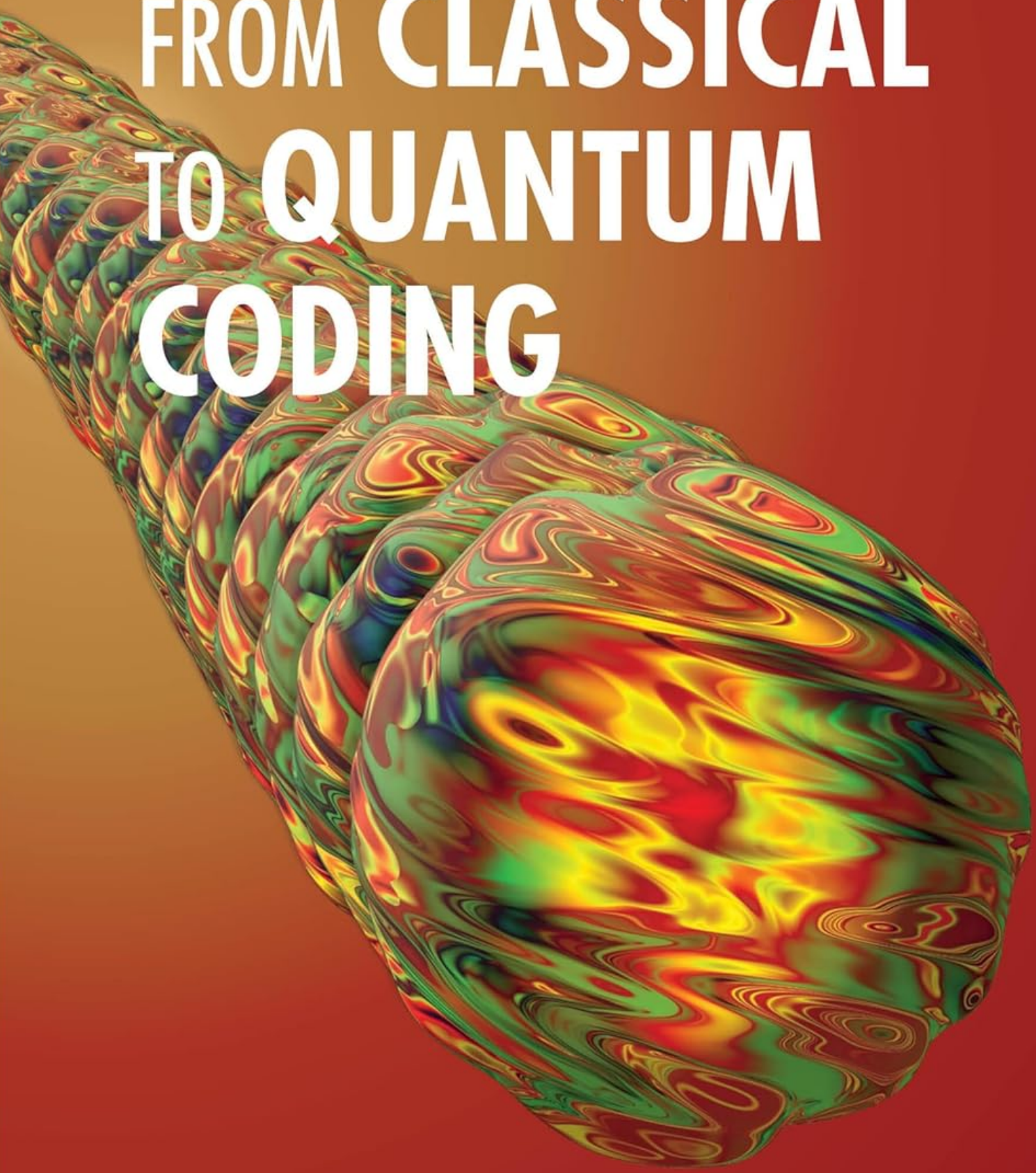
ZUNAIRA BABAR • DARYUS CHANDRA
SOON XIN NG • LAJOS HANZO

FROM CLASSICAL TO QUANTUM CODING



IEEE Press
WILEY

From Classical to Quantum Coding

by

©Z. Babar, D. Chandra, SX. Ng, L. Hanzo

School of Electronics and Computer Science,

University of Southampton, UK

November 22, 2025

About the Authors

Zunaira Babar received B.Eng. degree from the National University of Science and Technology (NUST), Pakistan, in 2008, and the M.Sc. (Hons) and Ph.D. degrees from the University of Southampton, UK, in 2011 and 2015, respectively. Currently, she is a staff research scientist at VIAVI Marconi Labs, UK. She published numerous journal papers on quantum communications.

Daryus Chandra received the B.Eng. and M.Eng. degrees from Universitas Gadjah Mada (UGM), Indonesia, in 2013 and 2014, respectively, and the Ph.D. degree from the University of Southampton, UK, in 2020. Currently, he is a research fellow at the University of Southampton. He published widely on quantum communications.

Soon Xin Ng (S'99-M'03-SM'08) received the B.Eng. degree (First class) in electronic engineering and the Ph.D. degree in telecommunications from the University of Southampton, Southampton, U.K., in 1999 and 2002, respectively. He is currently a Professor of Next Generation Communications at the University of Southampton. He has published widely in classical and quantum communications. He is a Senior Member of the IEEE, a Chartered Engineer, a Fellow of the Higher Education Academy in the UK and a Fellow of IET.

Lajos Hanzo (FIEEE'04, Fellow of the Royal Academy of Engineering (FREng), of the IET and of EURASIP) received Honorary Doctorates from the Technical University of Budapest (2009) and Edinburgh University (2015). He is a Foreign Member of the Hungarian Science-Academy, Fellow of the Royal Academy of Engineering (FREng), of the IET, of EURASIP and holds the IEEE Eric Sumner Technical Field Award.

Other Wiley/IEEE Press Books by the Authors

- R. Steele, L. Hanzo (Ed): *Mobile Radio Communications: Second and Third Generation Cellular and WATM Systems*, John Wiley and IEEE Press, 2nd edition, 1999, ISBN 07 273-1406-8, 1064 pages [1]
- L. Hanzo, T.H. Liew, B.L. Yeap: *Turbo Coding, Turbo Equalisation and Space-Time Coding*, John Wiley and IEEE Press, 2002, 751 pages
- L. Hanzo, C.H. Wong, M.S. Yee: *Adaptive Wireless Transceivers: Turbo-Coded, Turbo-Equalised and Space-Time Coded TDMA, CDMA and OFDM Systems*, John Wiley and IEEE Press, 2002, 737 pages
- L. Hanzo, L-L. Yang, E-L. Kuan, K. Yen: *Single- and Multi-Carrier CDMA: Multi-User Detection, Space-Time Spreading, Synchronisation, Networking and Standards*, John Wiley and IEEE Press, June 2003, 1060 pages
- L. Hanzo, M. Münster, T. Keller, B-J. Choi, *OFDM and MC-CDMA for Broadband Multi-User Communications, WLANs and Broadcasting*, John-Wiley and IEEE Press, 2003, 978 pages
- L. Hanzo, S-X. Ng, T. Keller and W.T. Webb, *Quadrature Amplitude Modulation: From Basics to Adaptive Trellis-Coded, Turbo-Equalised and Space-Time Coded OFDM, CDMA and MC-CDMA Systems*, John Wiley and IEEE Press, 2004, 1105 pages
- L. Hanzo, T. Keller: *An OFDM and MC-CDMA Primer*, John Wiley and IEEE Press, 2006, 430 pages
- L. Hanzo, F.C.A. Somerville, J.P. Woodard: *Voice and Audio Compression for Wireless Communications*, John Wiley and IEEE Press, 2007, 858 pages
- L. Hanzo, P.J. Cherriman, J. Streit: *Video Compression and Communications: H.261, H.263, H.264, MPEG4 and HSDPA-Style Adaptive Turbo-Transceivers* John Wiley and IEEE Press, 2007, 680 pages
- L. Hanzo, J.S. Blogh, S. Ni: *3G, HSDPA, HSUPA and FDD Versus TDD Networking: Smart Antennas and Adaptive Modulation* John Wiley and IEEE Press, 2008, 564 pages

[1] For detailed contents and sample chapters please refer to http://www-mobile.ecs.soton.ac.uk

- L. Hanzo, O. Alamri, M. El-Hajjar, N. Wu: *Near-Capacity Multi-Functional MIMO Systems: Sphere-Packing, Iterative Detection and Cooperation,* IEEE Press - John Wiley, 2009
- L. Hanzo, J. Akhtman, M. Jiang, L. Wang: *MIMO-OFDM for LTE, WIFI and WIMAX: Coherent versus Non-Coherent and Cooperative Turbo-Transceivers,* John Wiley and IEEE Press, 2010, 608 pages
- L. Hanzo, R.G. Maunder, J. Wang, L-L. Yang: *Near-Capacity Variable-Length Coding,* John Wiley and IEEE Press, 2010, 608 pages, 2011, 494 pages

Acknowledgements

We are indebted to our many colleagues who have enhanced our understanding of the subject. These colleagues and valued friends, too numerous to be mentioned individually, have influenced our views concerning the subject of the book. We thank them for the enlightenment gained from our collaborations on various projects, papers, and books. We are particularly grateful to our academic colleagues Professors Sheng Chen, Mohammed El-Hajjar, Rob Maunder and Lie-Liang Yang as well as to Dr Chao Xu. Finally, we would also like to express our appreciation to our former colleagues Drs. Dimitrios Alanis and Panagiotis Botsinis.

Zunaira Babar, Daryus Chandra, Soon-Xin Ng and Lajos Hanzo
School of Electronics and Computer Science
University of Southampton, UK

Foreword

Many thanks for your interest in this book, valued Colleague!

Naturally, our intention with the book is multifold, but primarily to pave the way for the classical coding community to enter the field of quantum coding.

Quantum error correction codes are expected to play a pivotal role in terms of mitigating the deleterious effects of quantum decoherence, which limits the performance of both quantum computers and quantum-secured communications systems. Indeed, error correction codes are ubiquitous even in less susceptible classical communications and storage systems. Explicitly, all digital wireless systems conceived over the past five decades heavily relied on the potent signal-decontamination capability of classical error correction codes. They are also capable of estimating the system's operating status by monitoring the prevalent error rate and exploit this for the near-instantaneous reconfiguration of the system. Similar benefits may be anticipated also for quantum systems upon harnessing quantum codes, because quantum systems are significantly more susceptible to corruption by the surrounding electromagnetic environment.

A complex-valued qubit may be exposed to a bit-flip, a phase-flip or to a simultaneous bit- and phase-flip. Hence the first ever 1/9-rate quantum code was conceived by Peter Shor by simply exploiting the fact that a 1/3-rate repetition code can separately be applied for guarding against bit- and phase-flips. Calderbank, Steane and Shor later defined the specific conditions under which this appealingly simple principle may be extended to the employment of more sophisticated near-capacity classical codes for conceiving powerful quantum codes. Hence in this self-contained monograph we portray this evolutionary process from the first principles all the way to the design of adaptive-rate near-hashing-bound codes.

Part I provides a gentle introduction to the rudimentary principles, paving the way for readers familiar with classical coding to quantum coding, including the associated quantum and classical coding basics. Part II is dedicated to the family of near-term quantum codes, which do not require a high number of qubits and hence are suitable for near-term intermediate scale (NISC) quantum computers, for example. Finally, Part III elaborates on the design of a suite of sophisticated long-term quantum coding solutions, when having a relatively high number of qubits becomes realistic, as quantum technology matures. A range of adaptive-rate quantum codes are also conceived for practical scenarios of time-variant depolarizing probability.

Wishing you intellectual stimulation, valued Colleague!

Contents

Acknowledgments v

I From Classical to Quantum Codes xiii

List of Acronyms xv

1 Introduction 1
- 1.1 Motivation . . . 1
- 1.2 Historical Overview . . . 5
  - 1.2.1 Quantum Stabilizer Codes . . . 5
  - 1.2.2 Quantum Topological Error Correction Codes . . . 8
  - 1.2.3 Quantum Convolutional Codes . . . 11
  - 1.2.4 Quantum Low Density Parity Check Codes . . . 13
  - 1.2.5 Quantum Turbo Codes . . . 14
  - 1.2.6 Entanglement-Assisted Quantum Codes . . . 15
  - 1.2.7 Protecting Quantum Gates . . . 17
- 1.3 Outline of the Book . . . 19

2 Preliminaries on Quantum Information 27
- 2.1 Introduction . . . 27
- 2.2 A Brief Review of Quantum Information . . . 27
- 2.3 Quantum Information Processing . . . 31
  - 2.3.1 Unitary Transformation . . . 31
    - 2.3.1.1 Pauli Gates . . . 32
    - 2.3.1.2 Hadamard Gate . . . 33
    - 2.3.1.3 Phase Gate . . . 34
    - 2.3.1.4 Controlled-NOT Gate . . . 34
    - 2.3.1.5 Toffoli Gate . . . 35
  - 2.3.2 Quantum Measurement . . . 36
- 2.4 Quantum Decoherence . . . 37

2.4.1 Symmetric Quantum Depolarizing Chanel . . . 42
2.4.2 Asymmetric Quantum Depolarizing Chanel . . . 42
2.4.3 Independent Binary-Symmetric Chanel . . . 43
2.5 No-Cloning Theorem . . . 44
2.6 Quantum Entanglement . . . 45
2.7 Quantum Channels . . . 46
2.8 Summary and Conclusions . . . 49

3 From Classical to Quantum Coding **51**
3.1 Introduction . . . 51
3.2 A Brief Review of Classical Syndrome-based Decoding . . . 52
3.3 A Brief Review of Quantum Stabilizer Codes . . . 57
3.4 Protecting A Single Qubit: Design Examples . . . 61
3.4.1 Classical and Quantum 1/3-rate Repetition Codes . . . 61
3.4.2 Shor's 9-Qubit Code . . . 67
3.4.3 Steane's 7-Qubit Code . . . 69
3.4.4 Laflamme's 5-Qubit Code - The Perfect Code . . . 72
3.5 Summary and Conclusions . . . 74

4 Revisiting Classical Syndrome Decoding **77**
4.1 Introduction . . . 77
4.2 Look-Up Table-Based Syndrome Decoding . . . 80
4.3 Trellis-Based Syndrome Decoding . . . 81
4.3.1 Linear Block Codes . . . 82
4.3.2 Convolutional Codes . . . 85
4.4 Block Syndrome Decoding . . . 91
4.4.1 General Formalism . . . 91
4.4.2 Block Syndrome Decoder for TTCM . . . 92
4.4.2.1 System Model . . . 93
4.4.2.2 Syndrome-Based MAP Decoder . . . 94
4.4.2.3 Error Estimation . . . 95
4.4.2.4 Syndrome-Based Blocking . . . 95
4.5 Results and Discussions . . . 97
4.5.1 Performance of BSD-TTCM over AWGN Channel . . . 97
4.5.2 Performance of BSD-TTCM over Uncorrelated Rayleigh Fading Channel . . . 98
4.5.3 Effect of Frame Length on the Performance of BSD-TTCM . . . 102
4.6 Summary and Conclusions . . . 103

Bibliography **105**

Subject Index **129**

5 Near-Capacity Codes for Entanglement-Aided Classical Communication **129**
5.1 Introduction . . . 129
5.2 Review of the Superdense Coding Protocol . . . 131
5.2.1 2-Qubit Superdense Coding . . . 131
5.2.2 N-Qubit Superdense Coding . . . 133
5.3 Entanglement-Assisted Classical Capacity . . . 134
5.4 Bit-Based Code Structure . . . 136
5.5 Near-Capacity Design . . . 138
5.5.1 EXIT Charts . . . 138
5.5.2 Near-Capacity IRCC-URC-SD Design . . . 139
5.6 Results and Discussions I . . . 143
5.6.1 Performance of IRCC-URC-2SD . . . 143
5.6.2 Performance of IRCC-URC-3SD . . . 146
5.7 Symbol-Based Code Structure . . . 150
5.8 Results and Discussions II . . . 151
5.9 Summary and Conclusions . . . 156

