approaching code design. The proposed QIRCC may be dynamically adapted to match any given inner code using EXIT charts, hence allowing us to dispense with the exhaustive code search. Section 6.8 quantified the impact of using QIRCC as the outer component in Figure 6.6, which was recorded in Figure 6.21. As evidenced in Figure 6.22, the QIRCC-based optimized design, when used in conjunction with the same inner code as that of the QTC in [88], outperformed the QTC of [88] both in terms of the convergence threshold as well as the achievable WER performance. The results are also summarized in Table 8.5. However, this was achieved at the cost of an increased decoding complexity, because the former invoked a maximum of $\mathtt{I}_{\max} = 15$ iterations in contrast to the $\mathtt{I}_{\max} = 8$ iterations invoked by the latter. The QIRCC-based optimized design of Figure 6.20 also had a lower WER than the exhaustive-search based optimized design of Section 6.6.3, as demonstrated in Figure 6.22.

- **Chapter 7:** Pursuing further the design of iterative code structures, Chapter 7 focused on

Code Structure	**Distance from Capacity**	
	Convergence Threshold	**Performance at WER** $= 10^{-3}$
QTC of [88]	0.9 dB	1.15 dB
QIRCC-based QTC	0.4 dB	0.7 dB

Table 8.5: Comparison of the performance of the QTC of [88] with our QIRCC-based QTC, when the same inner code was used.

Code Type	**Merits**	**Demerits**
Dual-containing CSS	Mackay's bicycle codes are so far the best among the QLDPC codes.	Numerous short cycles.
Non-dual-containing CSS	Few short cycles in GF(4) formalism and none in the binary formalism.	Difficult to find good non-dual-containing CSS QLDC codes.
Non-CSS	Efficiently exploit redundant qubits.	Difficult to find good non-CSS QLDC codes.
EA	Performance comparable to the classical LDPC codes.	Pre-shared entangled qubits required.

Table 8.6: Summary of the QLDPC constructions of Chapter 7.

Quantum Low Density Parity Check (QLDPC) codes, by providing insights into the various QLDPC structures as well as the associated decoding.

In line with the categorization of Table 8.3, Section 7.2 reviewed the QLDPC code construction methods, focusing in particular on the related design issues. A major design challenge highlighted in this discussion was the presence of unavoidable length-4 cycles, resulting from the commutativity requirement of the stabilizers. More explicitly, an arbitrary binary or quaternary code may be used for constructing a QLDPC matrix, if it meets the symplectic product or the Hermitian inner product criterion, as also summarized in Table 8.3. This in turn results in short (length-4) cycles. We summarize the design challenges of Section 7.2 in Table 8.6. Section 7.3 discussed the syndrome decoding of QLDPC codes. In particular, binary decoding was presented in Section 7.3.1, while its non-binary counterpart was the subject of Section 7.3.2. The differences highlighted between the two decoding schemes may be summarized as follows:

– In contrast to the binary decoding of Section 7.3.1, which assumes that bit and phase errors are independent, the non-binary decoding of Section 7.3.2 takes into account the correlation between them, which improves their performance.

- The number of length-4 cycles is higher in the non-binary formalism of the PCM as compared to the binary formalism. This tends to degrade the performance of the non-binary decoder.

Section 7.3.3 presented the decoding issues associated with QLDPC codes, which may be summarized as:

- **Degeneracy:** Quantum codes are inherently degenerate in nature. This may improve the associated decoding performance, provided that the decoder takes this degeneracy into account. Unfortunately, the Belief Propagation (BP) algorithm does not exploit this degeneracy. In fact, since BP is based on marginalized probabilities, the presence of degenerate errors impairs its performance.
- **Short cycles:** Unavoidable length-4 cycles found in QLDPC codes deteriorate the performance of BP. This gets even worse for the homogeneous CSS codes, when they are decoded in the non-binary domain.

Heuristic methods, namely random perturbation and enhanced feedback, were then invoked for alleviating these issues at the cost of an increased decoding complexity.

Section 7.4 conceived a formalism for constructing high-rate row-circulant QC-QLDPC codes from arbitrary row-circulant classical LDPC matrices. Since the proposed design is merely based on the transpose and column permutation operations, the characteristics of the underlying classical LDPC matrix are not compromised. In particular, the new formalism was applied to the family of BIBD-based classical LDPCs for evaluating their performance in Section 7.5. It was demonstrated in Figure 7.10 that the proposed construction outperforms the comparable bicycle code as well as the EA-QLDPC code. Quantitatively, our designed $[2534, 2172]$ QLDPC code of Table 7.5 is capable of tolerating three times higher depolarizing probability at a WER of 10^{-4} as compared to its EA counterpart and about four times higher depolarizing probability, when compared to an equivalent dual-containing bicycle code.

Section 7.6, we proposed a modified non-binary decoding algorithm for homogeneous CSS-type QLDPC codes, which successfully mitigated the issue of unavoidable length-4 cycles. In Section 7.8.1, it was demonstrated in Figure 7.14 to Figure 7.17 that the modified decoder exhibits a superior WER performance as well as a lower decoding complexity than the state-of-the-art benchmark decoding techniques. The improved decoding algorithm of Section 7.6 was also amalgamated with heuristic methods for attaining additional performance gains. The simulation results of Figure 7.14 to Figure 7.17 are summarized in Table 8.7.

Section 7.7 presented the Uniform ReWeighted BP (URW-BP) algorithm, which is known to alleviate the issue of short cycles in classical LDPC codes. In Section 7.8.2, we also amalgamated our modified algorithm of Section 7.6 with the URW-BP of Section 7.7. It was demonstrated

System	Dec. No.	Decoding Method	Improvement in p	$\mathtt{I}_{\mathbf{avg}}$	Reference
System I	1	Binary	-	1.7	Table 7.9, Figure 7.14, Figure 7.15
	2	Standard Non-Binary	26% w.r.t. Dec. 1	2.6	
	3	Random Perturbation	30% w.r.t. Dec. 2	4.19	
	4	Enhanced Feedback	30% w.r.t. Dec. 2	3.6	
	5	Modified Non-Binary	30% w.r.t. Dec. 2	2.47	
	6	Modified & Enhanced Feedback	21% w.r.t. Dec. 5	3.4	
System II	1	Binary	-	4	Table 7.12, Figure 7.16, Figure 7.17
	2	Standard Non-Binary	21% w.r.t. Dec. 1	5	
	3	Random Perturbation	10% w.r.t. Dec. 2	7.5	
	4	Enhanced Feedback	19% w.r.t. Dec. 2	7	
	5	Modified Non-Binary	23% w.r.t. Dec. 2	5.5	
	6	Modified & Enhanced Feedback	11% w.r.t. Dec. 5	7	

Table 8.7: Summary of the simulation results of Section 7.8.1, where performance is quantified for the dual-containing QLDPC (System I) and the homogeneous EA-QLDPC (System II) in terms of the achievable channel depolarizing probability p at a WER of 10^{-4} and the corresponding average number of decoding iterations $\mathtt{I}_{\text{avg}}$.

in Figure 7.19 that the URW-BP can be exploited for counteracting the problems imposed by short-cycles, particularly when the maximum number of decoding iterations is small.

8.2 Future Research Directions

In this section, we briefly discuss a number of possible future research avenues.

1. **Symbol-based IRCC-URC-SD for Entanglement-Assisted Classical Communication**

   In Chapter 3, we conceived bit-based IRCC-URC-SD schemes for approaching the bit-based EACC, while a CC-URC-SD arrangement was proposed for the sake of approaching the symbol-based EACC. More specifically, the symbol-based CC-URC-SD scheme outperformed the bit-based IRCC-URC-SD because the symbol-to-bit conversion invoked in the bit-based arrangement incurs an irrecoverable capacity loss. However, we exhaustively searched through all the possible convolutional codes for designing a near-capacity CC-URC-SD code. To dispense with the exhaustive search, it may be helpful to conceive a symbol-based IRCC, whose weighting coefficients can be dynamically adapted to provide the best EXIT-curve match with a given inner code.

2. **Near-Capacity Code Designs for Classical Communication Assisted by Less Noisy Qubits or by Noisy Qubits**

   In Chapter 3, we focused our attention on the code designs conceived for EA classical communication over a quantum depolarizing channel, assuming that the pre-shared entangled qubits are transmitted over a noiseless channel. Alternatively, we may assume that:

   - the ebits are *less* noisy [186], i.e. they only experience phase errors, which generally dominate the bit errors in realistic scenarios; or alternatively
   - the ebits are also shared over a noisy channel [112].

   From an implementational perspective, it might be useful to extend our proposed code designs to these two scenarios.

3. **Reduced-Complexity Block Syndrome Decoding for Quantum Turbo Codes**

   In Chapter 5, we conceived a reduced-complexity BSD for the classical TTCM, whose performance was analyzed for transmission over the classical AWGN and Rayleigh fading channels. Furthermore, we proposed Hashing bound approaching concatenated quantum codes, or more precisely QTCs, in Chapter 6. It will be beneficial to apply the concept of BSD to the decoding of QTCs (BSD-QTC). In TTCM decoding, the trellis of erroneous sub-blocks emerges from and terminates at the all-zero state. By contrast, a QTC is decoded over the equivalent circuit-based representation. Hence, in the context of the BSD-QTC, it will also be important to investigate how to initialize and terminate the memory states of the circuit-based representation.

4. **EXIT-Chart aided Design of Classically-Enhanced EA Quantum Turbo Codes**

   Our Hashing bound approaching QTCs of Chapter 6 only encode qubits. By contrast, classically-enhanced EA-QTCs [88, 202] support the simultaneous transmission of both classical as well as quantum information. In this context, it may be important to invoke the EXIT-chart aided design methodology for optimizing the constituent components of a classically enhanced EA-QTC, which aim for approaching the corresponding capacity limits/bounds of [203, 204].

5. **Union Bounds for Quantum Turbo Codes**

   The convergence threshold of our optimized QTC design of Section 6.6.3 is only 0.3 dB away from the achievable noise limit. However, it exhibits a high error floor in Figure 6.16. We managed to reduce the error floor by employing QIRCC as the outer component, as seen in Figure 6.17. Alternatively, it may be helpful to derive the QuBit Error Rate (QBER) union bound of the QTC based on its distance spectrum. We may then optimize the design with the aid of EXIT charts as well as the union bounds, where EXIT charts would ensure a Hashing bound approaching performance, while considering the union bounds would ensure a low error floor [115].

6. **Unity Rate Code or Short Block Code Aided Concatenated Quantum Codes**

   In Chapter 3, we demonstrated the beneficial impact of classical URC in achieving a near-capacity performance, when used as the inner code. By contrast, in Chapter 6, we used EA-QCCs as the inner components, which in turn yielded low-rate concatenated quantum codes. Alternatively, we may replace the inner EA-QCC by a Quantum Unity Rate Code (QURC), which would merely act as a scrambler. In this context, the design of a QURC is a promising research area. Since unassisted QCCs cannot be simultaneously recursive and non-catastrophic, the design of a QURC seems to be a formidable task. As an alternative option, we conjecture that if Quantum Short Block Codes (QSBCs) are conceived analogous to the classical Short Block Codes (SBCs) of [205, 206], which have a minimum distance of at least two, then they can be used as the inner component for reaching the $(1, 1)$ point of perfect convergence.

7. **Hashing Bound Approaching Quantum Low Density Parity Check Codes**

   The domain of QLDPC codes has a lot of research potential, both from the perspective of code design as well as decoder design. As discussed in Chapter 7, only the spatially coupled QLDPC codes have managed to approach the Hashing bound, which is achieved at the cost of an increased complexity or pre-shared noiseless qubits. Therefore, designing Hashing bound approaching QLDPC codes is still an open challenge. Furthermore, unlike the degenerate iterative decoding of QTCs, the decoding of QLDPC codes does not take into account the degeneracy issue. It is worth conceiving a degenerate decoding algorithm for QLDPC codes, or at least conceiving deterministic methods for improving the QLDPC decoding, rather than the heuristic methods of Section 7.3.3.

Appendix A

Construction of Syndrome Former

In this appendix, we detail the construction of the syndrome former, which was invoked in Chapter 5 for constructing the syndrome-based trellis. More specifically, in Section A.1, we derive the syndrome former of the Convolutional Code (CC) of Section 4.4, while in Section A.2 we construct the syndrome former of the Turbo Trellis Coded Modulation (TTCM) of Section 5.4.2.

A.1 Convolutional Codes

Recall from Section 4.4 that convolutional codes are equivalent to linear block codes having a semi-infinite length. Therefore, the generator polynomials of a convolutional code CC(n, k, m) may be arranged to construct the corresponding generator matrix G, as depicted in Eq. (4.60). For the special class of systematic CCs, whose n-bit codewords consist of k information bits followed by $(n-k)$ parity bits, the matrix G of Eq. (4.60) may also be represented as follows[1]:

$$G = \begin{pmatrix} \mathbf{IP}_0 & \mathbf{0P}_1 & \mathbf{0P}_2 & \dots & \mathbf{0P}_m & & & \\ & \mathbf{IP}_0 & \mathbf{0P}_1 & \mathbf{0P}_2 & \dots & \mathbf{0P}_m & & \\ & & \mathbf{IP}_0 & \mathbf{0P}_1 & \mathbf{0P}_2 & \dots & \mathbf{0P}_m \\ & & \ddots & & & \dots & \ddots & \end{pmatrix}, \tag{A.1}$$

where $\mathbf{I}$ is a $(k \times k)$-element identity matrix, $\mathbf{0}$ is a $(k \times k)$-element all-zero matrix and $\mathbf{P}_l$ is a $k \times (n-k)$-element matrix with entries:

$$\mathbf{P}_l = \begin{pmatrix} g_{1,l}^{(k)} & g_{1,l}^{(k+1)} & \dots & g_{1,l}^{(n-1)} \\ g_{2,l}^{(k)} & g_{2,l}^{(k+1)} & \dots & g_{2,l}^{(n-1)} \\ \vdots & \vdots & & \vdots \\ g_{k,l}^{(k)} & g_{k,l}^{(k+1)} & \dots & g_{k,l}^{(n-1)} \end{pmatrix}. \tag{A.2}$$

[1] Blank spaces in the matrix indicate zeros.

If $\mathbf{g}_i^{(j)} = (g_{i,0}^{(j)}, g_{i,1}^{(j)}, \ldots, g_{i,m}^{(j)})$ represents the generator polynomial for the ith input (information) bit and the jth output (coded) bit, then the resultant $(k \times n)$-element transform-domain matrix $G(D)$ is given by:

$$G(D) = \begin{pmatrix} 1 & 0 & \ldots & 0 & \mathbf{g}_1^{(k)}(D) & \mathbf{g}_1^{(k+1)}(D) & \ldots & \mathbf{g}_1^{(n-1)}(D) \\ 0 & 1 & \ldots & 0 & \mathbf{g}_2^{(k)}(D) & \mathbf{g}_2^{(k+1)}(D) & \ldots & \mathbf{g}_2^{(n-1)}(D) \\ \vdots & \vdots & & \vdots & \vdots & \vdots & & \vdots \\ 0 & 0 & \ldots & 1 & \mathbf{g}_k^{(k)}(D) & \mathbf{g}_k^{(k+1)}(D) & \ldots & \mathbf{g}_k^{(n-1)}(D) \end{pmatrix}, \tag{A.3}$$

where $\mathbf{g}_i^{(j)}(D) = g_{i,0}^{(j)} + g_{i,1}^{(j)}D + g_{i,2}^{(j)}D^2 + \cdots + g_{i,m}^{(j)}D^m$, given that D represents the delay of a single memory element (or register). Furthermore, the associated Parity Check Matrix (PCM) H of Eq. (4.62) takes the following form:

$$H = \begin{pmatrix} \mathbf{P}_0^T & \mathbf{I} & & & & & & & & & & \\ \mathbf{P}_1^T & \mathbf{0} & \mathbf{P}_0^T & \mathbf{I} & & & & & & & & \\ \mathbf{P}_2^T & \mathbf{0} & \mathbf{P}_1^T & \mathbf{0} & \mathbf{P}_0^T & \mathbf{I} & & & & & & \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & & & & & & \\ \mathbf{P}_m^T & \mathbf{0} & \mathbf{P}_{m-1}^T & \mathbf{0} & \mathbf{P}_{m-2}^T & \mathbf{0} & \ldots & \mathbf{P}_0^T & \mathbf{I} & & & \\ & & \mathbf{P}_m^T & \mathbf{0} & \mathbf{P}_{m-1}^T & \mathbf{0} & \mathbf{P}_{m-2}^T & \mathbf{0} & \ldots & \mathbf{P}_0^T & \mathbf{I} \\ & & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & & \vdots & \vdots \end{pmatrix}, \tag{A.4}$$

where $\mathbf{I}$ is an $(n-k) \times (n-k)$-element identity matrix, while $\mathbf{0}$ is an $(n-k) \times (n-k)$-element all-zero matrix. The corresponding $(n-k) \times n$ transform-domain PCM $H(D)$ is given by:

$$H(D) = \begin{pmatrix} \mathbf{g}_1^{(k)}(D) & \mathbf{g}_2^{(k)}(D) & \ldots & \mathbf{g}_k^{(k)}(D) & 1 & 0 & \ldots & 0 \\ \mathbf{g}_1^{(k+1)}(D) & \mathbf{g}_2^{(k+1)}(D) & \ldots & \mathbf{g}_k^{(k+1)}(D) & 0 & 1 & \ldots & 0 \\ \vdots & \vdots & & \vdots & \vdots & \vdots & & \vdots \\ \mathbf{g}_1^{(n-1)}(D) & \mathbf{g}_2^{(n-1)}(D) & \ldots & \mathbf{g}_k^{(n-1)}(D) & 0 & 0 & \ldots & 1 \end{pmatrix}. \tag{A.5}$$

Based on the above discussion, we next construct the syndrome former given in Eq. (5.14) for the generator matrix of Eq. (5.13), i.e. we have:

$$G(D) = \begin{pmatrix} 1+D & D & 1+D \\ D & 1 & 1 \end{pmatrix}. \tag{A.6}$$

We first find the equivalent systematic form analogous to Eq. (A.3) by applying elementary row operations to Eq. (A.6) as follows:

- **Divide Row 1 by** $(1+D)$

$$G(D) = \begin{pmatrix} 1 & D/(1+D) & 1 \\ D & 1 & 1 \end{pmatrix}. \tag{A.7}$$

- **Add** $(D\times$ **Row 1**$)$ **to Row 2**

$$G(D) = \begin{pmatrix} 1 & D/(1+D) & 1 \\ 0 & (1+D+D^2)/(1+D) & (1+D) \end{pmatrix}. \tag{A.8}$$

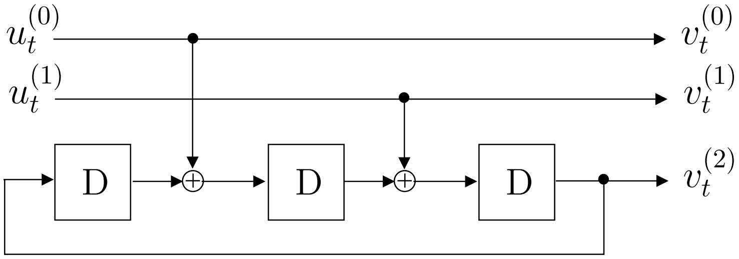


Figure A.1: Encoder of Ungerboeck's 8-state TCM-8PSK.