II Near-Term Quantum Codes **159**

6 Quantum Coding Bounds **161**
6.1 Introduction . . . 161
6.2 On Classical to Quantum Coding Bounds . . . 162
6.2.1 Singleton Bound . . . 162
6.2.2 Hamming Bound . . . 163
6.2.3 Gilbert-Varshamov Bound . . . 163
6.3 Quantum Coding Bounds in the Asymptotical Limit . . . 164
6.4 Quantum Coding Bounds on Finite-Length Codes . . . 168
6.5 The Bounds on Entanglement-Assisted Quantum Stabilizer Codes . . . 172
6.6 Summary and Conclusions . . . 176

7 Quantum Topological Error Correction Codes **179**
7.1 Introduction . . . 179
7.2 Classical Topological Error Correction Codes: Design Examples . . . 180
7.3 Quantum Topological Error Correction Codes: Design Examples . . . 189
7.4 Performance of Quantum Topological Error Correction Codes . . . 196
7.4.1 QBER Versus Depolarizing Probability . . . 199
7.4.2 QBER Versus Distance from Hashing Bound . . . 200
7.4.3 Fidelity . . . 203
7.5 Summary and Conclusions . . . 206

Part I

From Classical to Quantum Codes

List of Acronyms

BCH	Bose-Chaudhuri-Hocquenghem
BER	Bit Error Rate
CNOT	Controlled-NOT
CSS	Calderbank-Shor-Steane
DFS	Decoherence-Free Subspace
EA	Entanglement-Assisted
EXIT	EXtrinsic Information Transfer
GV	Gilbert-Varshamov
LDPC	Low-Density Parity-Check
LUT	Look-Up Table
MDS	Maximum Separable Distance
MRRW	McEliece-Rodemich-Rumsey-Welch
PCM	Parity-Check Matrix
QBCH	Quantum Bose-Chaudhuri-Hocquenghem
QBER	QuBit Error Ratio
QCC	Quantum Convolutional Code
QECC	Quantum Error Correction Code
QEDC	Quantum Error Detection Code
QIrCC	Quantum Irregular Convolutional Code
QLDPC	Quantum Low-Density Parity-Check
QPC	Quantum Polar Code
QRM	Quantum Reed-Muller
QRS	Quantum Reed-Solomon
QSBC	Quantum Short-Block Code
QSC	Quantum Stabilizer Code
QTC	Quantum Turbo Code
QTECC	Quantum Topological Error Correction Code
QURC	Quantum Unity-Rate Code
RS	Reed-Solomon
SBC	Short-Block Code
SPC	Single Parity-Check
TECC	Topological Error Correction Code

Chapter 1

Introduction

1.1 Motivation

In 1935, Erwin Schrödinger, an Austrian physicist, proposed a thought experiment to illustrate the interpretation of uncertainty in quantum mechanics. This thought experiment later acquired the fond connotation of the "Schrödinger's Cat" experiment [1, 2]. In this treatise, we will refer to another thought experiment, namely the "Black Box Experiment". Let us assume that we have a black box with a coin spinning in it. It does not have to be a fair coin. Inside the box, we then start spinning the coin. We do not have any prior knowledge about the coin, namely whether the coin is fair or biased and we do not know whether it landed on the face or the tail side when it finally stopped spinning. At this moment, we may say that the coin inside the box is within a *superposition* of two states and has a certain probability for each of the two states. Let us continue the experiment by using several boxes. We may proceed by using two, three, or even an arbitrarily number of $N \in \mathbb{N}$ boxes for this experiment. Similarly, we spin all the N coins simultaneously and close the boxes. This situation may be viewed as having 2^N states, suggesting that we can increase the computational power exponentially in line with 2^N if we can exploit the above-mentioned superposition. Let us now continue the experiment to the following step. Up to this moment, all of the boxes are completely sealed and we have secured the 2^N states of the coins simultaneously. Now, we decide to open all of the boxes at once and observe all the coins. After observing the state of each coin in each box, we are now in the position to observe a single specific state from the full set of 2^n possible states. This illustrates that after lifting the lid of the boxes the quantum states suddenly collapse into a single deterministic classical state due to the action of "observation", which is also often termed in parlance as a "measurement". This property is exploited by quantum computers, which will create a powerful computational tool that is potentially capable of breaking conventional cryptography.

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. This allows the parallel evaluation

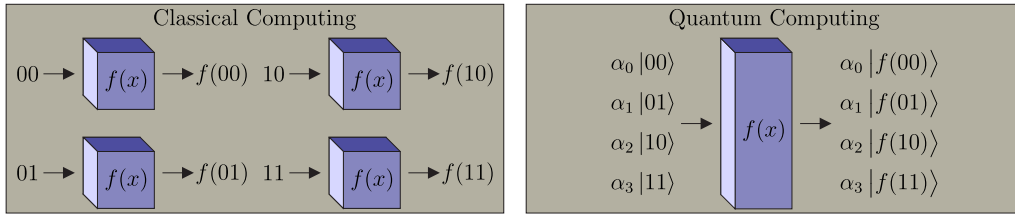


Figure 1.1: Quantum Parallelism: Given a function $f(x)$, which has a regular global structure such that $f(x) : \{0,1\}^2 \rightarrow \{0,1\}^2$, a classical system requires four evaluations to compute $f(x)$ for all possible $x \in \{00, 01, 10, 11\}$. By contrast, a 2-qubit quantum register can be in a superposition of all the four possible states concurrently. This may be written as $|\psi\rangle = \alpha_0 |00\rangle + \alpha_1 |01\rangle + \alpha_2 |10\rangle + \alpha_3 |11\rangle$, where the operation $|\,\rangle$ distinguishes the quantum states. Therefore quantum computing requires only a single classical evaluation/observation 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 [5, 6, 7, 8].

of certain functions with regular global structure at a complexity cost that is equivalent to a single classical evaluation [3, 4], as illustrated in Figure 1.1. Therefore, as exemplified by Shor's factorization algorithm [7] and Grover's search algorithm [8], quantum-based computation is capable of solving certain complex problems at a substantially lower complexity than classical computing.

In 1965, Gordon E. Moore released the general rule of thumb projecting the number of transistors on a single integrated circuit chip based in the semiconductor industry. This notable rule of thumb was later termed as "Moore's Law", which dictates that the number of the transistors on an integrated circuit chip will be doubled every 18 months or two years [10]. This law has maintained its validity over the past few decades, as illustrated in Fig. 1.2, but it was hypothesized that the trend is expected to slow down [11]. As the transistor's size is reduced, we encounter new physical phenomena as we enter the nano-scale world, which can only be described using the postulates of quantum mechanics [12]. We encounter both pros and cons as we embark on this journey into the unknown. Firstly, the ability to create the above-mentioned simultaneous states at any instant lends itself to high-power parallel computing by exploiting the quantum-domain superposition. Secondly, the collapse of quantum superposition into a classical state upon observation potentially allows us to conceive unbreachable communication schemes and detect eavesdropping, which is the main advantage of quantum communications.

The field of theoretical quantum computing was established five decades after the Schrödinger Cat experiment by the renowned physisist Richard Feynman who suggested how to simulate quantum phenomena using quantum computers [13]. Since then, diverse quantum-domain algorithms were invented and indeed have shown that the laws of quantum mechanics will help us to

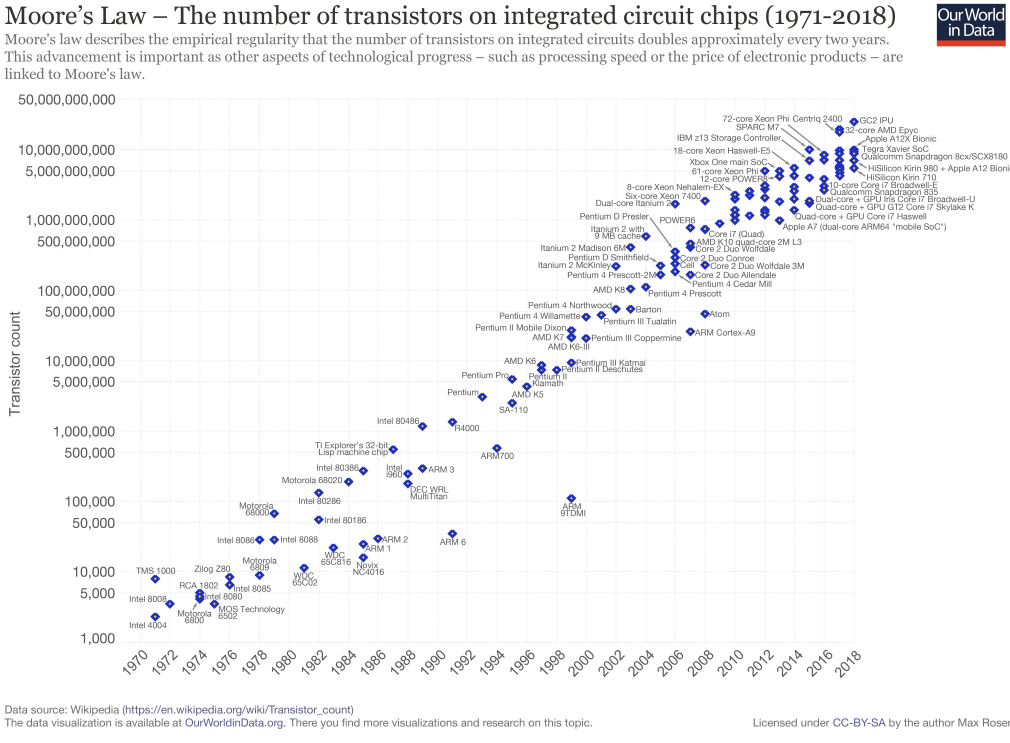


Figure 1.2: The number of transistors on an integrated circuit is doubled every two years since 1971, as predicted by Gordon E. Moore in 1985 [9].

speed up computations in some applications [14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30]. In parallel to the quantum computing developments, attractive quantum-based solutions capable of providing absolute security have also been conceived [31,32,33,34,35,36,37,38,39,40].

Despite the above-mentioned advances, the physical implementation of quantum computers is still far from perfection. Substantial efforts have been invested in building a scalable and reliable quantum computer relying on different solutions. For instance, by using spin electron based techniques [41,42], photonic chips [43,44,45], superconducting qubits [46,47,48], and recently also by using silicon [49,50] as well as microwave trapped-ions techniques have been conceived [51,52]. In order to arrive at the best architecture for quantum computers, the physical implementations of quantum computation have to satisfy the so-called "DiVincenzo's Criteria" [53] described below:

(a) **A scalable physical system with well-characterized qubits**. The elementary unit of information in classical computers is represented by *binary digits* or *bits*. Each bit can only hold a logical value of "0" or "1" at any instant, but not both. By contrast, the elementary unit of information in quantum computers is represented by a *quantum bit* or *qubit*. In quantum mechanics, the state of "0" and "1" are commonly represented using the Dirac notation, i.e. $|0\rangle$ for state "0" and $|1\rangle$ for state "1" [54]. Each qubit can hold a value of $|0\rangle$ or $|1\rangle$ or the superposition of both states. The physical realization of a qubit

should reliably distinguish the state $|0\rangle$ and $|1\rangle$ as well as the superposition of both states. For instance, we can have a two-level quantum system using the up/down spin states of a particle, or the ground and excited states of an atom, or the vertical and horizontal polarization of a single photon.

(b) **The ability to initialize the state of the qubits to a simple fiducial state**. One of the essential requirements in classical computing is to know the initial state of a register before starting the computations. Similar requirements are also applicable in the quantum domain. For instance, to initialize the process of quantum search and quantum factoring algorithms, the quantum registers have to supply a certain number of fresh auxiliary qubits in the state of $|0\rangle$. Similarly, operations such as quantum teleportation, quantum superdense coding, and quantum key distribution (QKD) also require the quantum registers to provide a continuous supply of fresh qubits in a certain superposition state.

(c) **Sufficiently long decoherence times, much longer than the gates' operation time**. In quantum computers, the qubits will be involved in a series of quantum-domain operations to carry out a certain quantum computation or quantum communication tasks. Ideally, a quantum computing algorithm and similarly a quantum communication scheme should be designed by ensuring that the computational process or transmission finishes before the qubits are corrupted by the decoherence phenomenon caused by their undesired coupling with the surrounding environment. However, a long decoherence time does not necessarily mean that the qubits are more reliable. We primarily care about the number of operations that can be completed before the qubits decohere. Hence, we should take into account the gate operation time. The maximal number of reliable operations in quantum computers is defined by the ratio of the qubits' decoherence time to the per-gate operation duration. A quantum solution having a higher maximal number of reliable operations may be more preferable, because as we scale-up the quantum computers, the number of gate operations will increase.

(d) **A universal set of quantum gates**. The power of quantum computers in speeding up some computations hinges on the availability of a universal set of quantum gates. More specifically, a universal set of quantum gates primarily entails the family of gates that can be simulated by classical probabilistic computers in polynomial time, namely the so-called Clifford group defined in [55]. Additionally, there are also non-Clifford gates, which impose higher simulation complexity. Hence, in order to develop fully functioning quantum computers achieving a beneficial quantum speed-up, a set of quantum gates outside the Clifford group also have to be realized.

(e) **A qubit-specific measurement capability**. Eventually, the results from a series of quantum operations carrying out certain computational tasks have to be read out. For this reason, quantum computers require so-called measurement operators, which have to be reliable and capable of operating in various measurement bases.

As for quantum communications, there are two additional criteria, as described below:

(a) **The ability to convert stationary qubits to 'flying' qubits and vice versa**. Numerous applications of quantum computers require the transmission of qubits to a different

location. Hence, the capability of converting the stationary qubits to flying qubits is essential.

(b) **The capability of reliably transmitting flying qubits between specific locations**. To guarantee that the state of the qubits remains intact after their displacement to a different location, their protection against quantum decoherence is required, since we cannot completely isolate the interaction of the qubits with the surrounding environment, even if high-grade electromagnetic shielding and near-absolute-zero temperature are used.

Quantum computers face the same problem as their classical counterparts, namely decoherence. The aforementioned criteria for developing scalable and reliable quantum computers may not be accomplishable if the deleterious effects of quantum decoherence cannot be mitigated. Therefore, it is a challenge to ensure the reliability of quantum computers in the face of ubiquitous quantum decoherence. Quantum error-correction codes (QECC) constitute one of the most popular techniques for tackling the deleterious effects of quantum decoherence. The employment of QECCs may enable quantum computers to achieve high-reliability reproducible operations, in-line with the DiVincenzo's Criteria [56]. Therefore, it might be surmised that one of the key ingredients of realizing reliable quantum computers is the employment of QECCs.

1.2 Historical Overview

1.2.1 Quantum Stabilizer Codes

The concept of protecting the quantum information from decoherence is similar to that of its classical counterpart by attaching redundancy to the information [57], which is then invoked later for error-correction. The quest for finding good QECCs was inspired by Shor, who conceived a 9-qubit code, which is judiciously often referred to as Shor's code [58]. Shor's code encodes a single information qubit, which is also referred to as the "logical qubit", into nine encoded qubits or "physical qubits". Shor's 9-qubit code is capable of protecting the nine physical qubits from any type of single-qubit error. Following the discovery of Shor's code, another QECC scheme, namely Steane's code, was proposed in [59]. The latter is capable of protecting the physical qubits from any single-qubit error by encoding a single logical qubit into seven physical qubits, instead of nine qubits. The question concerning the minimum number of physical qubits required to protect them from any type of single-qubit error was answered by Laflamme *et al.*, who proposed the 5-qubit quantum code having a quantum coding rate of 1/5 [60]. This 5-qubit may also be referred to as Laflamme's "perfect code" since the code construction achieves the quantum Hamming bound and the quantum Singleton bound of binary codes. This is the upper bound of the achievable quantum coding rate, when aiming for correcting a single qubit [61,62]. To elaborate briefly, each of the 5 complex-valued qubits may have a bit-flip, a phase-flip as well as a joint bit-flip and phase-flip error. The bit-flip and phase-flip errors are shown in the stylised illustration of Figure 1.3.