- **Multiply Row 2 by** $(1 + D)/(1 + D + D^2)$

$$G(D) = \begin{pmatrix} 1 & D/(1+D) & 1 \\ 0 & 1 & (1+D^2)/(1+D+D^2) \end{pmatrix}. \tag{A.9}$$

- **Add** $(D/1 + D \times$ **Row 2**$)$ **to Row 1**

$$G(D) = \begin{pmatrix} 1 & 0 & 1/(1+D+D^2) \\ 0 & 1 & (1+D^2)/(1+D+D^2) \end{pmatrix}. \tag{A.10}$$

Hence, Eq. (A.10) is the systematic form of the generator matrix given in Eq. (A.6). Based on this systematic matrix, we may deduce from Eq. (A.5) that the corresponding PCM $H(D)$ is as follows:

$$H(D) = \begin{pmatrix} 1/(1+D+D^2) & (1+D^2)/(1+D+D^2) & 1 \end{pmatrix}. \tag{A.11}$$

The equivalent non-systematic form of Eq. (A.11) can be derived by applying elementary row operations to Eq. (A.11). More specifically, multiply $H(D)$ with $1/(1 + D + D^2)$, which yields:

$$H(D) = \begin{pmatrix} 1 & 1+D^2 & 1+D+D^2 \end{pmatrix}. \tag{A.12}$$

A.2 Turbo Trellis Coded Modulation

In Section 5.4.2, we have employed Ungerboeck's 8-state TCM-8PSK [146], given in Figure A.1, for evaluating the performance of our proposed reduced-complexity decoder. The associated generator matrix is given by:

$$G(D) = \begin{pmatrix} 1 & 0 & D^2/(1+D^3) \\ 0 & 1 & D/(1+D^3) \end{pmatrix}. \tag{A.13}$$

The corresponding PCM is:

$$H(D) = \begin{pmatrix} D^2/(1+D^3) & D/(1+D^3) & 1 \end{pmatrix}. \tag{A.14}$$

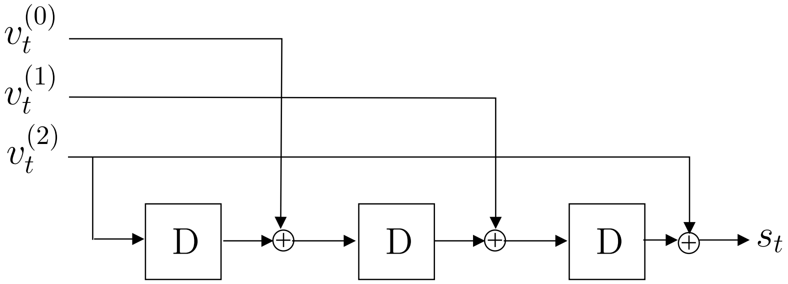


Figure A.2: Syndrome former of Ungerboeck's 8-state TCM-8PSK.

according to Eq. (A.5). We may further simplify Eq. (A.14) by multiplying it with $(1 + D^3)$, which yields:

$$H(D) = \begin{pmatrix} D^2 & D & 1 + D^3 \end{pmatrix}. \tag{A.15}$$

The syndrome former circuit of the PCM of Eq. (A.15) is given in Figure A.2.

Appendix B

Simulation of QLDPC Decoding

The syndrome-based Quantum Low Density Parity Check (QLDPC) decoding algorithms of Chapter 7 may be simulated by invoking the conventional codeword decoding approach in conjunction with the equivalent classical binary or quaternary Parity Check Matrix (PCM). Recall from Chapter 5 that syndrome decoding aims for finding the most likely error pattern inflicted by the channel given the observed syndrome, while the classic codeword decoding aims for finding the most likely transmitted codeword, given the received codeword. If $y = (y_0, y_1, \ldots, y_t, \ldots, y_{n-1})$ is the received sequence, then the transmitted code sequence $v = (v_0, v_1, \ldots, v_t, \ldots, v_{n-1})$ and the channel error $e = (e_0, e_1, \ldots, e_t, \ldots, e_{n-1})$ are related as follows:

$$y_t = v_t \oplus e_t, \tag{B.1}$$

where $\oplus$ denotes the modulo-2 addition. The most likely codeword estimated by the codeword decoding and the most likely error pattern estimated by the syndrome decoding are also related through Eq. (B.1). Hence, these approaches are equivalent.

The codeword-based counterpart of the non-binary QLDPC decoding of Section 7.3.2 may be implemented by:

- initializing the messages according to the received code sequence as well as the channel model; and
- only considering the zero-syndrome scenario of Eq. (7.19).

We assume furthermore the transmission of all-zero codewords for facilitating simulations. Let us consider the decoding example detailed in Section 7.6, where the channel error inflicted is $\mathcal{P} = \mathbf{XIIIIII}$, which is equivalent to (1 0 0 0 0 0 0) according to the Pauli-to-quaternary mapping. Since we are transmitting an all-zero codeword, i.e. $v = (0\,0\,0\,0\,0\,0\,0)$, the received codeword is $y = (1\,0\,0\,0\,0\,0\,0)$. For codeword decoding, the initialization step of Eq. (7.46) takes into account the received sequence

y, as follows:

$$m^{\hat{a}}_{v_t \rightarrow c_i} = \begin{cases} 0.74, & \text{if } \hat{a} = y_t \\ 0.0867, & \text{otherwise }, \end{cases} \tag{B.2}$$

where $\hat{a} \in \{0, 1, \omega, \overline{\omega}\}$ is the tth hypothetically transmitted bit. Furthermore, the Probability Density Function (PDF) of $\check{m}^{\hat{a}_s}_{c_i \rightarrow v_t}$, which is computed in Step 4 of the horizontal message exchange (Eq. (7.56) and (7.57)), is calculated by only using Eq. (7.19).

Analogous to the codeword-based counterpart of the non-binary QLDPC decoding, the codeword-based counterpart of our modified non-binary decoding of Section 7.6 also considers only the zero-syndrome scenario of Eq. (7.41). The resultant decoding algorithm is equivalent to running the conventional (codeword) classical GF(4) decoding over the first $m/2$ rows of the equivalent PCM of Eq. (7.40). However, in the conventional classical GF(4) decoding we have $\hat{a}_s = \hat{H}_{it} \times \hat{a}$, while for QLDPC decoding we use $\hat{a}_s = \hat{H}_{it} \times \overline{\hat{a}}$.

List of Abbreviations

AWGN	-	Additive White Gaussian Noise
APP	-	A-Posteriori Probability
BCH	-	Bose-Chaudhuri-Hocquenghen
BCJR	-	Bahl-Cocke-Jelinek-Raviv
BER	-	Bit Error Rate
BIBD	-	Balanced Incomplete Block Design
BP	-	Belief Propagation
BSC	-	Binary Symmetric Channel
BSD	-	Block Syndrome Decoder
CDF	-	Cyclic Difference Family
CC	-	Convolutional Code
CCC	-	Classical Convolutional Code
CPTP	-	Completely-Positive Trace-Preserving
CNOT	-	Controlled-NOT
CSS	-	Calderbank-Shor-Steane
EA	-	Entanglement-Assisted
EACC	-	Entanglement-Assisted Classical Capacity
EAP	-	Edge Appearance Probability
EG	-	Euclidean Geometry
EPR	-	Einstein-Podolsky-Rosen
EXIT	-	EXtrinsic Information Transfer
FAP	-	Factor Appearance Probability
FER	-	Frame Error Rate
FFT	-	Fast Fourier Transform
GHZ	-	Greenberger-Horne-Zeilinger
GM	-	Generator Matrix
IFFT	-	Inverse Fast Fourier Transform

IRCC	-	IRregular Convolutional Code
LDGM	-	Low Density Generator Matrix
LDPC	-	Low Density Parity Check Code
LLR	-	Log Likelihood Ratio
LUT	-	Look-Up Table
MAP	-	Maximum A-Posteriori
MI	-	Mutual Information
ML	-	Maximum Likelihood
NSD	-	N-qubit SuperDense
PCM	-	Parity Check Matrix
PSK	-	Phase-Shift Keying
QAM	-	Quadrature Amplitude Modulation
QBER	-	QuBit Error Rate
QC	-	Quasi-Cyclic
QCC	-	Quantum Convolutional Code
QECC	-	Quantum Error Correction Code
QIRCC	-	Quantum IRregular Convolutional Code
QKD	-	Quantum Key Distribution
QLDPC	-	Quantum Low-Density Parity Check Code
QSC	-	Quantum Stabilizer Code
QVA	-	Quantum Viterbi Algorithm
QTBC	-	Quantum Tail-biting Block Code
QTC	-	Quantum Turbo Code
RX	-	Receiver
SC	-	Spatially-Coupled
SD	-	SuperDense
SISO	-	Soft-In Soft-Out
SNR	-	Signal-to-Noise Ratio
TC	-	Turbo Code
TCM	-	Trellis Coded Modulation
TRW-BP	-	Tree-ReWeighted Belief Propagation
TTCM	-	Turbo Trellis Coded Modulation
TX	-	Transmitter
URC	-	Unity Rate Code
URW-BP	-	Uniformly-ReWeighted Belief Propagation
WER	-	Word Error Rate

Glossary

Abelian Group A multiplicative group G is Abelian iff $\forall a, b \in G, a \times b = b \times a$.

Basis Basis of a vector space V is a set of linearly independent vectors $\{|v_1\rangle, \ldots, |v_n\rangle\}$ such that any vector $|v\rangle$ in the vector space V can be written as a linear combination of the basis vectors $\{|v_1\rangle, \ldots, |v_n\rangle\}$, i.e. $|v\rangle = \sum_i a_i |v_i\rangle$.

Bell-Basis Measurement Bell-basis measurement is a joint measurement on a 2-qubit composite system for the sake of detecting the orthonormal Bell states.

Bell States The orthonormal 2-qubit states given by:

$$\begin{aligned} \frac{1}{\sqrt{2}}\left(|00\rangle + |11\rangle\right) \ , \quad & \frac{1}{\sqrt{2}}\left(|00\rangle - |11\rangle\right) \ , \\ \frac{1}{\sqrt{2}}\left(|01\rangle + |10\rangle\right) \ , \quad & \frac{1}{\sqrt{2}}\left(|01\rangle - |10\rangle\right) \ , \end{aligned}$$

are known as the Bell state, named after John S. Bell, which are also referred to as the Einstein-Podolsky-Rosen (EPR) pairs.

Centralizer The centralizer of a stabilizer code is a set of elements which commute with the stabilizer generators. It is denoted as $C(\mathcal{H})$, where $\mathcal{H}$ is the stabilizer group of the stabilizer code.

Clifford Operator An n-qubit Clifford operator $\mathcal{V}$ preserves the elements of the Pauli group under conjugation such that for $\mathcal{P} \in \mathcal{G}_n$, $\mathcal{V}\mathcal{P}\mathcal{V}^\dagger \in \mathcal{G}_n$.

Clifford Transformation An n-qubit Clifford transformation $\mathcal{V}$ maps an n-qubit Pauli group $\mathcal{G}_n$ onto itself under conjugation such that $\mathcal{V}\mathcal{G}_n\mathcal{V}^\dagger = \mathcal{G}_n$.

Coherence Time Coherence time may be defined as the time duration over which a qubit retains its coherent quantum state.

Commute Two operators A and B commute if and only if $A \times B = B \times A$.

Coset If G is a group, and H is a subgroup of G, and $g \in G$, then $gH = \{gh : h \in H\}$ is a left coset of H in G, and $Hg = \{hg : h \in H\}$ is a right coset of H in G.