Hence in Laflamme's code we have $3 \cdot 5 = 15$ legitimate error events plus the error-free scenario. Hence we have to attach 4 redundant qubits to the original logical qubit for distinguising

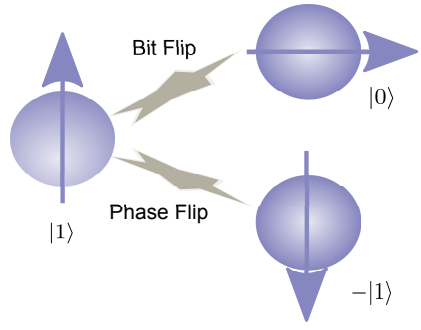


Figure 1.3: Quantum decoherence characterized by bit-flips and phase-flips. *The vertical polarization represents the state* $|1\rangle$*, while the horizontal polarization represents the state* $|0\rangle$.

the above-mentioned 16 possible error-scenarios. In exchange for this undesirably low code-rate we have a powerful error correction capability, since an arbitrary error can be eliminated in a set of five qubits, ie. when the error rate imposed by decoherence is as high as 20 %. Suffice to say in closing that it is perfectly feasible to create high-rate codes, which would however correct a lower error rate, as it will transpire throughout the rest of this monograph.

The field of QECCs entered its golden age following the invention of quantum stabilizer codes (QSCs) [63, 64]. The QSC paradigm allows us to transform the classical error correction codes into their quantum counterparts. The QSCs must be constructed for circumventing the problem of estimating both the number and the position of quantum-domain errors imposed by quantum decoherence without observing the actual quantum states, since observing the quantum states would collapse the qubits into classical bits. This extremely beneficial error estimation technique was achieved by introducing the syndrome-measurement based approach [63, 64]. In classical error correction codes, the syndrome-measurement based approach has been widely exploited as a popular error detection and correction procedure. Therefore, the formulation of QSCs expanded the search space of good QECCs to a broader horizon. This new paradigm of incorporating the classical to quantum isomorphisms, led to the transformation of classical codes to their quantum domain dual pairs, such as quantum Bose-Chaudhuri-Hocquenghem (QBCH) codes [65, 84], quantum Reed-Solomon (QRS) codes [66], quantum Reed-Muller (QRM) codes [67], quantum convolutional codes (QCCs) [70, 85], quantum low-density parity-check (QLDPC) codes [73], quantum turbo codes (QTCs) [78], and quantum polar codes (QPCs) [82]. A timeline that portrays the milestones of QSCs, at a glance is depicted in Fig. 1.4. Although the QSC formulation creates an important class of QECCs, we note that there are also other classes of QECCs beside the QSCs, such as the class of decoherence-free subspace (DFS) codes [86, 87, 88]. DFS codes can be viewed as a family of passive QECCs, while the QSCs constitute a specific example of the active ones. To elaborate a little further, DFS codes constitute a highly degenerate class of QECCs, which rely on the fact that the error patterns may preserve the state of physical qubits and therefore they do not necessarily require a recovery procedure. Due to their strong reliance on the degeneracy property exhibited by the set of QECCs that do not have a classical counterpart, the class of DFS codes bears no resemblance to any classical error



Figure 1.4: Timeline of important milestones in the QECC field, specifically in the development of QSCs. The code construction is highlighted with **bold** fonts, while the associated code type is printed in *italics*.

correction codes. Therefore, in this treatise, we focus our discussions purely on QSCs, which exhibit strong analogies with classical error correction codes.

Even though intensive research efforts have been invested in exploring the field of QSCs, one of the mysteries remains unresolved. Since the development of the first QSC, one of the open problems has been how to determine the realistically achievable size of the codebook $|\mathcal{C}| = 2^k$, given the number of physical qubits n, the minimum distance of d, and the quantum coding rate of $r_Q = k/n$, where k denotes the number of logical qubits. The minimum distance d of the legitimate codewords is the parameter that defines the error correction capability of the corresponding code. The complete formulation of the realistically achievable minimum distance d,

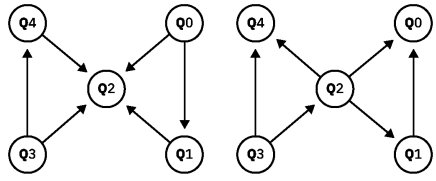
given the number of physical qubits n and the quantum coding rate r_Q is unknown at the time of writing, but several theoretical lower and upper bounds can be found in the literature [62,89,90]. Naturally, finding code constructions associated with growing minimum distances upon reducing the coding rate is desirable, since an increased minimum distance improves the reliability of quantum computation [56,91,92,93,94]. The trade-off between the quantum coding rate and the minimum distance as well as the codeword length is widely recognized, but the achievable minimum distance d of a quantum code given the quantum coding rate r_Q and codeword length n remains unresolved.

For example, for a given codeword length of $n = 128$ and quantum coding rate of $r_Q = 1/2$, the achievable minimum distance is loosely bounded by $11 < d < 22$, while for $n = 1024$ and $r_Q = 1/2$, the achievable minimum distance is bounded by $78 < d < 157$. Naturally, having such a wide range of minimum distances is undesirable. For binary classical codes, this problem has been circumvented by the closed-form approximation proposed by Akhtman *et al.* [95].

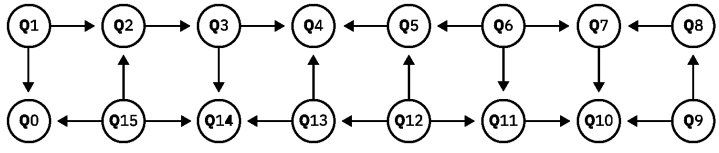
The challenge of creating the quantum counterpart of error correction codes lies in the fact that the QSC constructions have to mitigate not only bit-flip errors but also phase-flip errors or both bit-flip and phase-flip errors. Based on how we mitigate those different types of errors, we can simply categorize QSCs as belonging to the class of Calderbank-Shor-Steane (CSS) codes [59, 90,96] or to the complementary class of non-CSS codes [64]. The CSS codes handle qubit errors by treating bit-flip errors and phase-flip errors as separate entities. By contrast, the class of non-CSS codes treat both bit-flip errors and phase-flip errors simultaneously. Since the CSS codes treat the bit-flip and phase-flip error correction procedures separately, in general, they exhibit a lower coding rate than their non-CSS counterparts having the same coderate. Furthermore, if we also consider the presence of quantum entanglement, we may conceive more powerful quantum code constructions as discussed in [68,69]. To elaborate, the family of entanglement-assisted quantum stabilizer codes (EA-QSCs) is capable of operating at a higher quantum coding rate than the unassisted QSC constructions at the same error correction capability, provided that error-free maximally-entangled qubits have already been preshared [68, 69]. This presharing operation may be carried out, when the circuits are not actively harnessed for carrying out urgent impending operations.

1.2.2 Quantum Topological Error Correction Codes

We have established that one of the essential prerequisites of constructing quantum computers is the employment of QECCs for ensuring that the computers operate reliably by mitigating the deleterious effects of quantum decoherence [53,56,97]. However, the laws of quantum mechanics prevent us from transplanting classical error correction codes directly into the quantum domain. To circumvent the constraints imposed by the nature of quantum physics, the notion of quantum stabilizer codes (QSCs) emerged [63,64,98]. The invention of QECCs and specifically the QSC formalism did not immediately eradicate all of the obstacles of developing reliable quantum computers. Employing the QSCs requires redundancy in the form of *auxiliary* quantum bits (qubits) to encode the logical qubits onto physical qubits. The redundant qubits are then exploited during the error correction. These operations rely on the quantum encoder and decoder



(a) 5 qubits (ibmqx2). (b) 5 qubits (ibmqx4).



(c) 16 qubits (ibmqx5).

Figure 1.5: The qubit arrangement of IBM's superconducting quantum computers. The circles represent the qubits, while the arrows represent the possible qubit interactions within the computers [99].

circuits constructed from quantum gates. Therefore, the circuit-based implementation of a QSC itself has to be fault-tolerant to guarantee that the QSC circuit does not proliferate the existing errors.

The notion of QSC triggered numerous discoveries in the domain of QECCs, which are inspired by classical error correction codes. Essentially, QSCs represent the quantum-domain version of the classical syndrome decoding based error correction codes. Since the concept of utilizing the syndrome values for error correction is widely exploited in the classical domain, diverse classical error correction codes can be conveniently 'quantumized'. Consequently, we can find in the literature the quantum version of error correction codes based on algebraic formalisms such as those of the Bose-Chaudhuri-Hocquenghem (BCH) codes [65] and of Reed-Solomon (RS) codes [66], quantum codes based on a conventional trellis structure such as convolutional codes [70] and turbo codes [78, 81], quantum codes based on bipartite graphs, such as low-density parity-check (LDPC) codes [72, 73, 74, 75, 77], as well as quantum codes based on channel polarization, such as polar codes [82, 83].

Apart from exploiting the above isomorphism, there are also significant contributions on directly developing code constructions solely based on the pure quantum topology and homology, as exemplified by the family of so-called toric codes [100, 101, 102], surface codes [103, 104], colour codes [105], cubic codes [106], hyperbolic surface codes [107, 108], hyperbolic color codes [109], hypergraph product codes [110, 111, 112] and homological product codes [113]. Unfortunately, this concept has not been widely explored in the classical domain. By contrast, in the quantum domain, having a code construction relying on the physical configuration of qubits is highly desirable for the conception of low-complexity high-reliability quantum computers.

For instance, this strategy has been deployed for developing IBM's superconducting quantum computers, as shown in Fig. 1.5. From this figure, we can see the qubit arrangement of the three prototypes of IBM's quantum computer - which can be viewed online - namely the ibmqx2,

ibmqx4, and ibmqx5 configurations [99]. The first two of the quantum computers are the 5-qubit quantum computers, while the third one is a 16-qubit quantum computer. The circles in Fig. 1.5 represent the qubits, while the arrows represent all their two-qubit interactions. It can be seen that the existing architectures impose a limitation, namely the two-qubit interactions can be only performed between the neighbouring qubits. Even though this particular limitation potentially imposes additional challenges, when it comes to QSCs deployment, the stabilizer effect can still be achieved by the corresponding qubit arrangement by invoking the family of QTECCs. A timeline that portrays the milestones in the evolution of QTECCs is portrayed at a glance in Fig. 1.6.



Figure 1.6: Timeline of important milestones in the area of QTECCs. The code construction is highlighted with **bold** while the associated code type is marked in *italics*.

Again, Shor's 9-qubit code protects 9 physical qubits from any type of single-qubit error, namely bit-flip (**X**), phase-flip (**Z**), as well as from simultaneous bit and phase-flip (**Y**). Furthermore, as also alluded to above, not long after the discovery of the first QECCs, Steane invented the 7-qubit code, which was followed by Laflamme's perfect 5-qubit code [59, 60]. However, the construction of these codes does not naturally exhibit inherent fault-tolerance. Briefly, a QSC

is said to be fault-tolerant if the circuit-based implementation of the QSC does not introduce more errors than the error correction capability of the QSC, when the quantum gates required for implementing the QSC are imperfect. The quantum circuit based implementation of these pioneering QSCs always involves a high number of qubit interactions within the codeword of physical qubits. As a consequence, an error caused by an imperfect gate potentially propagates to other qubits and instead of being eliminated, the deleterious effects of quantum decoherence are further aggravated.

At the current development stage of quantum computers, a QSC exhibiting fault-tolerance is more favourable, since the reliability of quantum gates is substantially lower than that of classical logic gates. The employment of QSCs is expected to mitigate the deleterious effect of the imperfect quantum gates. However, the QSC circuit itself is prone to decoherence. For the sake of constructing a fault-tolerant QSC scheme, the notion of quantum topological error correction codes (QTECCs) was proposed [100, 101]. The formulation of QTECCs offers substantial fault-tolerance improvements because they exhibit an increased minimum distance upon increasing the codeword length. Furthermore, they rely on localized of stabilizer measurements. Nonetheless, one of the substantial drawbacks of QTECCs is their low quantum coding rate. More specifically, the quantum coding rate of QTECCs tends to zero for an asymptotically long codeword.

Another class of codes which are considered to be fault-tolerant QSCs is constituted by the family of QLDPC codes. The QLDPC codes inherit the property of fault-tolerance due to having a sparse parity-check matrix, which guarantees the limited interaction of the qubits within the same block of codewords. Even though QLDPC codes can achieve a good performance at a relatively high quantum coding rate, the construction of QLDPC has a bounded minimum distance [73, 112] and they tend to perform best for long codeword. Even though intensive research efforts have been invested in exploring the QECC field, the fundamental trade-off between the quantum coding rate and the minimum distance remains unresolved. Having a high minimum distance is important for guaranteeing a low error floor and robust fault-tolerance. However, it is unfeasible to construct a QSC exhibiting high minimum distance without unduly reducing the quantum coding rate or increasing the number of physical qubits. Indeed, this is only a specific example of the quantum coding rate versus minimum distance trade-off. Furthermore, the quantum coding rate versus minimum distance trade-off is not the only one involved in designing the QSCs, as seen in Fig. 1.7. The discussion of these intricately interlinked aspects will prevade the rest of the book.

1.2.3 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 [118], Chau conceived the first QCC in [119]. He also generalized the classical Viterbi decoding algorithm for the class of quantum codes in [120], but he overlooked some crucial encoding and decoding aspects. Later, Ollivier *et al.* [121, 122] 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

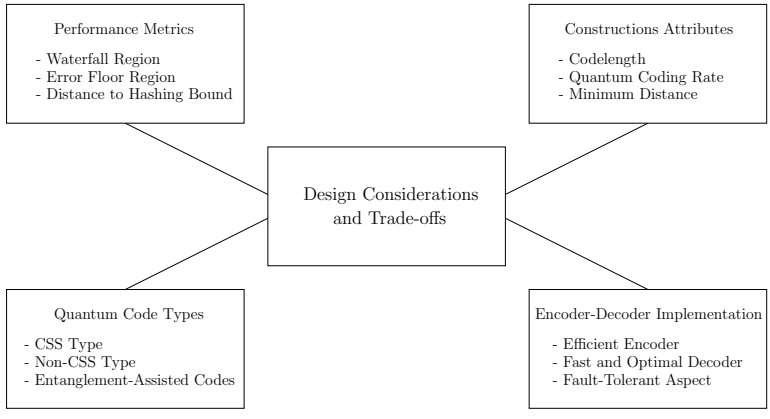


Figure 1.7: The conflicting design factors related to QECC code design.

Author(s)	**Coding Efficiency**	**Decoding Complexity**
Ollivier and Tillich [121, 122]	Low	Moderate
Almeida and Palazzo [124]	Moderate	Moderate
Forney *et al.* [125, 126]	High	Low

Table 1.1: Comparison of the Quantum Convolutional Code (QCC) structures.

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 [123]. Pursuing this line of research, Almeida *et al.* [124] 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 [121, 122]. 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.* [125, 126] designed rate-$(n-2)/n$ QCCs comparable to their classical counterparts, thus providing higher coding efficiencies than the comparable block codes. Forney *et al.* [125, 126] 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.* [125, 126] 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 1.1. Furthermore, in the spirit of finding new constructions for QCCs, Grassl *et al.* [127, 128] constructed QCCs using the classical self-orthogonal product codes, while Aly *et al.* explored various algebraic constructions in [129] and [130]. Particularly, the QCCs

Year	Author(s)	Contribution
1998	Chau [119]	The first QCCs were developed. Unfortunately, some important encoding/decoding aspects were ignored.
1999	Chau [120]	Classical Viterbi decoding algorithm was generalized to the quantum domain. However, similar to [119], some crucial encoding/decoding aspects were overlooked.
2003	Ollivier and Tillich [121, 122]	Stabilizer-based convolutional codes and their maximum likelihood decoding using the Viterbi algorithm were revisited to overcome the deficiencies of [119, 120]. Failed to provide better performance or decoding complexity than the comparable block codes.
2004	Almeida and Palazzo [124]	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.* [125, 126]	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 [127, 128]	Conceived a new construction for QCCs from the classical self-orthogonal product codes.
2007	Aly *et al.* [129]	Algebraic QCCs dervied from BCH codes.
2008	Aly *et al.* [130]	Algebraic QCCs constructed from Reed-Solomon and Reed-Muller Codes.
2013	Pelchat and Poulin [131]	Degenerate Viterbi decoding was conceived, which runs the MAP algorithm over the equivalent classes of degenerate errors, thereby improving the performance.

Table 1.2: Major contributions to the development of Quantum Convolutional Codes (QCCs).

of [129] were derived from classical BCH codes, while the QCCs of [130] 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 [131], which runs the Maximum *A Posteriori* (MAP) algorithm [132] over the equivalent classes of degenerate errors, thereby improving the attainable performance. The major contributions to the development of QCCs are summarized in Table 1.2.

1.2.4 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 [133], 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 [134] 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.* [135] 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 [136], which were further investigated by Couvreur *et al.* in [137,138].

Aly *et al.* contributed to these developments by constructing dual-containing QLDPC codes from finite geometries in [139], while Djordjevic exploited the Balanced Incomplete Block Designs (BIBDs) in [140], albeit neither of these provided any gain over Mackay's bicycle codes. Lou *et al.* [141,142] 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.* [143] 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 [143] was extended to non-binary QLDPC codes in [144,145], 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 [146]. While all the aforementioned QLDPC constructions were CSS-based, Camara *et al.* [147] were the first authors to conceive non-CSS QLDPC codes. Later, Tan *et al.* [148] 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[1] were derived from Kitaev's construction in [149,150,151]. 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 [152], which were further improved in [153]. The major contributions made in the context of QLDPC codes are summarized in Table 1.4, while the most promising QLDPC construction methods are compared in Table 1.3[2].

1.2.5 Quantum Turbo Codes

Pursuing further the direction of iterative code structures, Poulin *et al.* conceived QTCs in [154, 155], 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 [154,155,156] 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,

[1] Topological code structures are beyond the scope of this book.

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

Code Construction	Short Cycles	Minimum Distance	Delay	Decoding Complexity
Bicycle codes [135]	Yes	Upper Bounded	Standard	Standard
Cayley-graph based codes [136, 137, 138]	Yes	Increases with the code length	Standard	Increases with the code length
LDGM-based codes [141, 142]	Yes	Upper Bounded	Standard	High
Non-binary quasi-cyclic codes [144, 145]	No	Upper Bounded	Standard	High
Spatially-coupled quasi-cyclic codes [146]	No	Upper Bounded	High	High

Table 1.3: Comparison of the Quantum Low Density Parity Check (QLDPC) code structures.

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 [154, 155] had a bounded minimum distance. The QBER performance curves of the QTCs conceived in [154, 155] also failed to match the classical turbo codes. This issue was dealt with in [157], where the quantum turbo decoding algorithm of [155] 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 1.4.

1.2.6 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 [158, 159, 160, 161] 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 [162], EA-QCC [163], EA-QTC [164, 157] and EA-polar codes [165]. In [164, 157], 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 1.8.

Year	Author(s)	Code Type	Contribution
2001	Postol [134]	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.* [135]	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.* [141, 142]	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 [136]	Dual	Cayley graph-based QLDPC codes were proposed, which had numerous length-4 cycles.
2007	Camara *et al.* [147]	Non-CSS	QLDPC codes derived from classical self-orthogonal quaternary LDPC codes were conceived, which failed to outperform MacKay's bicycle codes.
2007	Hagiwara *et al.* [143]	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 [135].
2008	Aly *et al.* [139]	Dual	QLDPC codes were constructed from finite geometries, which failed to outperform Mackay's bicycle codes.
2008	Djordjevic [140]	Dual	BIBDs were exploited to design QLDPC codes, which failed to outperform Mackay's bicycle codes.
2010	Tan *et al.* [148]	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.* [137, 138]	Dual	Cayley graph-based QLDPC codes of [136] 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.
2011	Kasai [144, 145]	Non-dual	Quasi-cyclic QLDPC codes of [143] 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.
2011	Hagiwara *et al.* [146]	Non-dual	Spatially-coupled QC-QLDPC codes were developed, which outperformed the 'non-coupled' design of [143] at the cost of a small coding rate loss. Performance was similar to that of [144, 145], but larger block lengths were required.
2008	Poulin *et al.* [152]		Heuristic methods were developed to alleviate the performance degradation caused by unavoidable length-4 cycles and symmetric degeneracy error.
2012	Wang *et al.* [153]		Feedback mechanism was introduced in the context of the heuristic methods of [152] to further improve the performance.
2008	Poulin *et al.* [154, 155]	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.
2014	Wilde *et al.* [157]		The iterative decoding algorithm of [154, 155] 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 1.4: Major contributions to the development of iterative quantum codes.The code types 'dual-containing CSS' and 'non-dual-containing CSS' are abbreviated as 'dual' and 'non-dual', respectively.

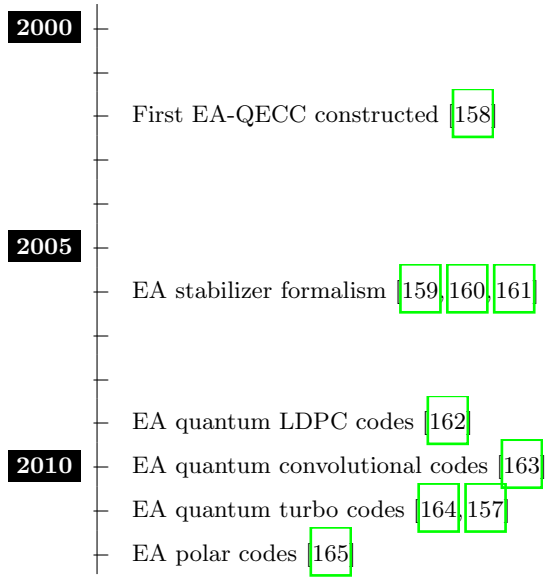


Figure 1.8: Major milestones achieved in the history of entanglement-assisted quantum error correction codes.

1.2.7 Protecting Quantum Gates

Although the field of QECCs benefitted from a rapid pace of development, because under certain conditions we can transform various classes of powerful classical error correction codes into their quantum counterparts, several challenges remain, hindering the immediate employment of these powerful QSCs in quantum computers. Firstly, the reliability of the state-of-the-art quantum gates is still significantly lower compared to classical gates. For example, the reliability of a two-qubit quantum gate is between $90.00\% - 99.90\%$ across the various technology platforms, such as spin electronics, photonics, superconducting, trapped-ion, and silicon solutions [41, 42, 43, 44, 46, 48, 49, 50, 166]. Similar to the classical domain, invoking an error correction code within a quantum computer requires additional components. However, adding components for error correction also implies that we unavoidably introduce an additional source of decoherence into the quantum computers, since the encoder and decoder of QSCs are also composed of quantum gates.

Secondly, the powerful QSCs such as QTCs, QPCs, and QLDPC codes require long codewords in order to operate close to the quantum Hashing bound. In other words, they require a very high number of physical quantum bits (qubits) to correct numerous errors. Additionally, the qubits have a relatively short coherence time [167], and hence, the error correction procedure has to be completed before the ensemble of the qubits starts decohering. Consequently, utilizing QSCs having a high number of qubits for correcting many errors has the potential threat of encountering an avalanche of more erroneous qubits before the error correction procedure is even completed. Thirdly, the state-of-the-art architecture of quantum computers imposes an additional challenge, where the interactions among the qubits are ideally limited to the nearest

neighbour qubits, which can be arranged by introducing a lattice-based topological architecture. However, the aforementioned challenges impose limitations on creating a fault-tolerant error correction architecture.

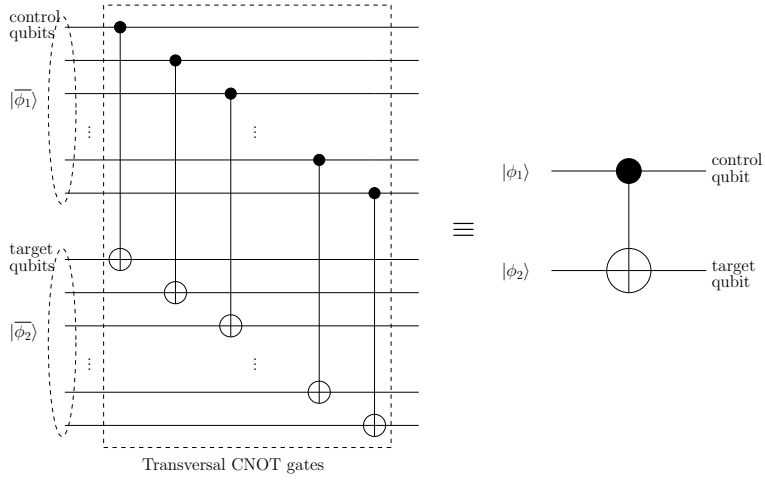


Figure 1.9: Under a certain formulation, a set of less-reliable identical quantum gates in transversal configuration can be used for conceiving a more reliable quantum gate.

The quest for creating fault-tolerant gates was inspired, when the notion of transversal configuration was introduced for quantum gates [56, 93]. Briefly, the concept of transversal gates relies on a parallel set of identical quantum gates invoked for carrying out the operation of a single quantum gate as illustrated in Fig. 1.9. At the right of the figure we can see a controlled NOT (CNOT) gate. This gate does not affect the target qubit, if the control bit is 0, otherwise it flips the target qubit. By contrast, the subfigure at the left is a stylized portrayal of this CNOT gate, where the multiple control and target qubits represent protected operands.

The fact that transversal quantum Clifford gates are suitable for combination with the stabilizer formalism creates an opportunity for employing a wide range of QSCs for protecting transversal quantum gates. However, the challenges we have described earlier suggest that the less populous family of QTECCs is the most suitable candidate for protecting the transversal quantum gates. Again, the motivation behind combining the QTECCs with the transversal configuration of quantum gates is that their benefits conveniently complement each other. More explicitly, the QTECCs provide localized stabilizer measurements, which consequently has the benefit of a constant number of qubit interactions, as we increase the number of physical qubits harnessed. Thus, the benefits provided by the topologically inspired stabilizer formalism will not be affected by the transversal implementation of quantum gates. Hence, the localized action of stabilizer operators amongst the adjacent qubits - which offers fault-tolerance - is still preserved even after the desired quantum operation has been carried out by the transversal quantum gates.

But again, the amalgamation of the QSCs and the transversal quantum gates can only be implemented for quantum Clifford gates, rather than for the entire wider family of universal quantum gates. To achieve the universality of quantum computation, a set of fault-tolerant non-Clifford quantum gates are also required. Fortunately, this wider set may also be created

by using QSCs by harnessing a method referred to as magic state distillation [168], which is however beyond the scope of this work. However, protecting Clifford gates is of significant importance, since the development of large-scale quantum computers relies heavily on improving their fidelity.

The threshold theorem defined in [169] was introduced for demonstrating that a quantum computation task subject to a vanishingly low qubit error ratio (QBER) can be carried out with the aid of QECCs even when relying on realistic error-prone quantum gates, provided that the error rate imposed by the quantum gates is below a certain threshold. Since the error-prone quantum gates may also provide error-prone stabilizer measurements, typically repeated stabilizer measurements are required for reliably concluding the error correction procedure. The number of stabilizer measurements required for making a high-fidelity observation tends to grow as we increase the number of physical qubits utilized for QECCs. This specific problem has led to the emergence of the so-called 'single-shot' QSCs [170,171,172,173]. Assuming that the syndrome values acquired from the syndrome measurements are not reliable, we can still achieve a vanishingly low QBER for a specific quantum computation task, given that we only perform a single stabilizer measurement for each stabilizer operator. However, beneficial QSC constructions have to exhibit a commensurately increased minimum distance as a function of the number of physical qubits. Unfortunately, the quantum coding rate of the two-dimensional QTECCs tends to zero as the codeword length increased [174,175,176]. We have to mention that reliably observing the values from the stabilizer measurements is also of current research interest, which is highly relevant for the study of single-shot QSCs [177,178,179,180]. Therefore, we can ask a judicious question: "Can we still utilize the two-dimensional QTECCs for fault-tolerant quantum computation, when relying only on a single stabilizer measurement for each of the stabilizer operators?" Arguably, the answer is yes, although certain conditions should be fulfilled before we can guarantee that the QSCs can offer substantial reliability improvements.

1.3 Outline of the Book

Part I of the book is constituted by Chapters 1 – 5 and it is dedicated to paving the way **from classical to quantum coding**. This part is organized as follows.

(a) **Chapter 2: Preliminaries on Quantum Information**

In **Chapter 2**, we will provide a rudimentary introduction to quantum information processing. We commence with the definition of the fundamental unit of quantum information constituted by the quantum bit (qubit) in Section 2.2. This is followed by a brief introduction to quantum information processing, including the reversible unitary transformations and the irreversible quantum measurement operation in Section 2.3. The various quantum decoherence models used are elaborated on in Section 2.4. In Section 2.5, we present the *no-cloning theorem*, while in Section 2.6, we highlight the concept of *quantum entanglement*.

(b) **Chapter 3: From Classical to Quantum Coding**

We continue in this chapter by presenting the duality of classical and quantum error correc-

tion codes with the aid of QSC constructions. Our objective is to highlight the similarities between the classical and quantum domain as well as to demonstrate how to transplant the well-known syndrome-based classical decoding concept into the quantum error correction codes. We commence with a brief review of classical syndrome-based decoding in Section 3.2. In Section 3.3, we present the similarities between the classical syndrome-based decoding and the quantum stabilizer codes (QSCs). Finally, in Section 3.4, we provide detailed examples of the QSC constructions protecting a single qubit, namely a 1/3-rate quantum repetition code, Shor's 9-qubit code, Steane's 7-qubit code, and the Laflamme's perfect 5-qubit code.

(c) **Chapter 4: Revisiting Classical Syndrome Decoding**

In this chapter we discuss the popular classical syndrome decoding techniques designed for classical channels. We commence our discussion with the conceptually simplest LUT-based syndrome decoding in Section 4.2, while Section 4.3 details the construction of the syndrome-based error trellis constructed for linear block codes and convolutional codes. Finally, in Section 4.4, we detail a Block Syndrome Decoding (BSD) technique designed for reducing the decoding complexity. In particular, we conceive a syndrome-based block decoder for classical Turbo Trellis Coded Modulation (TTCM) schemes.

(d) **Chapter 5: Near-Capacity Code Designs for Entanglement-Assisted Classical Communication**

In this chapter we invoke EXtrinsic Information Transfer (EXIT) chart aided near-capacity classical code designs conceived for reliable transmission of classical bits over quantum communication channels. More specifically, we focus our attention on the entanglement-assisted transmission of classical information over quantum channels[3], which is achieved with the aid of the SuperDense (SD) coding protocol. We commence by reviewing the SD protocol in Section 5.2, which is in essence the 'Bit $\rightarrow$ Qubit Mapper'. We next characterize the associated capacity in Section 5.3. In Section 5.4, 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 5.5, where the IRCC is optimized for achieving a near-capacity performance. Our bit-based code structure of Section 5.4 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 5.7, which employs a single-component Convolutional Code (CC) and a symbol interleaver in contrast to the IRCC and bit interleaver of Section 5.7.

Part II of the book is represented by Chapters 6 – 9 and it is focused on relatively low-complexity quantum codes requiring a limited number of Qubits. Hence these codes may be viewed as **near-term coding** solutions. This part of the book is organized as follows.