Decoherence Decoherence is the undesirable entanglement of qubits with the environment, which perturbs the fragile superposition of quantum states, thus leading to the detrimental effects of noise. The overall decoherence process may be characterized either by bit-flips or phase-flips or in fact possibly both, inflicted on the qubits.

Degeneracy Pauli errors which differ only by the stabilizer group have the same impact on all the codewords and therefore can be corrected by the same recovery operations. This is known as degeneracy.

Depolarizing Channel The depolarizing channel characterized by the probability p inflicts an n-tuple error $\mathcal{P} \in \mathcal{G}_n$ on n qubits, where the i^{th} qubit may experience either a bit flip ($\mathbf{X}$), a phase flip ($\mathbf{Z}$) or both ($\mathbf{Y}$) with a probability of $p/3$.

Dimension of Vector Space The number of elements in the basis set of the vector space V is defined to be the dimension of V.

Dual Code If G and H are the generator and parity-check matrices for any linear block code C, then its dual code $C^{\perp}$ is a unique code with H^T and G^T as the generator and parity-check matrices respectively.

Dual-Containing Code Code C with parity check matrix H is said to be dual-containing if it contains its dual code $C^{\perp}$, i.e. $C^{\perp} \subset C$ and $HH^T = 0$.

Eigenvector An eigenvector of a linear operator A is a non-zero vector $|v\rangle$ such that $Av = v|v\rangle$, where v is a complex number known as the eigenvalue of A.

Entanglement-Assisted Classical Capacity The Entanglement-Assisted Classical Capacity (EACC) of a quantum channel quantifies the capacity limit of reliable transmission of classical information over a noisy quantum channel, when an unlimited amount of noiseless entanglement is shared between the transmitter and the receiver.

Fidelity Fidelity is a measure of closeness of two quantum states.

Greenberger-Horne-Zeilinger (GHZ) Greenberger-Horne-Zeilinger (GHZ) state is an N-qubit entangled state for $N > 2$ given by:

$$|GHZ\rangle = \frac{1}{\sqrt{2}} \left(|0\rangle^{\otimes N} + |1\rangle^{\otimes N} \right).$$

Hashing Bound Hashing bound determines the code rate at which a random quantum code facilitates reliable transmission for a particular depolarizing probability p.

Hermitian Matrix Hermitian matrix (also called self-adjoint) is a square matrix, which is equivalent to its conjugate transpose, i.e. $A = A^{\dagger}$.

Hilbert Space A d-dimensional Hilbert space is a d-dimensional complex vector space with an inner product, and hence a norm. It is denoted as $\mathbb{C}^d$.

Inner Product Inner product is a function, which takes two complex-valued vectors $|a\rangle$ and $|b\rangle$ as input and yields a complex number as the output. More explicitly, we have:

$$(|a\rangle, |b\rangle) = [a_1^*, \ldots, a_n^*] \begin{pmatrix} b_1 \\ \vdots \\ b_n \end{pmatrix} = \sum_i a_i^* b_i.$$

Inner product $(|a\rangle, |b\rangle)$ is conventionally written as $\langle a|b\rangle$, where the bra notation $\langle v|$ represents the dual (adjoint or hermitian conjugate) of $|v\rangle$.

Noise Limit The noise limit of a quantum depolarizing channel is the maximum tolerable channel depolarizing probability.

Normalizer Normalizer of a stabilizer code is a set of elements which commute with the stabilizer generators. It is denoted as $N(\mathcal{H})$, where $\mathcal{H}$ is the stabilizer group of the stabilizer code.

Outer Product The outer product of a ket and a bra vector $|a\rangle\langle b|$ is a linear operator, whose action on a vector $|v\rangle$ is defined as follows:

$$\left(|a\rangle\langle b|\right)\left(|v\rangle\right) = |a\rangle\langle b|v\rangle.$$

More specifically, $|a\rangle\langle b|$ maps all input vectors $|v\rangle$ to $|a\rangle$ multiplied by the scalar product $\langle b|v\rangle$. The corresponding matrix for the linear operator $|a\rangle\langle b|$ may be computed in $\mathbb{C}^2$ as follows:

$$|a\rangle\langle b| = \begin{pmatrix} a_1 \\ a_2 \end{pmatrix} \begin{pmatrix} b_1^* & b_2^* \end{pmatrix} = \begin{pmatrix} a_1 b_1^* & a_1 b_2^* \\ a_2 b_1^* & a_2 b_2^* \end{pmatrix}.$$

Pauli Group A single qubit Pauli group $\mathcal{G}_1$ consists of all the Pauli matrices together with the multiplicative factors ± 1 and $\pm i$, i.e. we have:

$$\mathcal{G}_1 \equiv \{\pm\mathbf{I}, \pm i\mathbf{I}, \pm\mathbf{X}, \pm i\mathbf{X}, \pm\mathbf{Y}, \pm i\mathbf{Y}, \pm\mathbf{Z}, \pm i\mathbf{Z}\}.$$

Pauli Operators The **I**, **X**, **Y** and **Z** Pauli operators are defined by the following matrices:

$$\mathbf{I} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \ \mathbf{X} = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix},$$
$$\mathbf{Y} = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}, \ \mathbf{Z} = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix},$$

where the **X**, **Y** and **Z** operators anti-commute with each other.

Positive Operator A positive operator A is defined to be an operator such that for any vector $|v\rangle$, the inner product $(|v\rangle, A|v\rangle)$ is a real and non-negative number. If $(|v\rangle, A|v\rangle) > 0$ for all $|v\rangle \neq 0$, then A is said to be positive definite.

Projector The outer product of a unit vector $|a\rangle$ with itself, i.e. $\mathbf{P}_a = |a\rangle\langle a|$, is said to be a project on to the state $|a\rangle$, as shown in Figure B.1.

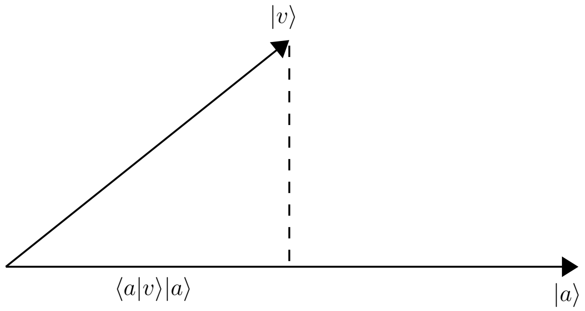


Figure B.1: Geometric interpretation of the projector operator.

Furthermore, $\mathbf{P}_a^2 = \mathbf{P}_a$ and a complete set of orthogonal projectors in n-dimensional Hilbert space is a set $\{\mathbf{P}_1, \dots \mathbf{P}_m\}$ such that $\sum_{i=1}^{m} \mathbf{P}_i = 1$.

Quantum Interleaver An N-qubit quantum interleaver is an N-qubit symplectic transformation, which randomly permutes the N qubits and also applies single-qubit symplectic transformations to the individual qubits.

Stabilizer Code An $[n, k]$ quantum stabilizer code, constructed over a code space $\mathcal{C}$, which maps the information word (logical qubits) $|\psi\rangle \in \mathbb{C}^{2^k}$ onto the codeword (physical qubits) $|\overline{\psi}\rangle \in \mathbb{C}^{2^n}$, where $\mathbb{C}^d$ denotes the d-dimensional Hilbert space, is defined by a set of $(n-k)$ independent commuting n-tuple Pauli operators g_i, for $1 \leq i \leq (n-k)$. More specifically, the corresponding stabilizer group $\mathcal{H}$ contains both g_i and all the products of g_i for $1 \leq i \leq (n-k)$ and forms an Abelian subgroup of $\mathcal{G}_n$. A unique feature of these operators is that they do not change the state of valid codewords, while yielding an eigenvalue of -1 for corrupted states.

Stabilizer Group see Stabilizer Code.

Symplectic Condition If $\mathbf{a} = (a_1, \ldots, a_n)$, $\mathbf{b} = (b_1, \ldots, b_n)$, $\mathbf{a}' = (a'_1, \ldots, a'_n)$ and $\mathbf{b}' = (b'_1, \ldots, b'_n)$, then the symplectic condition between $(\mathbf{a}'|\mathbf{b}')$ and $(\mathbf{a}|\mathbf{b})$[1] is defined as:

$$\langle(\mathbf{a}'|\mathbf{b}'), (\mathbf{a}|\mathbf{b})\rangle_S = \mathbf{a}'\mathbf{b}^{\mathbf{T}} - \mathbf{b}'\mathbf{a}^{\mathbf{T}} = 0. \tag{B.3}$$

Symplectic Inner Product If $\mathbf{a} = (a_1, \ldots, a_n)$, $\mathbf{b} = (b_1, \ldots, b_n)$, $\mathbf{a}' = (a'_1, \ldots, a'_n)$ and $\mathbf{b}' = (b'_1, \ldots, b'_n)$, then the symplectic inner product (also called twisted product or symplectic product) between $(\mathbf{a}'|\mathbf{b}')$ and $(\mathbf{a}|\mathbf{b})$ is defined as:

$$\langle(\mathbf{a}'|\mathbf{b}'), (\mathbf{a}|\mathbf{b})\rangle_S = \mathbf{a}'\mathbf{b}^{\mathbf{T}} - \mathbf{b}'\mathbf{a}^{\mathbf{T}}. \tag{B.4}$$

It can also be represented as $(\mathbf{a}'|\mathbf{b}') \star (\mathbf{a}|\mathbf{b})$.

Tensor Product Given two vector subspaces A and B with dimensions m and n respectively, then the tensor product $A \otimes B$ yields an mn dimensional vector space, whose elements are linear combinations of the tensor products $|a\rangle \otimes |b\rangle$ of elements $|a\rangle$ of A and $|b\rangle$ of B. More explicitly, let $|a\rangle = a_1|0\rangle + a_2|1\rangle$ and $|b\rangle = b_1|0\rangle + b_2|1\rangle$, then we have:

$$\begin{pmatrix} a_1 \\ a_2 \end{pmatrix} \otimes \begin{pmatrix} b_1 \\ b_2 \end{pmatrix} = \begin{pmatrix} a_1 b_1 \\ a_1 b_2 \\ a_2 b_1 \\ a_2 b_2 \end{pmatrix}.$$

Tensor product $|a\rangle \otimes |b\rangle$ may also be abbreviated as $|a\rangle|b\rangle$, $|a, b\rangle$ or $|ab\rangle$.