(a) **Chapter 6: Quantum Coding Bounds**

[3]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.

In this chapter we investigate the trade-off between the quantum coding rate versus the minimum distance of QSCs. We commence with the survey of the existing quantum coding bounds in the literature. This is followed by our proposal of a simple and invertible closed-form approximation for determining the realistically achievable minimum distance, given the quantum coding rate of both idealized infinite-length and practical finite-length codewords. Specifically, in Section 6.2, we survey the existing quantum coding bounds and derive some bounds by exploiting the classical-to-quantum isomorphism. These discussions are followed by our proposed closed-form approximation for the idealized asymptotical limit of having an infinite-length codeword in Section 6.3. Since the asymptotical limit has a little relevance for practical implementations, we also proposed an approximate formula for finite-length codewords. In order to unify our quantum coding bound formulation for both the entanglement-assisted QSCs and the unassisted QSCs dispensing with entanglement, we derive a closed-form approximation for arbitrarily entangled QSCs in Section 6.5.

(b) **Chapter 7: Quantum Topological Error Correction Codes**

In this chapter we continue our discussions by the detailed construction of classical topological error correction codes (TECCs) and their quantum-domain dual pairs, namely of the family of quantum topological error correction codes (QTECCs). We carry out a detailed parametric study and derive the QBER upper bound expression. Explicitly, in Section 7.2, we commence with design examples of classical TECCs to pave the way for delving into the quantum domain, while in Section 7.3 we detail the corresponding QSC design examples of QTECCs. We continue by characterizing the performance of QTECCs in the context of the popular quantum depolarizing channel in terms of their QBER, their distance from Hashing bound, and their fidelity in Section 7.4.

(c) **Chapter ??: Protecting Quantum Gates Using Quantum Topological Error Correction Codes**

We then concentrate our attention on the general framework of protecting quantum gates using QSCs. Here, we consider the amalgamation of the transversal configuration of quantum Clifford gates and the QTECCs. This combination has been opted for because it retains the desirable properties of resulting in fault-tolerant QECCs, as a joint benefit of stabilizer preservation and localized stabilizer measurements. First, we proceed with the formulation of our framework in Section **??**. This is followed by the design examples of QSC-protected Hadamard and CNOT gates in Section **??**, where we invoke a simple quantum repetition code. In order to evaluate the performance of our proposed framework, in Section **??**, we present the decoherence model utilized in our simulations. Then in Section **??**, we quantify the performance of QTECC-protected transversal Hadamard gates and CNOT gates both in terms of their QBER and fidelity along with the derivation of the upper and lower bound of the attainable analytical QBER performance in the face of quantum depolarizing channel.

(d) **Chapter ??: Universal Decoding of Quantum BCH and Polar Codes via Classical Guesswork**

In this chapter a universal decoding scheme is conceived for quantum stabilizer codes (QSCs) by appropriately adapting the 'guessing random additive noise decoding' (GRAND)

philosophy of classical domain codes. In the spirit of universality, we demonstrate that the generalized quantum decoder conceived is eminently suitable for different QSC decoding paradigms, namely for both stabilizer-measurement-based as well as the inverse-encoder-based decoding of diverse codes. We then harness the resultant decoder for both quantum Bose-Chaudhuri-Hocquenghem (BCH) codes and quantum polar codes and quantify both their quantum block error rate (QBLER), and QBLER per logical qubits as well as their decoding complexity. Furthermore, we provide a parametric study of the associated design trade-offs and offer design guideline for the implementation of GRAND-based QSC decoders.

Finally, **Part III of the book – namely Chapters 10 to 14** – delves into the design of more **advanced quantum coding** solutions of the future, when a high number of qubits becomes available. This part is outlined as follows.

(a) **Chapter ??: Revisiting the Classical to Quantum Coding Evolution**

As our discussions deepen, in this chapter we revisit the classical to quantum isomorphism in more detail. In Section **??**, we review the family of classical linear block codes. We next discuss the class of QSCs in Section **??**, which are derived from the classical linear block codes of Section **??**. In particular, we highlight the underlying quantum to classical isomorphism, which forms the basis for morphing arbitrary classical codes into the quantum domain. We then extend our discussions to the construction of QCCs from the CCCs in Section **??**, while Section **??** presents EA-QSCs, which facilitate the design of quantum codes from arbitrary classical codes without imposing any stringent requirements.

(b) **Chapter ??: Near-Hashing-Bound Concatenated Quantum Codes**

Pursuing further the design of QECCs, in this chapter we will construct near-Hashing-bound QECCs. We will commence our discourse by laying out the design objectives in Section **??**. Section **??** 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 **??**. Finally, in Section **??**, we extend the application of classical nonbinary EXIT charts to the circuit-based syndrome decoder of QTCs for approaching the Hashing bound[4]. 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 **??**, which constitutes the outer component of a concatenated quantum code.

(c) **Chapter ??: Near-Hashing-Bound Quantum Turbo Short-Block Codes**

In this chapter we set out to conceive the concept of near-hashing bound quantum turbo short-block codes relying on different-rate quantum encoders for combatting diverse quantum depolarizing probabilities. More explicitly, this multiple-rate scheme was conceived by concatenating quantum short-block codes (QSBCs) as the outer codes with a quantum unity-rate code (QURC) as the inner code, which we refer to as the QSBC-QURC construction. In contrast to two-dimensional QTECCs, whose quantum coding rate tends to zero for long codewords, the resultant QSBC-QURC scheme exhibits a relatively high

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

quantum coding rate. Explicitly in Section **??**, we present the general formulation of QSBCs in terms of their code construction, quantum encoder, and stabilizer measurement. This is followed by Section **??**, where we propose a novel family of serially-concatenated QTCs by utilizing QSBCs as the outer codes and a QURC as the inner code, which we refer to as the QSBC-QURC scheme. We analyze the convergence behaviour of our iterative-decoding-aided QSBC-QURC scheme using extrinsic information transfer (EXIT) charts and evaluate its QBER and goodput in Section **??**.

(d) **Chapter ??: EXIT-Chart Aided Design of Irregular Multiple-Rate Quantum Turbo Block Codes**

This chaper is dedicated to the EXIT-chart aided design of so-called irregular quantum turbo short-block codes, which rely on multiple-rate quantum short-block codes (MR-QSBCs) as the outer codes and a quantum unity-rate code (QURC) as the inner code. The proposed design is denoted as MR-QSBC-QURC.More specifically, the proposed design exhibits multiple quantum coding rates despite relying only on a single quantum encoder. The benefit of having multiple rates is that this scheme is capable of activating that specific code-rate/throughput combination, which meets the specific fidelity requirement. Moreover, the flexibility offered by the single-encoder MR-QSBCs enables us to leverage extrinsic information transfer (EXIT)-chart based heuristic optimization for determining the optimal weighting of each specific code-rate to be used in the MR-QSBCs scheme. Our simulation results show that the MR-QSBC-QURC scheme conceived performs relatively close to the ultimate limit of the quantum hashing bound. Specifically, when considering the target quantum coding rates of $r_Q = \{0.3, 0.4, 0.5, 0.6, 0.7\}$, the MR-QSBC-QURC operates at a distance of $D = \{0.042, 0.029, 0.030, 0.024, 0.017\}$ from the quantum hashing bound, respectively, at a quantum bit error ratio (QBER) of 10^{-3}.

(e) **Chapter ??: Quantum Low Density Parity Check Codes**

Pursuing further the design of iterative code structures, we focus our efforts on QLDPC codes, which may be constructed from the classical binary as well as quaternary codes. In this context, Section **??** reviews the various QLDPC construction methods, while the QLDPC decoding methods and the associated challenges are discussed in Section **??**. In Section **??**, we propose a formalism for constructing high-rate row-circulant QC-QLDPC codes from arbitrary row-circulant classical LDPC matrices. In Section **??**, 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 **??** details the reweighted BP algorithm, which is known to alleviate the structural flaw of short cycles in classical LDPC codes.

(f) Finally, in **Chapter 15** we summarize our findings along with a range of promising future research directions.

xx

Chapter 2

Preliminaries on Quantum Information

2.1 Introduction

Following the easy-reading historic introduction to the family of quantum error correction codes in Chapter 1, in this chapter, we will highlight the concepts of quantum information processing required for paving the way from classical to quantum information theory. The rest of this chapter is organized as follows. We commence with an introduction to quantum information in Section 2.2, followed by a brief tour of quantum information processing in Section 2.3. The quantum channel models used on this treatise are elaborated on in Section 2.4. In Section 2.5, we present the *no-cloning theorem*, followed by *quantum entanglement* in Section 2.6. Finally, we conclude in Section 2.8.

2.2 A Brief Review of Quantum Information

We live in a world where information is transmitted and computed in binary form. Therefore, the fundamental unit of information in the classical domain is the *binary digit* or referred to as a *bit*, which can be defined as follows:

$$c = \{0, 1\}. \tag{2.1}$$

Consequently, in the classical domain, each of the classical bits can only carry the value of 0 or 1, not both. By contrast, in the quantum domain, the fundamental unit of information is represented by a *quantum bit* or *qubit*. The state of a qubit can be described as a linear combination of 0 and 1, which can be described in form of their superposition. However, this

superposition of the two states will collapse to the corresponding classical state 0 or 1 upon observation or measurement. More specifically, the quantum state of a single qubit $|\psi\rangle$ can be formally expressed as

$$|\psi\rangle = \alpha_0|0\rangle + \alpha_1|1\rangle,\ \alpha_0, \alpha_1 \in \mathbb{C}, \tag{2.2}$$

where the probability of obtaining the classical state 0 and 1 upon measurement is given by $|\alpha_0|^2$ and $|\alpha_1|^2$, respectively. Since the values of α_0 and α_1 are associated with probability values, the unitary constraint of $|\alpha_0|^2 + |\alpha_1|^2 = 1$ is satisfied. A single-qubit system can also be viewed as a two-dimensional Hilbert space, where the computational basis vectors $|0\rangle$ and $|1\rangle$ are defined as follows:

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

Consequently, the quantum state of a single qubit given in Eq. (2.2), can also be represented by a two-dimension complex vector as follows:

$$|\psi\rangle = \alpha_0|0\rangle + \alpha_1|1\rangle = \alpha_0 \begin{pmatrix} 1 \\ 0 \end{pmatrix} + \alpha_1 \begin{pmatrix} 0 \\ 1 \end{pmatrix} = \begin{pmatrix} \alpha_0 \\ \alpha_1 \end{pmatrix},\ \alpha_0, \alpha_1 \in \mathbb{C}. \tag{2.4}$$

The representation of the basis vector of '0' using the notation $|0\rangle$ and the basis vector '1' using the notation $|1\rangle$ is referred to as the *ket* notation. The terminology *ket* comes from the *bra-ket* notation [54], where the *bra* notation refers to $\langle\psi|$, while the *ket* notation is used for $|\psi\rangle$. The relationship between $|\psi\rangle$ and the $\langle\psi|$ is defined as follows:

$$\langle\psi| = |\psi\rangle^\dagger, \tag{2.5}$$

where the notation $|\psi\rangle^\dagger$ indicates the conjugate transpose of $|\psi\rangle$. Explicitly, based on the vector representation of Eq. (2.4) and the definition of Eq. (2.5), we have

$$\langle\psi| = \begin{pmatrix} \alpha_0^* & \alpha_1^* \end{pmatrix}, \tag{2.6}$$

where α^* denotes the complex conjugate of α. Therefore, the following equality holds:

$$\langle\psi|\psi\rangle \equiv \langle\psi| \cdot |\psi\rangle = 1. \tag{2.7}$$

Since the coefficients α_0 and α_1 are complex numbers, without loss of generality, the state of a qubit can be more explicitly written as follows:

$$|\psi\rangle = e^{i\gamma}\left(\cos\frac{\theta}{2}|0\rangle + e^{i\varphi}\sin\frac{\theta}{2}|1\rangle\right). \tag{2.8}$$

Furthermore, since the coeffcient of $e^{i\gamma}$ has no observable effect, i.e. $e^{i\gamma}|\psi\rangle$ and $|\psi\rangle$ provide us with identical output probabilities upon measurements, the state of the qubit in Eq. (2.8) can

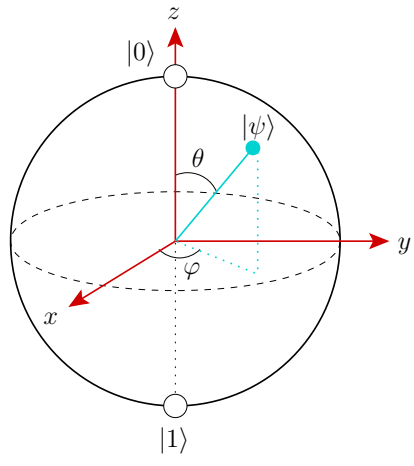


Figure 2.1: The Bloch sphere is the 3D representation of two-dimension complex vector space, which is parametrized by the variables θ and φ. A pure quantum state is represented by a point on the surface of a unit-radius. The computational basis of $|0\rangle$ corresponds to the north pole of the sphere, while the basis of $|1\rangle$ corresponds to the south pole.

be simplified to the following:

$$|\psi\rangle = \cos\frac{\theta}{2}|0\rangle + e^{i\varphi}\sin\frac{\theta}{2}|1\rangle. \tag{2.9}$$

Therefore, the quantum state of a qubit can be represented as a point on the surface of a unit-radius sphere, which is referred to as the Bloch sphere [181]. The point can be anywhere on the sphere and can be charcterized by two phase variables θ and φ. More explicitly, the 3D representation of a quantum state using the Bloch sphere is depicted in Fig. 2.1.