Trace Trace of a square matrix A is the sum of its diagonal elements, i.e. $\mathrm{tr}(A) = \sum_i A_{ii}$.

[1] Solid vertical bar, i.e. $|$, denotes the concatenation operation, e.g. concatenation of $\mathbf{a}$ and $\mathbf{b}$ is given by $(\mathbf{a}|\mathbf{b})$.

Trace Distance Trace distance between two operators A and B is given by:

$$||A - B||_1 = \text{tr}\left\{\sqrt{(A - B)^\dagger(A - B)}\right\}$$

Trace distance is a measure of separation between two quantum states with density operators ρ and σ. Equivalent quantum states have $||\rho - \sigma||_1 = 0$, while orthogonal states have $||\rho - \sigma||_1 = 2$.

Twisted Product see Symplectic Inner Product.

Unbounded Minimum Distance The unbounded minimum distance of a code implies that its minimum distance increases almost linearly with the interleaver length.

Unitary Operator An operator U is unitary if $UU^\dagger = \mathbf{I}$, where $U^\dagger$ is the adjoint (hermitian conjugate) of U and $\mathbf{I}$ is identity matrix. Unitary operator preserves the inner product.

Bibliography

[1] Z. Babar, P. Botsinis, D. Alanis, S. X. Ng, and L. Hanzo, "Fifteen years of quantum LDPC coding and improved decoding strategies," *IEEE Access (to be submitted).*

[2] Z. Babar, P. Botsinis, D. Alanis, S. X. Ng, and L. Hanzo, "High-rate QLDPC codes from row-circulant classical LDPCs," *IEEE Communications Letters (to be submitted).*

[3] Z. Babar, P. Botsinis, D. Alanis, S. X. Ng, and L. Hanzo, "The road from classical to quantum codes: A hashing bound approaching design procedure," *IEEE Access*, vol. 3, pp. 146–176, 2015.

[4] Z. Babar, S. X. Ng, and L. Hanzo, "EXIT-chart-aided near-capacity quantum turbo code design," *IEEE Transactions on Vehicular Technology*, vol. 64, pp. 866–875, March 2015.

[5] Z. Babar, S. X. Ng, and L. Hanzo, "Near-capacity code design for entanglement-assisted classical communication over quantum depolarizing channels," *IEEE Transactions on Communications*, vol. 61, pp. 4801–4807, December 2013.

[6] Z. Babar, S. X. Ng, and L. Hanzo, "Reduced-complexity syndrome-based TTCM decoding," *IEEE Communications Letters*, vol. 17, pp. 1220–1223, June 2013.

[7] Z. Babar, S. X. Ng, and L. Hanzo, "EXIT-chart aided code design for symbol-based entanglement-assisted classical communication over quantum channels," in *IEEE Vehicular Technology Conference (VTC Fall)*, pp. 1–5, Sept 2014.

[8] L. Hanzo, H. Haas, S. Imre, D. O'Brien, M. Rupp, and L. Gyongyosi, "Wireless myths, realities, and futures: From 3G/4G to optical and quantum wireless," *Proceedings of the IEEE*, vol. 100, pp. 1853–1888, May 2012.

[9] S. Imre and F. Balazs, *Quantum Computing and Communications: An Engineering Approach.* John Wiley & Sons, 2005.

[10] P. Botsinis, S. X. Ng, and L. Hanzo, "Quantum search algorithms, quantum wireless, and a low-complexity maximum likelihood iterative quantum multi-user detector design," *IEEE Access*, vol. 1, pp. 94–122, 2013.

[11] D. Deutsch, "Quantum theory, the Church-Turing principle and the universal quantum computer," *Proceedings of the Royal Society of London. Series A, Mathematical and Physical Sciences*, vol. 400, pp. 97–117, 1985.

[12] D. Deutsch and R. Jozsa, "Rapid solution of problems by quantum computation," *Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, vol. 439, no. 1907, pp. 553–558, 1992.

[13] P. W. Shor, "Algorithms for quantum computation: discrete logarithms and factoring," in *Proceedings of the 35th Annual Symposium on Foundations of Computer Science*, (Washington, DC, USA), pp. 124–134, IEEE Computer Society, 1994.

[14] L. K. Grover, "A fast quantum mechanical algorithm for database search," in *Proceedings of the Twenty-eighth Annual ACM Symposium on Theory of Computing*, STOC '96, (New York, NY, USA), pp. 212–219, ACM, 1996.

[15] P. Botsinis, S. X. Ng, and L. Hanzo, "Fixed-complexity quantum-assisted multi-user detection for CDMA and SDMA," *IEEE Transactions on Communications*, vol. 62, pp. 990–1000, March 2014.

[16] P. Botsinis, D. Alanis, S. X. Ng, and L. Hanzo, "Low-complexity soft-output quantum-assisted multiuser detection for direct-sequence spreading and slow subcarrier-hopping aided SDMA-OFDM systems," *IEEE Access*, vol. 2, pp. 451–472, 2014.

[17] D. Alanis, P. Botsinis, S. X. Ng, and L. Hanzo, "Quantum-assisted routing optimization for self-organizing networks," *IEEE Access*, vol. 2, pp. 614–632, 2014.

[18] M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information.* Cambridge University Press, 2000.

[19] S. Wiesner, "Conjugate coding," *SIGACT News*, vol. 15, pp. 78–88, January 1983.

[20] C. H. Bennett and G. Brassard, "Quantum cryptography: public key distribution and coin tossing," in *Proceedings of the IEEE International Conference on Computers, Systems and Signal Processing*, (New York), pp. 175–179, IEEE Press, 1984.

[21] A. Beige, B.-G. Englert, C. Kurtsiefer, and H. Weinfurter, "Secure communication with single-photon two-qubit states," *Journal of Physics A: Mathematical and General*, vol. 35, no. 28, p. L407, 2002.

[22] K. Boström and T. Felbinger, "Deterministic secure direct communication using entanglement," *Phys. Rev. Lett.*, vol. 89, p. 187902, Oct 2002.

[23] C. Wang, F.-G. Deng, Y.-S. Li, X.-S. Liu, and G. L. Long, "Quantum secure direct communication with high-dimension quantum superdense coding," *Phys. Rev. A*, vol. 71, p. 044305, Apr 2005.

[24] R. A. Malaney, "Location-dependent communications using quantum entanglement," *Phys. Rev. A*, vol. 81, p. 042319, Apr 2010.

[25] P. W. Shor, "Scheme for reducing decoherence in quantum computer memory," *Physical Review A*, vol. 52, pp. R2493–R2496, Oct. 1995.

[26] J. Preskill, "Battling decoherence: the fault-tolerant quantum computer," *Physics Today*, vol. 52, pp. 24–32, 1999.

[27] C. P. Williams and S. H. Clearwater, *Ultimate zero and one - computing at the quantum frontier.* Copernicus, 2000.

[28] C. H. Bennett, "Communication via one- and two-particle operators on Einstein-Podolsky-Rosen states," *Physical Review Letters*, vol. 69, no. 20, p. 2881, 1992.

[29] J. Preskill, "Reliable quantum computers," in *Proc. Royal Soc. of London A*, vol. 454, pp. 385–410, 1998.

[30] D. G. Cory, M. D. Price, W. Maas, E. Knill, R. Laflamme, W. H. Zurek, T. F. Havel, and S. S. Somaroo, "Experimental quantum error correction," *Phys. Rev. Lett.*, vol. 81, pp. 2152–2155, Sep 1998.

[31] M. D. Reed, L. DiCarlo, S. E. Nigg, L. Sun, L. Frunzio, S. M. Girvin, and R. J. Schoelkopf, "Realization of three-qubit quantum error correction with superconducting circuits," *Nature*, vol. 482, pp. 382–385, Feb. 2012.

[32] G. Arrad, Y. Vinkler, D. Aharonov, and A. Retzker, "Increasing sensing resolution with error correction," *Phys. Rev. Lett.*, vol. 112, p. 150801, Apr 2014.

[33] A. R. Calderbank and P. W. Shor, "Good quantum error-correcting codes exist," *Phys. Rev. A*, vol. 54, pp. 1098–1105, Aug 1996.

[34] A. Steane, "Multiple-particle interference and quantum error correction," *Royal Society of London Proceedings Series A*, vol. 452, pp. 2551–2577, Nov. 1995.

[35] A. M. Steane, "Error correcting codes in quantum theory," *Phys. Rev. Lett.*, vol. 77, pp. 793–797, Jul 1996.

[36] R. Laflamme, C. Miquel, J. P. Paz, and W. H. Zurek, "Perfect quantum error correcting code," *Phys. Rev. Lett.*, vol. 77, pp. 198–201, Jul 1996.

[37] C. H. Bennett, D. P. DiVincenzo, J. A. Smolin, and W. K. Wootters, "Mixed-state entanglement and quantum error correction," *Phys. Rev. A*, vol. 54, pp. 3824–3851, Nov 1996.

[38] D. Gottesman, “Class of quantum error-correcting codes saturating the quantum Hamming bound,” *Phys. Rev. A*, vol. 54, pp. 1862–1868, Sep 1996.

[39] D. Gottesman, *Stabilizer Codes and Quantum Error Correction.* PhD thesis, California Institute of Technology, 1997.

[40] M. Grassl, T. Beth, and T. Pellizzari, “Codes for the quantum erasure channel,” *Phys. Rev. A*, vol. 56, pp. 33–38, Jul 1997.

[41] A. Calderbank, E. Rains, P. Shor, and N. J. A. Sloane, “Quantum error correction via codes over GF(4),” *IEEE Transactions on Information Theory*, vol. 44, pp. 1369–1387, Jul 1998.

[42] M. Grassl and T. Beth, “Quantum BCH Codes,” *Proceedings of International Symposium on Theoretical Electrical Engineering Magdeburg*, pp. 207–212, Oct. 1999.

[43] A. Steane, “Enlargement of Calderbank-Shor-Steane quantum codes,” *IEEE Transactions on Information Theory*, vol. 45, pp. 2492–2495, Nov 1999.

[44] A. Steane, “Quantum Reed-Muller codes,” *IEEE Transactions on Information Theory*, vol. 45, pp. 1701–1703, Jul 1999.

[45] M. Grassl, W. Geiselmann, and T. Beth, “Quantum Reed-Solomon codes,” in *Proceedings Applied Algebra, Algebraic Algorithms and Error-Correcting Codes*, pp. 231–244, Springer. See, 1999.