In general, a pair of vectors can be used as the basis vectors as long as both of them are orthonormal, i.e. both normalized and mutually orthogonal. For example, apart from the computational basis of $|0\rangle$ and $|1\rangle$ in the field of QECCs, the following Hadamard basis is also widely used:

$$|+\rangle = \frac{1}{\sqrt{2}}\begin{pmatrix}1\\1\end{pmatrix},\ |-\rangle = \frac{1}{\sqrt{2}}\begin{pmatrix}1\\-1\end{pmatrix}. \tag{2.10}$$

More explicitly, the Hadamard basis can be viewed as the equal-weight superposition of the computational basis $|0\rangle$ and $|1\rangle$ according to the following definition:

$$|+\rangle = \frac{|0\rangle + |1\rangle}{\sqrt{2}},\ |-\rangle = \frac{|0\rangle - |1\rangle}{\sqrt{2}}, \tag{2.11}$$

and vice versa, the computational basis $|0\rangle$ and $|1\rangle$ can be expressed as an equal-weight super-

position of the vectors from the Hadamard basis:

$$|0\rangle = \frac{|+\rangle + |-\rangle}{\sqrt{2}},\ |1\rangle = \frac{|+\rangle - |-\rangle}{\sqrt{2}}. \tag{2.12}$$

In order to extend the concept of quantum information to multi-qubit systems, we have to introduce the Kronecker tensor product or simply tensor product. Explicitly, for a pair of matrices $\mathbf{P}$ and $\mathbf{Q}$ having $(a \times b)$ elements and $(x \times y)$ elements, respectively, the resultant tensor product is a matrix having $(ax \times by)$ elements formulated by

$$\mathbf{P} \otimes \mathbf{Q} = \begin{pmatrix} p_{11}\mathbf{Q} & \cdots & p_{1(b-1)}\mathbf{Q} & p_{1b}\mathbf{Q} \\ p_{21}\mathbf{Q} & \cdots & p_{2(b-1)}\mathbf{Q} & p_{2b}\mathbf{Q} \\ \vdots & \ddots & \vdots & \vdots \\ p_{(a-1)1}\mathbf{Q} & \cdots & p_{(a-1)(b-1)}\mathbf{Q} & p_{(a-1)b}\mathbf{Q} \\ p_{a1}\mathbf{Q} & \cdots & p_{a(b-1)}\mathbf{Q} & p_{ab}\mathbf{Q} \end{pmatrix}. \tag{2.13}$$

For instance, a two-qubit system is represented by the tensor product between a pair of two-element vectors given in Eq. (2.4). More explicitly, let us consider two qubits having the state of $|\psi_1\rangle = \alpha_0|0\rangle + \alpha_1|1\rangle$ and $|\psi_2\rangle = \beta_0|0\rangle + \beta_1|1\rangle$. The superimposed state can be described as follows:

$$\begin{aligned} |\psi\rangle = |\psi_1\rangle \otimes |\psi_2\rangle &= \begin{pmatrix} \alpha_0 \\ \alpha_1 \end{pmatrix} \otimes \begin{pmatrix} \beta_0 \\ \beta_1 \end{pmatrix} = \begin{pmatrix} \alpha_0\beta_0 \\ \alpha_0\beta_1 \\ \alpha_1\beta_0 \\ \alpha_1\beta_1 \end{pmatrix} \\ &\equiv \alpha_0\beta_0|00\rangle + \alpha_0\beta_1|01\rangle + \alpha_1\beta_0|10\rangle + \alpha_1\beta_1|11\rangle, \end{aligned} \tag{2.14}$$

where $\alpha_0, \alpha_1, \beta_0, \beta_1 \in \mathbb{C}$. It can be observed that a two-qubit state is a superposition of all four possible states that can be generated by two classical bits i.e. 00, 01, 10 and 11. Additionally, the unitary constraint of $|\alpha_0\beta_0|^2 + |\alpha_0\beta_1|^2 + |\alpha_1\beta_0|^2 + |\alpha_1\beta_1|^2 = 1$ still holds. The tensor product of a pair of two-element vectors yields a vector consisting of 2^2 elements. Hence, the N-qubit system produces all of the 2^N possible states that can be generated by an N-bit sequence. If i is the decimal representation of an N-bit sequence, the N-qubit superposition state can be expressed by the Dirac notation as follows:

$$|\psi\rangle = \sum_{i=0}^{2^N-1} \alpha_i|i\rangle \text{ where } \alpha_i \in \mathbb{C} \text{ and } \sum_{i=0}^{2^N-1} |\alpha_i|^2 = 1. \tag{2.15}$$

As an instance of a very special case, where we have N qubits all having $|+\rangle$ state, provides us with the equal-weight superposition of 2^N possible states generated by all possible combination

of N-bit sequences as follows:

$$|+\rangle^{\otimes N} \equiv |+\rangle_1 \otimes |+\rangle_2 \otimes \cdots \otimes |+\rangle_N = \frac{1}{\sqrt{2^N}} \sum_{i=0}^{2^N-1} |i\rangle, \tag{2.16}$$

where the superscript $\otimes N$ of $|+\rangle$ represents the N-fold tensor product. It can be observed that the probability of obtaining each of the 2^N quantum states upon their observation in the computational basis is equal to $\frac{1}{2^N}$. This particular state is often used as the initialized quantum state for various quantum computing algorithms, such as Shor's quantum factoring algorithm [15,16] and Grover's quantum search algorithm [17,20] as well as for quantum error correction codes.

2.3 Quantum Information Processing

So far we have described the quantum state of a qubit. Similar to the classical domain, the qubit can be manipulated to carry out a specific quantum computation or communication task. The evolution of quantum states can be classified into two categories: reversible and irreversible evolution. A reversible evolution is constituted by a unitary transformation, while an irreversible evolution is due to the measurement or observation of our quantum states, which involves an interaction with the so-called *observer*.

2.3.1 Unitary Transformation

For quantum computation and communication, the desired unitary transformations are carried out by components termed as quantum gates. In classical computers, the circuits rely on logical gates such as AND, OR, and XOR gates. Similar to the classical computer, a quantum computer relies on quantum gates, which can be mathematically represented by unitary transformation satisfying the following properties:

(a) Quantum gates, which are denoted by U, act linearly on the superposition of quantum states, which is defined as

$$U(\alpha_0|0\rangle + \alpha_1|1\rangle) = \alpha_0 U(|0\rangle) + \alpha_1 U(|1\rangle). \tag{2.17}$$

(b) The quantum gate U is characterized by a unitary matrix for ensuring that the final probability of all possible quantum states after the transformation is equal to 1. The unitary nature of quantum gates is described as

$$UU^\dagger = \mathbf{I}, \tag{2.18}$$

where $U^\dagger$ is the Hermitian conjugate of U and $\mathbf{I}$ is an identity matrix.

Some of the basic quantum gates used in quantum computation and communication will be

discussed in the following subsections.

2.3.1.1 Pauli Gates

First, we would like to introduce the quantum gates acting on a single qubit. Pauli gates or Pauli operators are the most common single-qubit quantum gates used for manipulating the quantum state of a single-qubit. Pauli gates are defined by Pauli matrices as follows:

$$\mathbf{I} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \ \mathbf{X} = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \ \mathbf{Y} = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}, \ \mathbf{Z} = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}. \tag{2.19}$$

Let us assume that we have a single-qubit having a quantum state of $|\psi\rangle = \alpha_0|0\rangle + \alpha_1|1\rangle$. The Pauli matrix $\mathbf{X}$ transforms the quantum state of a single qubit into $|\psi'\rangle$ as follows:

$$\begin{aligned} |\psi'\rangle = \mathbf{X}|\psi\rangle &= \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} \alpha_0 \\ \alpha_1 \end{pmatrix} = \begin{pmatrix} \alpha_1 \\ \alpha_0 \end{pmatrix} \\ &\equiv \alpha_1|0\rangle + \alpha_0|1\rangle. \end{aligned} \tag{2.20}$$

The transformation due to the Pauli matrix $\mathbf{Z}$ is given by

$$\begin{aligned} |\psi'\rangle = \mathbf{Z}|\psi\rangle &= \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \begin{pmatrix} \alpha_0 \\ \alpha_1 \end{pmatrix} = \begin{pmatrix} \alpha_0 \\ -\alpha_1 \end{pmatrix} \\ &\equiv \alpha_0|0\rangle - \alpha_1|1\rangle, \end{aligned} \tag{2.21}$$

while the Pauli matrix $\mathbf{Y}$ transforms a single qubit $|\psi\rangle$ as follows:

$$\begin{aligned} |\psi'\rangle = \mathbf{Y}|\psi\rangle &= \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix} \begin{pmatrix} \alpha_0 \\ \alpha_1 \end{pmatrix} = \begin{pmatrix} i\alpha_1 \\ -i\alpha_0 \end{pmatrix} \\ &\equiv i\alpha_1|0\rangle - i\alpha_0|1\rangle. \end{aligned} \tag{2.22}$$

Normally, we use unitary transformations for activating the desired quantum-domain evolution of the quantum states of qubits. However, due to the interaction with the environment, or due to the imperfection of quantum gates themselves, some undesired unitary transformations may also take place. In this case, each of the Pauli matrices may also be used for reflecting the discrete set of errors that may occur in a single-qubit system, such as a bit-flip error ($\mathbf{X}$), a phase-flip error ($\mathbf{Z}$), as well as both bit-flip and phase-flip error ($i\mathbf{XZ} = \mathbf{Y}$). Finally, the Pauli matrix $\mathbf{I}$ represents the identity operator. More details concerning this issue will be presented in Subsection 2.4.

2.3.1.2 Hadamard Gate

The Hadamard gate maps a pure state of $|0\rangle$ and $|1\rangle$ into an equiprobable superposition of both states. The transformations mapping the pure states by Hadamard gates are described below:

$$\mathbf{H}|0\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \equiv |+\rangle, \tag{2.23}$$

$$\mathbf{H}|1\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle) \equiv |-\rangle. \tag{2.24}$$

The Hadamard gates may be represented using a 2×2 matrix as

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

The Hadamard gates can be used for transforming the computational basis of $|0\rangle$ and $|1\rangle$ into the Hadamard basis of $|+\rangle$ and $|-\rangle$, which will be shown later to be very useful for handling different types of quantum errors imposed by quantum decoherence. The interesting property of this transformation is that a Pauli matrix $\mathbf{X}$ in the computational basis $\{|0\rangle, |1\rangle\}$ behaves similarly to a Pauli matrix $\mathbf{Z}$ in the Hadamard basis $\{|+\rangle, |-\rangle\}$. More explicitly, let us assume that we have a single qubit having a quantum state of $|\psi\rangle = \alpha_0|0\rangle + \alpha_1|1\rangle$. This specific qubit can also be rewritten in the Hadamard basis as $|\psi\rangle = \beta_0|0\rangle + \beta_1|1\rangle$, where we have $\beta_0 = \frac{\alpha_0+\alpha_1}{\sqrt{2}}$ and $\beta_1 = \frac{\alpha_0-\alpha_1}{\sqrt{2}}$. The action of the Pauli matrix $\mathbf{X}$ in the computational basis $|0\rangle$ and $|1\rangle$ can be described as follows:

$$|\psi'\rangle = \mathbf{X}|\psi\rangle = \alpha_1|0\rangle + \alpha_0|1\rangle. \tag{2.26}$$

Now, the action of the Pauli matrix $\mathbf{Z}$ in the Hadamard basis is as follows:

$$|\psi'\rangle = \mathbf{Z}|\psi\rangle = \beta_1|+\rangle + \beta_0|-\rangle. \tag{2.27}$$

It can be observed from Eq. (2.26) and (2.27) that the effect of swapping the complex-valued coefficient between the basis vector in computational basis $|0\rangle$ and $|1\rangle$ is due to the Pauli matrix $\mathbf{X}$, but in Hadamard basis this effect is the result of the Pauli matrix $\mathbf{Z}$. Similarly, let us observe the effect of the Pauli matrix $\mathbf{Z}$ on the quantum state of a single qubit in the computational basis, which can be expressed as follows:

$$|\psi'\rangle = \mathbf{Z}|\psi\rangle = \alpha_0|0\rangle - \alpha_1|1\rangle. \tag{2.28}$$

By contrast, the effect of the Pauli matrix $\mathbf{X}$ on a single-qubit quantum state in the Hadamard basis can be written as

$$|\psi'\rangle = \mathbf{X}|\psi\rangle = \beta_0|0\rangle - \beta_1|1\rangle. \tag{2.29}$$

From Eq. (2.28) and (2.29), we can observe that the Pauli matrix $\mathbf{Z}$ flips the sign of the complex-valued coefficient of the basis vector $|1\rangle$. However, the sign of the complex-valued coefficient of the basis vector $|-\rangle$ is changed by the operation of the Pauli matrix $\mathbf{X}$ in Hadamard basis. This specific property will be beneficially exploited in the construction of QSCs, specifically when we

are dealing with different types of errors imposed by quantum decoherence, as it will be detailed in Chapter 3.

2.3.1.3 Phase Gate

A phase gate **S** or equivalently i-phase shift gate transforms the quantum state of a single qubit by shifting the phase of state $|1\rangle$ by a factor of i, while the state of $|0\rangle$ remains intact. More specifically, the phase gate **S** is defined by a 2×2 matrix as follows:

$$\mathbf{S} = \begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix}. \tag{2.30}$$

The relationship between the phase gate **S** and Pauli gate **Z** is formulated below:

$$\mathbf{S} = \sqrt{\mathbf{Z}}, \tag{2.31}$$

$$\mathbf{S}^2 = \mathbf{Z}. \tag{2.32}$$

2.3.1.4 Controlled-NOT Gate

A controlled-NOT or CNOT gate acts similarly to the XOR logic gate of a classical computer. To elaborate, the inputs of the CNOT gate are labelled as a control qubit $|c\rangle$ and a target qubit $|x\rangle$. When the control qubit $|c\rangle$ is in the state of $|1\rangle$, the target qubit $|x\rangle$ undergoes the NOT operation, or equivalently it is subjected to a Pauli matrix **X** imposing a bit-flip. Otherwise, the state of the target qubit $|x\rangle$ is left unchanged. Therefore, the transformation performed by a CNOT gate can be defined as

$$\mathbf{CNOT}(|c, x\rangle) \equiv |c, (c \oplus x)\rangle. \tag{2.33}$$

The CNOT gate can also be viewed as a controlled Pauli **X** gate and the corresponding matrix describing the CNOT gate operation is defined as follows:

$$\mathbf{CNOT} = \mathbf{CX} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} = \begin{pmatrix} \mathbf{I} & 0 \\ 0 & \mathbf{X} \end{pmatrix}. \tag{2.34}$$

The quantum circuit representation of a CNOT gate is protrayed in Fig. 2.2.

The CNOT gate together with the Pauli gates, the Hadamard gate, and the phase gate create a special class of quantum gates called the Clifford group [182]. A special property of the Clifford group is that they can be efficiently simulated using classical computers. Consequently, a

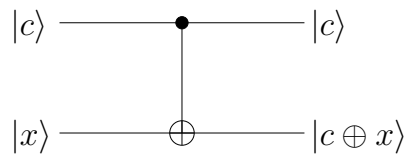


Figure 2.2: The quantum circuit of the CNOT gate, where $|c\rangle$ is the control qubit and $|x\rangle$ is the target qubit.

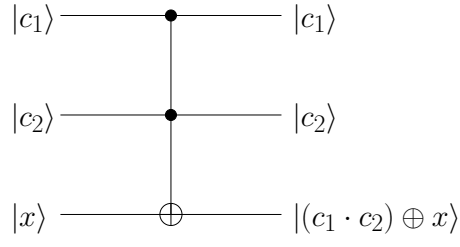


Figure 2.3: The quantum circuit for Toffoli gate.

quantum circuit purely relying on quantum Clifford gates can be simulated efficiently in classical computer and hence, is not capable of offering any substantial quantum advantage in terms of computational speed-up. By contrast, a more general class of quantum computers capable of achieving computational advantages has to be capable of carrying out unitary transformations including the so-called non-Clifford quantum gates.

2.3.1.5 Toffoli Gate

A very popular example of non-Clifford quantum gates is constituted by Toffoli gate [183], which acts as a controlled-CNOT gate. However, the main difference that it has one target qubit $|x\rangle$ and two control qubits, $|c_1\rangle$ and $|c_2\rangle$. The target qubit $|x\rangle$ undergoes a bit flip (**X**) only when both of the control qubits $|c_1\rangle$ and $|c_2\rangle$ are in the state of $|1\rangle$. Otherwise, the state of the target qubit $|x\rangle$ remains intact. The Toffoli gate processes the quantum state of three qubits, hence its matrix representation reflects the unitary transformation of 2^3 Hilbert space vectors. Explicitly,

the unitary matrix of a Toffoli gate is defined as

$$\mathbf{CCNOT} \quad = \quad \mathbf{CCX} \quad = \quad \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix}. \tag{2.35}$$

The quantum circuit representation of a Toffoli gate is portrayed in Fig. 2.3. Furthermore, the generalized version of Toffoli gates is required for executing Grover's quantum search algorithm, which has been proved to offer a beneficial quantum computation advantage [30, 184]. The generalized version of Toffoli gate, namely the $(n-1)$-controlled-NOT gate [185], which is denoted by $\mathbf{C}^{\otimes(\mathbf{n-1})}\mathbf{NOT}$, transforms a vector of 2^n dimensions in Hilbert space. This can be represented by a $(2^n \times 2^n)$-element unitary matrix as follows:

$$\mathbf{C}^{\otimes(\mathbf{n-1})}\mathbf{NOT} = \begin{pmatrix} \mathbf{I}^{2^n-2} & \mathbf{0} \\ \mathbf{0} & \mathbf{X} \end{pmatrix}, \tag{2.36}$$

where $\mathbf{I}^{2^n-2}$ is an identity matrix of dimension (2^n-2), $\mathbf{X}$ is the Pauli matrix, and the rest of the elements of the matrix are equal to zero. According to the unitary transformation of Eq. (2.36), the Pauli matrix $\mathbf{X}$ will be applied to the target qubit $|x\rangle$, if all the $(n-1)$ quantum states of the control qubits sequence are equal to $|1\rangle$. This is a conceptually simple yet powerful unitary transformation carried out by a quantum gate, but it cannot be simulated efficiently using classical computers.