[46] M. S. Postol, “A proposed quantum low density parity check code,” *arXiv:quant-ph/0108131v1*, 2001.

[47] D. MacKay, G. Mitchison, and P. McFadden, “Sparse-graph codes for quantum error correction,” *IEEE Transactions on Information Theory*, vol. 50, pp. 2315–2330, Oct 2004.

[48] T. Camara, H. Ollivier, and J.-P. Tillich, “Constructions and performance of classes of quantum LDPC codes,” *arXiv:quant-ph/0502086v2*, 2005.

[49] T. Camara, H. Ollivier, and J.-P. Tillich, “A class of quantum LDPC codes: construction and performances under iterative decoding,” in *IEEE International Symposium on Information Theory*, pp. 811–815, June 2007.

[50] H. Ollivier and J.-P. Tillich, “Description of a quantum convolutional code,” *Phys. Rev. Lett.*, vol. 91, p. 177902, Oct 2003.

[51] H. Ollivier and J.-P. Tillich, “Quantum convolutional codes: fundamentals,” *quant-ph/0401134*, 2004.

[52] G. Forney and S. Guha, “Simple rate-1/3 convolutional and tail-biting quantum error-correcting codes,” in *International Symposium on Information Theory*, pp. 1028 –1032, Sept. 2005.

[53] G. D. Forney, M. Grassl, and S. Guha, "Convolutional and tail-biting quantum error-correcting codes," *IEEE Transactions on Information Theory*, vol. 53, pp. 865–880, March 2007.

[54] D. Poulin, J.-P. Tillich, and H. Ollivier, "Quantum serial turbo-codes," in *IEEE International Symposium on Information Theory*, pp. 310–314, July 2008.

[55] D. Poulin, J.-P. Tillich, and H. Ollivier, "Quantum serial turbo codes," *IEEE Transactions on Information Theory*, vol. 55, pp. 2776–2798, June 2009.

[56] J. M. Renes, F. Dupuis, and R. Renner, "Efficient polar coding of quantum information," *Phys. Rev. Lett.*, vol. 109, p. 050504, Aug 2012.

[57] M. M. Wilde and S. Guha, "Polar codes for degradable quantum channels," *IEEE Transactions on Information Theory*, vol. 59, pp. 4718–4729, July 2013.

[58] J. Renes and M. M. Wilde, "Polar codes for private and quantum communication over arbitrary channels," *IEEE Transactions on Information Theory*, vol. 60, pp. 3090–3103, June 2014.

[59] P. Piret, *Convolutional Codes: An Algebraic Approach.* MIT Press, Cambridge, MA, 1988.

[60] H. F. Chau, "Quantum convolutional error-correcting codes," *Phys. Rev. A*, vol. 58, pp. 905–909, Aug 1998.

[61] H. F. Chau, "Good quantum-convolutional error-correction codes and their decoding algorithm exist," *Phys. Rev. A*, vol. 60, pp. 1966–1974, Sep 1999.

[62] A. Aido De Almeida and J. Palazzo, R., "A concatenated [(4, 1, 3)] quantum convolutional code," in *IEEE Information Theory Workshop*, pp. 28 – 33, Oct. 2004.

[63] M. Grassl and M. Rotteler, "Quantum block and convolutional codes from self-orthogonal product codes," in *International Symposium on Information Theory*, pp. 1018 –1022, Sep. 2005.

[64] M. Grassl and M. Rotteler, "Constructions of quantum convolutional codes," in *IEEE International Symposium on Information Theory*, pp. 816–820, June 2007.

[65] S. Aly, M. Grassl, A. Klappenecker, M. Rotteler, and P. Sarvepalli, "Quantum convolutional BCH codes," in *10th Canadian Workshop on Information Theory*, pp. 180 –183, June 2007.

[66] S. A. Aly, A. Klappenecker, and P. K. Sarvepalli, "Quantum convolutional codes derived from generalized Reed-Solomon codes," in *IEEE International Symposium on Information Theory*, pp. 821 –825, June 2007.

[67] E. Pelchat and D. Poulin, "Degenerate Viterbi decoding," *IEEE Transactions on Information Theory*, vol. 59, no. 6, pp. 3915–3921, 2013.

[68] L. Hanzo, T. H. Liew, B. L. Yeap, R. Y. S. Tee and S. X. Ng, *Turbo Coding, Turbo Equalisation and Space-Time Coding: EXIT-Chart-Aided Near-Capacity Designs for Wireless Channels, 2nd Edition.* New York, USA: John Wiley IEEE Press, March 2011.

[69] R. Gallager, “Low-density parity-check codes,” *IRE Transactions on Information Theory*, vol. 8, pp. 21–28, January 1962.

[70] D. Mackay, G. Mitchison, and A. Shokrollahi, “More sparse-graph codes for quantum error-correction,” *www.inference.phy.cam.ac.uk/mackay/cayley.pdf*, 2007.

[71] A. Couvreur, N. Delfosse, and G. Zemor, “A construction of quantum LDPC codes from cayley graphs,” in *IEEE International Symposium on Information Theory Proceedings*, pp. 643 –647, 31 2011-aug. 5 2011.

[72] A. Couvreur, N. Delfosse, and G. Zemor, “A construction of quantum LDPC codes from cayley graphs,” *IEEE Transactions on Information Theory*, vol. 59, pp. 6087–6098, Sept 2013.

[73] S. Aly, “A class of quantum LDPC codes constructed from finite geometries,” in *IEEE Global Telecommunications Conference*, pp. 1 –5, 30 2008-dec. 4 2008.

[74] I. Djordjevic, “Quantum LDPC codes from balanced incomplete block designs,” *IEEE Communications Letters*, vol. 12, pp. 389 –391, may 2008.

[75] H. Lou and J. Garcia-Frias, “Quantum error-correction using codes with low-density generator matrix,” in *IEEE 6th Workshop on Signal Processing Advances in Wireless Communications*, pp. 1043 – 1047, june 2005.

[76] H. Lou and J. Garcia-Frias, “On the application of error-correcting codes with low-density generator matrix over different quantum channels,” *International ITG-Conference on Source and Channel Coding*, pp. 1 –6, april 2006.

[77] M. Hagiwara and H. Imai, “Quantum quasi-cyclic LDPC codes,” in *IEEE International Symposium on Information Theory*, pp. 806 –810, june 2007.

[78] K. Kasai, M. Hagiwara, H. Imai, and K. Sakaniwa, “Non-binary quasi-cyclic quantum LDPC codes,” in *IEEE International Symposium on Information Theory Proceedings*, pp. 653 –657, 31 2011-aug. 5 2011.

[79] K. Kasai, M. Hagiwara, H. Imai, and K. Sakaniwa, “Quantum error correction beyond the bounded distance decoding limit,” *IEEE Transactions on Information Theory*, vol. 58, pp. 1223 –1230, feb. 2012.

[80] M. Hagiwara, K. Kasai, H. Imai, and K. Sakaniwa, “Spatially coupled quasi-cyclic quantum LDPC codes,” in *IEEE International Symposium on Information Theory Proceedings*, pp. 638 –642, 31 2011-aug. 5 2011.

[81] P. Tan and J. Li, “Efficient quantum stabilizer codes: LDPC and LDPC-convolutional constructions,” *IEEE Transactions on Information Theory*, vol. 56, pp. 476 –491, jan. 2010.

[82] N. Delfosse, “Tradeoffs for reliable quantum information storage in surface codes and color codes,” in *IEEE International Symposium on Information Theory Proceedings*, pp. 917–921, July 2013.

[83] A. A. Kovalev and L. P. Pryadko, “Quantum Kronecker sum-product low-density parity-check codes with finite rate,” *Phys. Rev. A*, vol. 88, p. 012311, Jul 2013.

[84] J.-P. Tillich and G. Zemor, “Quantum LDPC codes with positive rate and minimum distance proportional to the square root of the blocklength,” *IEEE Transactions on Information Theory*, vol. 60, pp. 1193–1202, Feb 2014.

[85] D. Poulin and Y. Chung, “On the iterative decoding of sparse quantum codes,” *Quantum Info. Comput.*, vol. 8, pp. 987–1000, Nov. 2008.

[86] Y.-J. Wang, B. Sanders, B.-M. Bai, and X.-M. Wang, “Enhanced feedback iterative decoding of sparse quantum codes,” *IEEE Transactions on Information Theory*, vol. 58, pp. 1231 –1241, feb. 2012.

[87] M. Houshmand and M. M. Wilde, “Recursive quantum convolutional encoders are catastrophic: A simple proof,” *IEEE Transactions on Information Theory*, vol. 59, no. 10, pp. 6724–6731, 2013.

[88] M. M. Wilde, M.-H. Hsieh, and Z. Babar, “Entanglement-assisted quantum turbo codes,” *IEEE Transactions on Information Theory*, vol. 60, pp. 1203–1222, Feb 2014.

[89] G. Bowen, “Entanglement required in achieving entanglement-assisted channel capacities,” *Phys. Rev. A*, vol. 66, p. 052313, Nov 2002.

[90] T. A. Brun, I. Devetak, and M.-H. Hsieh, “Correcting quantum errors with entanglement,” *Science*, vol. 314, oct. 2006.

[91] T. A. Brun, I. Devetak, and M.-H. Hsieh, “General entanglement-assisted quantum error-correcting codes,” in *IEEE International Symposium on Information Theory*, pp. 2101 –2105, june 2007.

[92] M.-H. Hsieh, I. Devetak, and T. Brun, “General entanglement-assisted quantum error-correcting codes,” *Phys. Rev. A*, vol. 76, p. 062313, Dec 2007.

[93] M.-H. Hsieh, T. A. Brun, and I. Devetak, “Entanglement-assisted quantum quasicyclic low-density parity-check codes,” *Phys. Rev. A*, vol. 79, p. 032340, Mar 2009.

[94] M. M. Wilde and T. A. Brun, “Entanglement-assisted quantum convolutional coding,” *Phys. Rev. A*, vol. 81, p. 042333, Apr 2010.

[95] M. M. Wilde and M.-H. Hsieh, “Entanglement boosts quantum turbo codes,” in *IEEE International Symposium on Information Theory Proceedings*, pp. 445 – 449, Aug. 2011.

[96] M. M. Wilde and J. Renes, “Quantum polar codes for arbitrary channels,” in *IEEE International Symposium on Information Theory Proceedings*, pp. 334 –338, july 2012.

[97] P. A. Dirac, *The Principles of Quantum Mechanics.* Oxford University Press, 1982.

[98] G. M. Dan C.Marinescu, *Classical & Quantum Information.* Academic Press, 2010.

[99] W. K. Wootters and W. H. Zurek, “A single quantum cannot be cloned,” *Nature*, vol. 299, pp. 802–803, Oct. 1982.

[100] M. Born, *The Born-Einstein letters.* Walker, 1971.

[101] C. H. Bennett, G. Brassard, C. Crépeau, R. Jozsa, A. Peres, and W. K. Wootters, “Teleporting an unknown quantum state via dual classical and Einstein-Podolsky-Rosen channels,” *Phys. Rev. Lett.*, vol. 70, pp. 1895–1899, Mar 1993.

[102] E. Desurvire, *Classical and Quantum Information Theory: An Introduction for the Telecom Scientist.* Cambridge University Press, 2009.

[103] L. Ioffe and M. Mézard, “Asymmetric quantum error-correcting codes,” *Phys. Rev. A*, vol. 75, p. 032345, Mar 2007.

[104] P. Sarvepalli, A. Klappenecker, and M. Rotteler, “Asymmetric quantum LDPC codes,” in *IEEE International Symposium on Information Theory*, pp. 305 –309, july 2008.

[105] Y. Fujiwara and V. Tonchev, “A characterization of entanglement-assisted quantum low-density parity-check codes,” *IEEE Transactions on Information Theory*, vol. 59, pp. 3347–3353, June 2013.

[106] C. H. Bennett, P. W. Shor, J. A. Smolin, and A. V. Thapliyal, “Entanglement-assisted classical capacity of noisy quantum channels,” *Phys. Rev. Lett.*, vol. 83, pp. 3081–3084, Oct 1999.

[107] A. Holevo, “On entanglement-assisted classical capacity,” *J. Math. Phys.*, vol. 43, no. 9, pp. 4326–4333, 2002.

[108] S. Bose, V. Vedral, and P. L. Knight, “A multiparticle generalization of entanglement swapping,” *Phys. Rev. A*, vol. 57, no. quant-ph/9708004. 2, pp. 822–829, 1998.

[109] A. Chiuri, S. Giacomini, C. Macchiavello, and P. Mataloni, “Experimental achievement of the entanglement-assisted capacity for the depolarizing channel,” *Phys. Rev. A*, vol. 87, p. 022333, Feb 2013.

[110] P. Kaye, R. Laflamme, and M. Mosca, *An introduction to quantum computing.* Oxford University Press, 2007.

[111] D. M. Greenberger, M. A. Horne, A. Shimony, and A. Zeilinger, “Bell’s theorem without inequalities,” *American Journal of Physics*, vol. 58, pp. 1131–1143, Dec. 1990.

[112] Z. Shadman, H. Kampermann, C. Macchiavello, and D. Bruss, "Optimal super dense coding over noisy quantum channels," *New Journal of Physics*, vol. 12, July 2010.

[113] C. E. Shannon, "A Mathematical Theory of Communication," *The Bell system Technical Journal*, vol. 27, pp. 379–656, July 1948.

[114] J. Kliewer, S. X. Ng, and L. Hanzo, "Efficient computation of EXIT functions for non-binary iterative decoding," *IEEE Transactions on Communications*, vol. 54, pp. 2133–2136, December 2006.

[115] S. X. Ng, O. Alamri, Y. Li, J. Kliewer, and L. Hanzo, "Near-capacity turbo trellis coded modulation design based on EXIT charts and union bounds," *IEEE Transactions on Communications*, vol. 56, pp. 2030 –2039, December 2008.

[116] S. ten Brink, "Convergence behaviour of iteratively decoded parallel concatenated codes," *IEEE Transactions on Communications*, vol. 49, pp. 1727–1737, October 2001.

[117] M. El-Hajjar and L. Hanzo, "EXIT charts for system design and analysis," *IEEE Communications Surveys Tutorials*, pp. 1–27, 2013.

[118] M. Tüchler and J. Hagenauer, "EXIT charts of irregular codes," in *Proceedings of Conference on Information Science and Systems*, (Princeton University), pp. 465–490, 20-22 March 2002.

[119] M. Tüchler, "Design of serially concatenated systems depending on the block length," *IEEE Transactions on Communications*, vol. 52, pp. 209–218, February 2004.

[120] K. R. Narayanan, "Effect of precoding on the convergence of turbo equalization for partial response channels," *IEEE Journal on Selected Areas in Communications*, vol. 19, pp. 686–698, April 2001.

[121] A. Ashikhmin, G. Kramer, and S. ten Brink, "Extrinsic information transfer functions: model and erasure channel properties," *IEEE Transactions on Information Theory*, vol. 50, pp. 2657–2673, November 2004.

[122] C. Weidmann and G. Lechner, "A fresh look at coding for q-ary symmetric channels," *IEEE Transactions on Information Theory*, vol. 58, pp. 6959–6967, Nov 2012.

[123] R. Cleve, "Quantum stabilizer codes and classical linear codes," *Phys. Rev. A*, vol. 55, pp. 4054–4059, Jun 1997.

[124] S. Lin and D. J. Costello, *Error Control Coding.* Pearson Education India.

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

[126] J. Wolf, "Efficient maximum likelihood decoding of linear block codes using a trellis," *IEEE Transactions on Information Theory*, vol. 24, pp. 76 – 80, jan 1978.

[127] J. Schalkwijk and A. Vinck, "Syndrome decoding of convolutional codes," *IEEE Transactions on Communications*, vol. 23, pp. 789 – 792, jul 1975.

[128] J. Schalkwijk and A. Vinck, "Syndrome decoding of binary rate-1/2 convolutional codes," *IEEE Transactions on Communications*, vol. 24, pp. 977 – 985, sep 1976.

[129] J. Schalkwijk, A. Vinck, and K. Post, "Syndrome decoding of binary-rate k/n convolutional codes," *IEEE Transactions on Information Theory*, vol. 24, pp. 553 – 562, sep 1978.

[130] M. Ariel and J. Snyders, "Soft syndrome decoding of binary convolutional codes," *IEEE Transactions on Communications*, vol. 43, pp. 288 – 297, Feb./Mar./Apr. 1995.

[131] V. Sidorenko and V. Zyablov, "Decoding of convolutional codes using a syndrome trellis," *IEEE Transactions on Information Theory*, vol. 40, pp. 1663 –1666, sep 1994.

[132] T. Minowa and H. Imai, "Decoding of high-rate turbo codes using a syndrome trellis," *IEEE International Conference on Communications*, vol. 1, pp. 74 – 78 vol.1, Jun. 2001.

[133] J. Geldmacher, K. Hueske, and J. Götze, "Syndrome based block decoding of convolutional codes," *IEEE International Symposium on Wireless Communication Systems*, pp. 542 – 546, Oct. 2008.

[134] K. Hueske, J. Geldmacher, and J. Götze, "Adaptive decoding of convolutional codes," *Advances in Radio Science*, vol. 5, pp. 209–214, 2007.

[135] K. Hueske, J. Geldmacher, and J. Götze, "Multi core implementation of a trellis based syndrome decoder with adaptive complexity," in *International Symposium on Wireless Communication Systems*, pp. 859 –863, sept. 2010.

[136] J. Geldmacher, K. Hueske, J. Götze, and M. Kosakowski, "Low complexity syndrome based decoding of turbo codes," *IEEE International Symposium on Information Theory Proceedings*, pp. 2371 – 2375, Jul. 2012.

[137] J. Geldmacher, K. Hueske, J. Götze, and S. Bialas, "Adaptive low complexity MAP decoding for turbo equalization," *International Symposium on Turbo Codes and Iterative Information Processing*, pp. 63 – 67, Sept. 2010.

[138] J. Geldmacher, K. Hueske, and J. Götze, "An adaptive and complexity reduced decoding algorithm for convolutional codes and its application to digital broadcasting systems," in *International Conference on Ultra Modern Telecommunications Workshops*, pp. 1 –7, oct. 2009.

[139] K. Hueske, J. Geldmacher, and J. Götze, "Syndrome based adaptive complexity channel decoding and turbo equalization for atsc dtv," in *Conference Record of the Forty Fourth Asilomar Conference on Signals, Systems and Computers*, pp. 155 –159, nov. 2010.

[140] J. Geldmacher, K. Hueske, J. Götze, and M. Kosakowski, "Hard decision based low SNR early termination for LTE turbo decoding," in *International Symposium on Wireless Communication Systems*, pp. 26 –30, nov. 2011.

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

[142] M. Tajima, K. Shibata, and Z. Kawasaki, "Relation between encoder and syndrome former variables and symbol reliability estimation using a syndrome trellis," *IEEE Transactions on Communications*, vol. 51, pp. 1474 – 1484, sept. 2003.

[143] T. Filler, J. Judas, and J. Fridrich, "Minimizing additive distortion in steganography using syndrome-trellis codes," *IEEE Transactions on Information Forensics and Security*, vol. 6, pp. 920 –935, sept. 2011.

[144] M. Tajima, K. Shibata, and Z. Kawasaki, "Modification of syndrome trellises for high-rate convolutional codes," *IEICE Technical Report*, vol. 102, pp. 7 – 12, May 2002.

[145] P. Robertson and T. Worz, "Bandwidth-efficient turbo trellis-coded modulation using punctured component codes," *IEEE Journal on Selected Areas in Communications*, vol. 16, pp. 206 – 218, Feb. 1998.

[146] G. Ungerboeck, "Channel coding with multilevel/phase signals," *IEEE Transactions on Information Theory*, vol. 28, pp. 55 – 67, Jan. 1982.

[147] H. Chen and A. Haimovich, "Exit charts for turbo trellis-coded modulation," *IEEE Communications Letters*, vol. 8, pp. 668 – 670, nov. 2004.

[148] S. Lloyd, "Capacity of the noisy quantum channel," *Phys. Rev. A*, vol. 55, pp. 1613–1622, Mar 1997.

[149] P. W. Shor, "The quantum channel capacity and coherent information," *Lecture Notes, MSRI Workshop on Quantum Computation*, 2002.

[150] I. Devetak, "The private classical capacity and quantum capacity of a quantum channel," *IEEE Transactions on Information Theory*, vol. 51, pp. 44–55, Jan 2005.