2.3.2 Quantum Measurement

We have briefly described the reversible evolution of quantum information, which is mathematically represented by unitary transformations and it is physically realized by quantum gates. Here, we continue by briefly describing the irreversible evolution of quantum information, namely quantum measurement. The final values of a specific quantum computation or quantum communication task have to be read out at the end. Hence, we need the so-called quantum measurement operators. Quantum measurement is described by a collection of measurement operators $\{M_i\}$, where i denotes the measurement outcome that may occur after observations. If a quantum system is in the state of $|\psi\rangle$ before measurement, the probability of the result i may be expressed

as

$$p(i) = \langle\psi|M_i^\dagger M_i|\psi\rangle. \tag{2.37}$$

Consequently, the resultant state after measurement is given by

$$|\psi'\rangle = \frac{M_i|\psi\rangle}{\sqrt{\langle\psi|M_i^\dagger M_i|\psi\rangle}} = \frac{M_i|\psi\rangle}{\sqrt{p(i)}}. \tag{2.38}$$

The set of measurement operators have to satisfy the so-called completeness criteria, which are defined as

$$\sum_i M_i^\dagger M_i = \mathbf{I}, \tag{2.39}$$

$$\sum_i p(i) = \sum_i \langle\psi|M_i^\dagger M_i|\psi\rangle = 1. \tag{2.40}$$

For example, for a pair of computational basis states $|0\rangle$ and $|1\rangle$, we have two measurement operators $M_0 = |0\rangle\langle 0|$ and $M_1 = |1\rangle\langle 1|$. For an arbitrary qubit having a quantum state of $|\psi\rangle = \alpha_0|0\rangle + \alpha_1|1\rangle$, the probability of obtaining each state is formulated by

$$p(0) = |\psi\rangle M_0^\dagger M_0 \langle\psi| = |\alpha_0|^2, \tag{2.41}$$

$$p(1) = |\psi\rangle M_1^\dagger M_1 \langle\psi| = |\alpha_1|^2. \tag{2.42}$$

Hence, the qubit state after measurement may be expressed as

$$|\psi'\rangle_{m=0} = \frac{M_0|\psi\rangle}{|\alpha_0|} = \frac{\alpha_0|0\rangle}{|\alpha_0|}, \tag{2.43}$$

$$|\psi'\rangle_{m=1} = \frac{M_1|\psi\rangle}{|\alpha_1|} = \frac{\alpha_1|1\rangle}{|\alpha_1|}. \tag{2.44}$$

Equations (2.43) and (2.44) explicitly reflect that a qubit will collapse into a classical bit after measurement. The quantum measurement is an irreversible process, because once we obtain our measurement results in form of classical states, it is impossible to learn the values of the complex coefficients of each of the basis vectors. Therefore, we cannot reconstruct the original quantum state from a single measurement result. Having said that, the quantum state can nonetheless be approximated with a specific level of certainty, as long as we can provide a reasonably large number of qubits prepared in an identical quantum state using a method called as quantum state tomography [186, 187].

2.4 Quantum Decoherence

Quantum computers are composed of numerous quantum gates, which are prone to environmental impairments resulting in a short coherence time. Consequently, due to the deleterious effects of quantum decoherence, the resultant quantum state at the output may not be the desired outcome of the quantum computation. In this treatise, the terminology quantum channels will

be used for encapsulating all the aforementioned imperfections caused by the quantum decoherence. A quantum channel inflicting a single qubit error can be represented by the Pauli group $\mathcal{P}_1$, which defines the discrete set of possible unitary transformations imposed on a single qubit. Explicitly, the Pauli group $\mathcal{P}_1$ is defined as

$$\mathcal{P}_1 = \{eP : P \in \{\mathbf{I}, \mathbf{X}, \mathbf{Y}, \mathbf{Z}\}, e \in \{\pm 1, \pm i\}\}, \tag{2.45}$$

which is closed under multiplication. The unitary matrices $\mathbf{X}$ and $\mathbf{Z}$ represent the bit-flip and the phase-flip, respectively, while the matrix $\mathbf{Y}$ represents a simultaneous bit-flip and phase-flip. Finally, the identity unitary matrix $\mathbf{I}$ denotes the absence of error. These Pauli matrices have been defined in Eq. (2.19). Since the qubits whose quantum states only differ in their global phase can be deemed to be equivalent, the reduced Pauli group $\mathcal{P}_1$ denoted by $\mathcal{P}_1^*$ is often used for the sake of simulating the quantum errors by exploiting the Pauli-to-binary isomorphism [181], which is defined as

$$\mathcal{P}_1^* = \{\mathbf{I}, \mathbf{X}, \mathbf{Y}, \mathbf{Z}\}. \tag{2.46}$$

From a different perspective, we can also investigate the quantum channel affecting on quantum information by describing the error effects as a unitary transformation $U(\Delta\theta.\Delta\varphi)$, which maps a point on the surface of the Bloch sphere into a different coordinate. Explicitly, using Eq. (2.9), we can define the quantum error imposed by the quantum channel as follows:

$$U(\Delta\theta, \Delta\varphi)|\psi\rangle = \cos\left(\frac{\theta + \Delta\theta}{2}\right)|0\rangle + e^{(i\varphi + \Delta\varphi)} \sin\left(\frac{\theta + \Delta\theta}{2}\right)|1\rangle, \tag{2.47}$$

where $(\theta + \Delta\theta)$ and $(\varphi + \Delta\varphi)$ are the new variables defining the resultant quantum state. By expanding the quantum state in Eq. (2.47), we can rewrite the unitary transformation of $U(\Delta\theta, \Delta\varphi)$ as follows [188]:

$$U(\Delta\theta, \Delta\varphi)|\psi\rangle = \alpha_{\mathbf{I}}\mathbf{I}|\psi\rangle + \alpha_{\mathbf{X}}\mathbf{X}|\psi\rangle + \alpha_{\mathbf{Z}}\mathbf{Z}|\psi\rangle + \alpha_{\mathbf{Y}}\mathbf{Y}|\psi\rangle, \tag{2.48}$$

where $\alpha_{\mathbf{I}}$, $\alpha_{\mathbf{X}}$, $\alpha_{\mathbf{Z}}$, and $\alpha_{\mathbf{Y}}$ are the resultant expansion coefficients. Here, we want to highlight that although the nature of quantum errors is continuous, the unitary transformation can be expressed as a linear combination of Pauli matrices $\mathbf{I}$, $\mathbf{X}$, $\mathbf{Z}$, and $\mathbf{Y}$. Furthermore, noting that apart from a phase-difference, the Pauli matrix $\mathbf{Y}$ is equivalent to the product of the Pauli matrices $\mathbf{X}$ and $\mathbf{Z}$, i.e we have $i\mathbf{XZ} = \mathbf{Y}$, the expression given in Eq. (2.48) can be further simplified into the following:

$$U(\Delta\theta, \Delta\varphi)|\psi\rangle = \alpha_{\mathbf{I}}\mathbf{I}|\psi\rangle + \alpha_{\mathbf{X}}\mathbf{X}|\psi\rangle + \alpha_{\mathbf{Z}}\mathbf{Z}|\psi\rangle + \alpha_{\mathbf{XZ}}\mathbf{XZ}|\psi\rangle, \tag{2.49}$$

where now the quantum error can be described as a linear combination of the Pauli matrices $\mathbf{I}$, $\mathbf{X}$, $\mathbf{Z}$, and $\mathbf{XZ}$. At the end of the quantum computation or communication task, the continuous nature of quantum errors will be projected into one of the following possibilites: the absence of error ($\mathbf{I}$), a bit-flip error $\mathbf{X}$, a phase-flip error $\mathbf{Z}$, or both bit-flip and phase-flip errors $\mathbf{XZ}$. This concept is known as the discretization of quantum errors, which is a very useful tool when it comes to designing the associated QECCs for mitigating these error effects.

Now, let us now consider the more general concept of quantum channels affecting an N-qubit system. When considering the quantum state of an N-qubit system, the quantum channel may be described by the Pauli group $\mathcal{P}_n$, which is represented by an n-fold tensor product of $\mathcal{P}_1$ as defined below:

$$\mathcal{P}_n = \{P_1 \otimes P_2 \cdots \otimes P_n | P_j \in \mathcal{P}_1\}, \tag{2.50}$$

where the index j represents the j-th qubit of a system having n physical qubits. An operator $P \in \mathcal{P}_n$ transforms the legitimate quantum state $|\psi\rangle$ into an impaired quantum state $|\widehat{\psi}\rangle$, as formally described below:

$$|\widehat{\psi}\rangle = P|\psi\rangle. \tag{2.51}$$

The quantum channel inflicts an error $P \in \mathcal{P}_n$ on an N-qubit string, where each qubit may independently experience either a bit-flip error ($\mathbf{X}$), a phase-flip error ($\mathbf{Z}$), or both bit-flip and phase-flip error ($i\mathbf{XZ} = \mathbf{Y}$). The effect of each Pauli matrix has been described earlier in Subsection 2.3.1.1. Let us now proceed by applying the unitary transformations to the multi-qubit state of Eq. (2.14), which can also be represented as a four-element complex vector as follows:

$$|\psi\rangle = \begin{pmatrix} \alpha_{00} \\ \alpha_{01} \\ \alpha_{10} \\ \alpha_{11} \end{pmatrix}. \tag{2.52}$$

For instance, let us assume that the quantum channel inflicts a two-qubit unitary transformation of $(\mathbf{X} \otimes \mathbf{I})$[1] upon a two-qubit state. The evolution of the quantum state can be described as

[1]For the sake of simplifying the notation, a set of Pauli matrices for defining a multi-qubit unitary transformation usually does not include the "$\otimes$" operator. For example, a unitary transformation $(\mathbf{X} \otimes \mathbf{Z} \otimes \mathbf{X} \otimes \mathbf{I})$ acting upon a 4-qubit operand can simply be rewritten as $\mathbf{XZXI}$. In the rest of Part 1, the latter representation will be used for stabilizer operators, unless it is stated otherwise.

follows:

$$
\begin{aligned}
|\psi'\rangle &= (\mathbf{X}\otimes\mathbf{I})\,|\psi\rangle \\
&= \left(\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \otimes \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}\right) \cdot \begin{pmatrix} \alpha_{00} \\ \alpha_{01} \\ \alpha_{10} \\ \alpha_{11} \end{pmatrix} \\
&= \begin{pmatrix} 0 & & \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \\ \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} & & 0 \end{pmatrix} \cdot \begin{pmatrix} \alpha_{00} \\ \alpha_{01} \\ \alpha_{10} \\ \alpha_{11} \end{pmatrix} \\
&= \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix} \cdot \begin{pmatrix} \alpha_{00} \\ \alpha_{01} \\ \alpha_{10} \\ \alpha_{11} \end{pmatrix} = \begin{pmatrix} \alpha_{10} \\ \alpha_{11} \\ \alpha_{00} \\ \alpha_{01} \end{pmatrix} \\
&\equiv \alpha_{10}|00\rangle + \alpha_{11}|01\rangle + \alpha_{00}|10\rangle + \alpha_{01}|11\rangle. \qquad (2.53)
\end{aligned}
$$

The final state of Eq. (2.53) can also be obtained without expanding the tensor product of the unitary transformation by flipping the state of the first qubit, since the unitary transformation of $\mathbf{XI}$ represents a bit-flip error of the first qubit owing to the bit-flip ($\mathbf{X}$), while the second qubit does not experience any impairment owing to the effect of identity ($\mathbf{I}$). More explicitly, due to the unitary transformation $\mathbf{XI}$, the state of $|00\rangle$ is changed to the state of $|10\rangle$. The same transformation is also applied to the states of $|01\rangle$, $|10\rangle$, and $|11\rangle$, where they are transformed to the states of $|11\rangle$, $|00\rangle$, $|01\rangle$, respectively. Hence, the magnitude associated with the state of $|00\rangle$ is no longer α_{00} - it becomes α_{10}. Therefore, the coefficients associated with the states of $|01\rangle$, $|10\rangle$, and $|11\rangle$ now are α_{11}, α_{00}, and α_{01}, respectively.

In this treatise, we focus our discussions on the family of QSCs, which can be derived from their classical counterparts. Even though most of the well-known QSCs are derived on the basis of the classical-to-quantum isomorphism, there is a certain property of the QSCs, which can only be found in the quantum domain, i.e. it has no classical counterpart. This is the so-called degeneracy property. More explicitly, a degeneracy property implies that a set of different error patterns of $P \in \mathcal{P}_n$ may yield an identical corrupted state and consequently we only need a single error recovery operator for reinstating the original quantum state. For example, let us consider a two-qubit system having the following quantum state:

$$
|\psi\rangle = \frac{1}{\sqrt{2}}\left(|00\rangle + |11\rangle\right), \qquad (2.54)
$$

and let us consider two different error patterns, which can be described as a pair of two-qubit unitary transformations given by $P_1 = \mathbf{IZ}$ and $P_2 = \mathbf{ZI}$. The resultant state after the error pattern P_1 is imposed on the two-qubit system can be described as follows:

$$\begin{aligned} |\psi_1'\rangle &= \mathbf{IZ}|\psi\rangle \\ &= \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \end{pmatrix} \cdot \begin{pmatrix} \frac{1}{\sqrt{2}} \\ 0 \\ 0 \\ \frac{1}{\sqrt{2}} \end{pmatrix} = \begin{pmatrix} \frac{1}{\sqrt{2}} \\ 0 \\ 0 \\ -\frac{1}{\sqrt{2}} \end{pmatrix} \\ &\equiv \frac{1}{\sqrt{2}} \left(|00\rangle - |11\rangle \right), \end{aligned} \tag{2.55}$$

while applying P_2 to the state of $|\psi\rangle$ will result in the following quantum state:

$$\begin{aligned} |\psi_2'\rangle &= \mathbf{ZI}|\psi\rangle \\ &= \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & -1 \end{pmatrix} \cdot \begin{pmatrix} \frac{1}{\sqrt{2}} \\ 0 \\ 0 \\ \frac{1}{\sqrt{2}} \end{pmatrix} = \begin{pmatrix} \frac{1}{\sqrt{2}} \\ 0 \\ 0 \\ -\frac{1}{\sqrt{2}} \end{pmatrix} \\ &\equiv \frac{1}{\sqrt{2}} \left(|00\rangle - |11\rangle \right). \end{aligned} \tag{2.56}$$

Since the error patterns $P_1 = \mathbf{IZ}$ and $P_2 = \mathbf{ZI}$ yield identical corrupted states $|\psi_1'\rangle = |\psi_2'\rangle$, they undoubtly require an identical recovery procedure. Indeed, exploiting the degeneracy property may potentially increase the error correction capability of QSCs. However, the question as to whether there exist degenerate QSCs that are capable of operating beyond the quantum Hamming bound - which is only applicable to non-degenerate QSCs - remains unresolved at the time of writing. Having said that, some research on finding the bounds of degenerate quantum codes can be found in [61, 189, 190].

The terminology of quantum channel incorporates numerous physical phenomena, including the imperfect quantum gates or quantum memories. In either of these cases, the qubits undergo a state change, which can be a linear combination of two types of quantum errors, as described below:

- Amplitude error or bit-flip which turns the state $|0\rangle$ into $|1\rangle$, or state $|1\rangle$ into $|0\rangle$. We also encounter this type of errors in classical settings.
- Phase error or phase-flip which has no impact on the state $|0\rangle$, but it turns the state $|1\rangle$ into $-|1\rangle$. This type of error distinguishes a quantum system from classical systems.

The quantum channel may inflict an individual bit-flip ($\mathbf{X}$), a phase-flip ($\mathbf{Z}$), as well as a simultaneous bit-flip and phase-flip ($\mathbf{Y}$) error with a probability of $p_{\mathbf{X}}$, $p_{\mathbf{Z}}$, and $p_{\mathbf{Y}}$, respectively.