[151] C. Berrou, A. Glavieux, and P. Thitimajshima, "Near Shannon limit error-correcting coding and decoding: Turbo-codes. 1," in *IEEE International Conference on Communications*, vol. 2, pp. 1064–1070 vol.2, May 1993.

[152] D. P. DiVincenzo, P. W. Shor, and J. A. Smolin, "Quantum-channel capacity of very noisy channels," *Phys. Rev. A*, vol. 57, pp. 830–839, Feb 1998.

[153] G. Smith and J. A. Smolin, "Degenerate quantum codes for pauli channels," *Phys. Rev. Lett.*, vol. 98, p. 030501, Jan 2007.

[154] C.-Y. Lai, T. A. Brun, and M. M. Wilde, “Dualities and identities for entanglement-assisted quantum codes,” *Quantum Information Processing*, vol. 13, no. 4, pp. 957–990, 2014.

[155] J. Dehaene and B. De Moor, “Clifford group, stabilizer states, and linear and quadratic operations over GF(2),” *Phys. Rev. A*, vol. 68, p. 042318, Oct 2003.

[156] D. Gottesman, *The Heisenberg Representation of Quantum Computers.* [online] http://arxiv.org/pdf/quant-ph/9807006v1.pdf, December 2001.

[157] S. Ten Brink, “Rate one-half code for approaching the Shannon limit by 0.1 dB,” *Electronics Letters*, vol. 36, no. 15, pp. 1293–1294, 2000.

[158] L. Kong, S. X. Ng, R. Maunder, and L. Hanzo, “Maximum-throughput irregular distributed space-time code for near-capacity cooperative communications,” *IEEE Transactions on Vehicular Technology*, vol. 59, no. 3, pp. 1511–1517, 2010.

[159] S. Ibi, T. Matsumoto, R. Thoma, S. Sampei, and N. Morinaga, “EXIT chart-aided adaptive coding for multilevel BICM with turbo equalization in frequency-selective MIMO channels,” *IEEE Transactions on Vehicular Technology*, vol. 56, no. 6, pp. 3757–3769, 2007.

[160] A. Grant, “Convergence of non-binary iterative decoding,” in *IEEE Global Telecommunications Conference*, vol. 2, pp. 1058–1062 vol.2, 2001.

[161] M. M. Wilde, *Quantum Information Theory.* Cambridge University Press, May 2013.

[162] D. P. Divincenzo, D. Leung, and B. Terhal, “Quantum data hiding,” *IEEE Transactions on Information Theory*, vol. 48, pp. 580–598, Mar 2002.

[163] E. Rains, “Nonbinary quantum codes,” *IEEE Transactions on Information Theory*, vol. 45, no. 6, pp. 1827–1832, 1999.

[164] D. MacKay, “Good error-correcting codes based on very sparse matrices,” *IEEE Transactions on Information Theory*, vol. 45, pp. 399–431, Mar 1999.

[165] S.-Y. Chung, J. Forney, G.D., T. Richardson, and R. Urbanke, “On the design of low-density parity-check codes within 0.0045 db of the shannon limit,” *IEEE Communications Letters*, vol. 5, pp. 58–60, Feb 2001.

[166] N. Bonello, S. Chen, and L. Hanzo, “Low-density parity-check codes and their rateless relatives,” *IEEE Communications Surveys Tutorials*, vol. 13, pp. 3–26, First 2011.

[167] N. Bonello, S. Chen, and L. Hanzo, “Design of low-density parity-check codes,” *IEEE Vehicular Technology Magazine*, vol. 6, pp. 16–23, Dec 2011.

[168] H. Wymeersch, F. Penna, and V. Savic, “Uniformly reweighted belief propagation: A factor graph approach,” in *IEEE International Symposium on Information Theory Proceedings*, pp. 2000–2004, July 2011.

[169] H. Wymeersch, F. Penna, and V. Savic, "Uniformly reweighted belief propagation for estimation and detection in wireless networks," *IEEE Transactions on Wireless Communications*, vol. 11, pp. 1587–1595, April 2012.

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

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

[172] B. Ammar, B. Honary, Y. Kou, J. Xu, and S. Lin, "Construction of low-density parity-check codes based on balanced incomplete block designs," *IEEE Transactions on Information Theory*, vol. 50, pp. 1257 – 1269, june 2004.

[173] A. Kitaev, "Fault-tolerant quantum computation by anyons," *Annals of Physics*, vol. 303, no. 1, pp. 2 – 30, 2003.

[174] H. Bombin and M. A. Martin-Delgado, "Homological error correction: Classical and quantum codes," *Journal of Mathematical Physics*, vol. 48, no. 5, pp. –, 2007.

[175] H. Bombin and M. Martin-Delgado, "Topological quantum distillation," *Phys. Rev. Lett.*, vol. 97, p. 180501, Oct 2006.

[176] S. ten Brink, "Code doping for triggering iterative decoding convergence," in *IEEE International Symposium on Information Theory*, pp. 235–, 2001.

[177] M. Hagiwara and H. Imai, "Quantum quasi-cyclic LDPC codes," in *arXiv:quant-ph/0701020v4*, Aug 2010.

[178] I. Andriyanova, D. Maurice, and J.-P. Tillich, "Spatially coupled quantum ldpc codes," in *IEEE Information Theory Workshop*, pp. 327–331, Sept 2012.

[179] D. Maurice, J.-P. Tillich, and I. Andriyanova, "A family of quantum codes with performances close to the hashing bound under iterative decoding," in *IEEE International Symposium on Information Theory Proceedings*, pp. 907–911, July 2013.

[180] M.-H. Hsieh, W.-T. Yen, and L.-Y. Hsu, "Performance of entanglement-assisted quantum LDPC codes constructed from finite geometries," *http://arxiv-web3.library.cornell.edu/abs/0906.5532v1*, 2009.

[181] M.-H. Hsieh, W.-T. Yen, and L.-Y. Hsu, "High performance entanglement-assisted quantum LDPC codes need little entanglement," *Information Theory, IEEE Transactions on*, vol. 57, pp. 1761 –1769, march 2011.

[182] I. B. Djordjevic, "Photonic entanglement-assisted quantum low-density parity-check encoders and decoders," *Opt. Lett.*, vol. 35, pp. 1464 – 1466, May 2010.

[183] Y. Fujiwara, D. Clark, P. Vandendriessche, M. De Boeck, and V. Tonchev, "Entanglement-assisted quantum low-density parity-check codes," *Phys. Rev. A*, vol. 82, p. 042338, Oct 2010.

[184] Y. Fujiwara, A. Gruner, and P. Vandendriessche, "High-rate quantum low-density parity-check codes assisted by reliable qubits," *IEEE Transactions on Information Theory*, vol. 61, pp. 1860–1878, April 2015.

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

[186] Y. Fujiwara, "Quantum error correction via less noisy qubits," *Phys. Rev. Lett.*, vol. 110, p. 170501, Apr 2013.

[187] D. J. MacKay, *Information Theory, Inference, and Learning Algorithms.* Cambridge University Press, 2003.

[188] E. Berlekamp, R. McEliece, and H. Van Tilborg, "On the inherent intractability of certain coding problems (corresp.)," *IEEE Transactions on Information Theory*, vol. 24, pp. 384–386, May 1978.

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

[190] D. Lidar and T. Brun, *Quantum Error Correction.* Cambridge University Press, 2013.

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

[192] S. Johnson and S. R. Weller, "A family of irregular LDPC codes with low encoding complexity," *IEEE Communications Letters*, vol. 7, pp. 79–81, Feb 2003.

[193] H. Park, S. Hong, J.-S. No, and D.-J. Shin, "Construction of high-rate regular quasi-cyclic ldpc codes based on cyclic difference families," *IEEE Transactions on Communications*, vol. 61, pp. 3108–3113, August 2013.

[194] R. Bose, "On the construction of balanced incomplete block designs," *Ann. Eugen*, vol. 9, pp. 353 – 399, 1939.

[195] W. Ryan and S. Lin, *Channel Codes: Classical and Modern.* Cambridge University Press, 2009.

[196] M. Wainwright, T. Jaakkola, and A. Willsky, "A new class of upper bounds on the log partition function," *IEEE Transactions on Information Theory*, vol. 51, pp. 2313–2335, July 2005.

[197] T. Roosta, M. Wainwright, and S. Sastry, "Convergence analysis of reweighted sum-product algorithms," *IEEE Transactions on Signal Processing*, vol. 56, pp. 4293–4305, Sept 2008.

[198] J. Liu and R. de Lamare, "Knowledge-aided reweighted belief propagation decoding for regular and irregular LDPC codes with short blocks," in *International Symposium on Wireless Communication Systems*, pp. 984–988, Aug 2012.

[199] J. Liu and R. de Lamare, "Low-latency reweighted belief propagation decoding for LDPC codes," *IEEE Communications Letters*, vol. 16, pp. 1660–1663, October 2012.

[200] J. Liu, R. de Lamare, and H. Wymeersch, "Locally-optimized reweighted belief propagation for decoding finite-length LDPC codes," in *IEEE Wireless Communications and Networking Conference*, pp. 4311–4316, April 2013.

[201] H. Wymeersch, F. Penna, V. Savic, and J. Zhao, "Comparison of reweighted message passing algorithms for LDPC decoding," in *IEEE International Conference on Communications*, pp. 3264–3269, June 2013.

[202] I. Kremsky, M.-H. Hsieh, and T. A. Brun, "Classical enhancement of quantum-error-correcting codes," *Phys. Rev. A*, vol. 78, p. 012341, Jul 2008.

[203] M.-H. Hsieh and M. M. Wilde, "Entanglement-assisted communication of classical and quantum information," *IEEE Transactions on Information Theory*, vol. 56, pp. 4682 –4704, sept. 2010.

[204] M.-H. Hsieh and M. M. Wilde, "Trading classical communication, quantum communication, and entanglement in quantum shannon theory," *IEEE Transactions on Information Theory*, vol. 56, pp. 4705–4730, Sept 2010.

[205] Nasruminallah and L. Hanzo, "Short block codes for guaranteed convergence in soft-bit assisted iterative joint source and channel decoding," *Electronics Letters*, vol. 44, pp. 1315–1316, October 2008.

[206] Nasruminallah and L. Hanzo, "EXIT-chart optimized short block codes for iterative joint source and channel decoding in h.264 video telephony," *IEEE Transactions on Vehicular Technology*, vol. 58, pp. 4306–4315, Oct 2009.