These deleterious effects may be inflicted upon each of the qubits within the N-qubit block. For the sake of accommodating three different types of errors, the following are the common quantum channel models widely used for simulation.

2.4.1 Symmetric Quantum Depolarizing Chanel

The *symmetric depolarizing channel* is characterized by the depolarizing probability p. It creates an N-tuple of Pauli matrices $P \in \mathcal{P}_n$ for an N-qubit system, where the i-th qubit undergoes either a bit-flip ($\mathbf{X}$) error, a phase-flip ($\mathbf{Z}$) error, or simultaneous bit-flip and phase-flip ($\mathbf{Y}$) errors with an equal probability of $p_e = p/3$, where p denotes the overall depolarizing probability [77, 81, 191]. Hence the probability of error-free transmission is simply given by

$$p_{\mathbf{I}} = 1 - 3p_e = 1 - p. \tag{2.57}$$

In the realms of practical physical implementations, it has been observed that the probability of bit-flip errors ($p_{\mathbf{X}}$) is not equal to the probability of phase-flip errors ($p_{\mathbf{Z}}$) [192, 193, 194, 195, 196]. More specifically, popular materials invoked for producing quantum gates often exhibit an asymmetric behaviour, where the phase-flip errors are several orders of magnitude more likely to occur than the bit-flip errors.

2.4.2 Asymmetric Quantum Depolarizing Chanel

The more realistic quantum channel model which relies on this assumption is referred to as an *asymmetric quantum depolarizing channel* [197]. In the asymmetric case, an extra parameter referred to as the channel's asymmetry ratio α is introduced for portraying the ratio of the phase-flip probability $p_{\mathbf{Z}}$ and the bit-flip probability $p_{\mathbf{X}}$ as follows:

$$\alpha = \frac{p_{\mathbf{Z}}}{p_{\mathbf{X}}}. \tag{2.58}$$

In practice, the channel's asymmetry ratio has popular values of $\alpha = 10^2, 10^4, 10^6$ [192, 193, 194, 195, 196]. Again, since the probability of phase-flip errors ($p_{\mathbf{Z}}$) is several orders of magnitudes higher than that of bit-flip errors ($p_{\mathbf{X}}$), the probability of both phase and bit-flip errors ($p_{\mathbf{Y}}$) is assumed to be close to $p_{\mathbf{X}}$, which can be written as

$$p_{\mathbf{Y}} \approx p_{\mathbf{X}}. \tag{2.59}$$

Consequently, we can directly describe the parameters of the asymmetric depolarizing channel model using the depolarizing probability (p) and the channel's asymmetry ratio (α) as follows:

$$p_{\mathbf{I}} = 1 - p, \tag{2.60}$$

$$p_{\mathbf{X}} = \left(\frac{1}{\alpha + 2}\right) p, \tag{2.61}$$

$$p_{\mathbf{Y}} = \left(\frac{1}{\alpha + 2}\right) p, \tag{2.62}$$

$$p_{\mathbf{Z}} = \left(\frac{\alpha}{\alpha + 2}\right) p. \tag{2.63}$$

2.4.3 Independent Binary-Symmetric Chanel

By contrast, the *independent binary-symmetric channel* model assumes that each qubit may experience bit-flip errors ($\mathbf{X}$) and phase-flip errors ($\mathbf{Z}$) independently. This model is equivalent to the combination of two independent binary symmetric channels, where one channel inflicts only bit-flip ($\mathbf{X}$) errors and the other channel only inflict phase flip ($\mathbf{Z}$) errors. Therefore, the parameters describing the independent binary-symmetric channel model are

$$p_{\mathbf{I}} = 1 - p_{\mathbf{X}} - p_{\mathbf{Y}} - p_{\mathbf{Z}}, \tag{2.64}$$

$$p_{\mathbf{X}} = p_e^x \left(1 - p_e^z\right), \tag{2.65}$$

$$p_{\mathbf{Y}} = p_e^x p_e^z, \tag{2.66}$$

$$p_{\mathbf{Z}} = p_e^z \left(1 - p_e^x\right), \tag{2.67}$$

where p_e^x denotes the crossover or flip probability in the channel $\mathbf{X}$ and p_e^z the crossover or flip probability in the channel $\mathbf{Z}$. For the sake of approximating the symmetric quantum depolarizing channel, the crossover probability for each channel is assumed to be equal to $2p/3$, which can be written as

$$p_e^x = p_e^z = \frac{2p}{3}. \tag{2.68}$$

Therefore, the final parameters defining the independent binary-symmetric channel in Eq. (2.67), can be rewritten as follows:

$$p_{\mathbf{I}} = 1 - \left(\frac{4p}{3} - \frac{4p^2}{9}\right), \tag{2.69}$$

$$p_{\mathbf{X}} = \frac{2p}{3} - \frac{4p^2}{9}, \tag{2.70}$$

$$p_{\mathbf{Y}} = \frac{4p^2}{9}, \tag{2.71}$$

$$p_{\mathbf{Z}} = \frac{2p}{3} - \frac{4p^2}{9}. \tag{2.72}$$

Additionally, the independent binary-symmetric channel model can be extended directly to its the asymmetric version, which has $p_e^x \neq p_e^z$. It is important to note that most of our simulations in this treatise rely on the independent binary-symmetric channel model, while most of the

analytical results are derived using the assumption of having a symmetric quantum depolarizing channel.

2.5 No-Cloning Theorem

The concept of copying or cloning the information to protect it is widely used in classical communication and information theory. Unfortunately, the same concept is no longer valid due to the *no-cloning theorem*, as we enter into the quantum domain. Explicitly, the *no-cloning theorem* states that no unitary transformation can copy an arbitrary superposition of quantum states from one qubit to another qubit.

Proof. Let us assume that we have two orthogonal basis vector of $|\psi_1\rangle$ and $|\psi_2\rangle$. Let us consider a single-qubit system having a quantum state of $|\psi_1\rangle$, which acts as the data qubit, and another single-qubit system having a quantum state of $|\psi_t\rangle$, which acts as the target qubit for copying the quantum state from data qubit. Thus, the initial value before activating the copying mechanism is given by

$$|\psi'\rangle = |\psi_1\rangle \otimes |\psi_t\rangle. \tag{2.73}$$

Let us assume furthermore that there exists a unitary transformation U having a copying mechanism. Then ideally the final joint quantum state after copying is formulated as

$$U(|\psi_1\rangle \otimes |\psi_t\rangle) = |\psi_1\rangle \otimes |\psi_1\rangle. \tag{2.74}$$

Upon assuming that this copying mechanism is applied to the second data qubit having a quantum state of $|\psi_2\rangle$, the copying procedure yields

$$U(|\psi_2\rangle \otimes |\psi_t\rangle) = |\psi_2\rangle \otimes |\psi_2\rangle. \tag{2.75}$$

Let us now proceed with the cloning of a superimposed quantum state, i.e. $|\lambda\rangle = \alpha_1|\psi_1\rangle + \alpha_2|\psi_2\rangle$, where $\alpha_1, \alpha_2 \in \mathbb{C}$. The copying mechanism can be formulated as

$$U(|\lambda\rangle \otimes |\psi_t\rangle) = |\lambda\rangle \otimes |\lambda\rangle. \tag{2.76}$$

Expanding the left-hand side of Eq. (2.76) yields

$$\begin{aligned} U(|\lambda\rangle \otimes |\psi_t\rangle) &= U\left[(\alpha_1|\psi_1\rangle + \alpha_2|\psi_2\rangle) \otimes |\psi_t\rangle\right] \\ &= U(\alpha_1|\psi_1\rangle \otimes |\psi_t\rangle) + U(\alpha_2|\psi_2\rangle \otimes |\psi_t\rangle) \\ &= \alpha_1|\psi_1\rangle \otimes |\psi_1\rangle + \alpha_2|\psi_2\rangle \otimes |\psi_2\rangle \\ &\equiv \alpha_1|\psi_1, \psi_1\rangle + \alpha_2|\psi_2, \psi_2\rangle. \end{aligned} \tag{2.77}$$

Upon simplifying the right-hand side of Eq. (2.76), we obtain

$$\begin{aligned}|\lambda\rangle \otimes |\lambda\rangle &= (\alpha_1|\psi_1\rangle + \alpha_2|\psi_2\rangle)(\alpha_1|\psi_1\rangle + \alpha_2|\psi_2\rangle) \\ &= \alpha_1^2|\psi_1\rangle|\psi_1\rangle + \alpha_1\alpha_2|\psi_1\rangle|\psi_2\rangle + \alpha_1\alpha_2|\psi_2\rangle|\psi_1\rangle + \alpha_2^2|\psi_2\rangle|\psi_2\rangle \\ &\equiv \alpha_1^2|\psi_1,\psi_1\rangle + \alpha_1\alpha_2|\psi_1,\psi_2\rangle + \alpha_1\alpha_2|\psi_2,\psi_1\rangle + \alpha_2^2|\psi_2,\psi_2\rangle. \end{aligned} \tag{2.78}$$

Equating Eq. (2.77) and Eq. (2.78) yields

$$\alpha_1(\alpha_1 - 1)|\psi_1,\psi_1\rangle + \alpha_1\alpha_2|\psi_1,\psi_2\rangle + \alpha_1\alpha_2|\psi_2,\psi_1\rangle + \alpha_2(\alpha_2 - 1)|\psi_2,\psi_2\rangle = 0. \tag{2.79}$$

Equation (2.79) implies that the solution of the equation is given by $\alpha_1, \alpha_2 = \{0, 1\}$, which results in the quantum state $|\psi_1\rangle$ or $|\psi_2\rangle$ itself. We arrive at the conclusion that a quantum-domain copying operation is only capable of cloning the basis states $|\psi_1\rangle$ and $|\psi_2\rangle$, but fails to clone the superposition of the two states $|\lambda\rangle = \alpha_1|\psi_1\rangle + \alpha_2|\psi_2\rangle$.

2.6 Quantum Entanglement

Quantum entanglement is a unique property that only exists in the quantum domain, but it has no counterparts in classical systems. It may be described as a phenomenon where a pair or a group of qubits is generated in such way that the quantum state of the constituent qubits cannot be described indepedently. For instance, let us consider a two-qubit quantum system having a quantum state of $|\psi\rangle$. The quantum state of $|\psi\rangle$ is said to be separable if it can be expressed as a tensor product of two independent quantum states as follows:

$$|\psi\rangle = |\psi_1\rangle \otimes |\psi_2\rangle. \tag{2.80}$$

Otherwise, the quantum state $|\psi\rangle$ is said to be entangled. More explicitly, to provide a clearer picture about quantum entanglement, let us consider an arbitrary two-qubit state as follows:

$$|\psi\rangle = \alpha_{00}|00\rangle + \alpha_{11}|11\rangle, \tag{2.81}$$

where $\alpha_{00}, \alpha_{11} \neq 0$. Let us now consider a quantum state constituted by the superposition of two individual qubits, which may be described as follows:

$$\begin{aligned}|\psi\rangle &= (\alpha_0|0\rangle + \alpha_1|1\rangle) \otimes (\beta_0|0\rangle + \beta_1|1\rangle) \\ &= \alpha_0\beta_0|00\rangle + \alpha_0\beta_1|01\rangle + \alpha_1\beta_0|10\rangle + \alpha_1\beta_1|11\rangle. \end{aligned} \tag{2.82}$$

By equating Eq. (2.81) and Eq. (2.82), we arrive at

$$\alpha_{00}|00\rangle + \alpha_{11}|11\rangle \quad = \quad \alpha_0\beta_0|00\rangle + \alpha_0\beta_1|01\rangle + \alpha_1\beta_0|10\rangle + \alpha_1\beta_1|11\rangle. \tag{2.83}$$

Since the non-zero solution for α_{00}, α_{11}, α_0, α_1, β_0, and β_1 does not exist for Eq. (2.83), it is impossibe to decompose the state of the qubits in Eq. (2.81) into two individual qubit states as

in Eq. (2.82).

To elaborate a little further, let us now consider a very specific quantum state defined as follows:

$$|\Phi^+\rangle = \frac{1}{\sqrt{2}} \left(|00\rangle + |11\rangle\right). \tag{2.84}$$

In this case, if we perform a quantum measurement on the first qubit in Eq. (2.84), we can determine the value of the second qubit immediately with absolute certainty. If the measurement result of the first qubit is 0, the result of measuring the second qubit will also be 0. Similarly, if the measurement result of the first qubit is 1, the result of observing the second qubit will also be 1. Consequently, this inherent correlation between the two qubits having a quantum state of $|\Phi^+\rangle$ is established even if they are separated in space. This interesting phenomenon was given the fond connotation of "spooky action at a distance" by Einstein.

Quantum entanglement enables a pair or a group of qubits to interact over a distance simultaneously, even if they are at the opposite sides of the universe. However, this intriguing fact does not mean that we can communicate faster than the speed of light. Although there is a connection between the entangled qubits, we still do not know what information is going to be transmitted, since we cannot observe the state of entangled qubits before their measurement. Hence, this connection cannot be used for information transmission. This condition is further elaborated on by the *no communication theorem*, which is a specific theorem of quantum information theory. However, the ability of creating entanglement may be exploited for beneficial applications such as for example QECCs [68, 69], quantum key distribution [32], quantum superdense coding [33], quantum teleportation [34], and quantum secure-direct communication [35].

2.7 Quantum Channels

Again, quantum decoherence is a major impediment in the way of realizing practical 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 [198]:

$$\begin{aligned} |e_0\rangle\,|0\rangle &\rightarrow |a_0\rangle\,|0\rangle + |a_1\rangle\,|1\rangle\,, \\ |e_0\rangle\,|1\rangle &\rightarrow |a_2\rangle\,|0\rangle + |a_3\rangle\,|1\rangle\,, \end{aligned} \tag{2.85}$$

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. (2.85) is unitary. Consequently, a qubit in the state $|\psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle$ decoheres as:

$$|e_0\rangle\,|\psi\rangle \rightarrow \alpha\left(|a_0\rangle\,|0\rangle + |a_1\rangle\,|1\rangle\right) + \beta\left(|a_2\rangle\,|0\rangle + |a_3\rangle\,|1\rangle\right). \tag{2.86}$$

Eq. (2.86) can be rearranged as:

$$|e_0\rangle|\psi\rangle \rightarrow \frac{1}{2}\left(|a_0\rangle + |a_3\rangle\right)\left(\alpha|0\rangle + \beta|1\rangle\right) + \frac{1}{2}\left(|a_0\rangle - |a_3\rangle\right)\left(\alpha|0\rangle - \beta|1\rangle\right) + \frac{1}{2}\left(|a_1\rangle + |a_2\rangle\right)\left(\alpha|1\rangle + \beta|0\rangle\right) + \frac{1}{2}\left(|a_1\rangle - |a_2\rangle\right)\left(\alpha|1\rangle - \beta|0\rangle\right), \tag{2.87}$$

which is equivalent to:

$$|e_0\rangle|\psi\rangle \rightarrow \frac{1}{2}\left(|a_0\rangle + |a_3\rangle\right)\mathbf{I}|\psi\rangle + \frac{1}{2}\left(|a_0\rangle - |a_3\rangle\right)\mathbf{Z}|\psi\rangle + \frac{1}{2}\left(|a_1\rangle + |a_2\rangle\right)\mathbf{X}|\psi\rangle + \frac{-i}{2}\left(|a_1\rangle - |a_2\rangle\right)\mathbf{Y}|\psi\rangle. \tag{2.88}$$

Hence, as we may observe in Eq. (2.87), 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. (2.87) 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 1.3. 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 [199]:

- **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:

$$\mathbf{E}_0 = \sqrt{1-p}\,\mathbf{I} = \sqrt{1-p}\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \quad \mathbf{E}_1 = \sqrt{p}\,\mathbf{X} = \sqrt{p}\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \tag{2.89}$$

  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:

$$\mathbf{E}_0 = \sqrt{1-p}\,\mathbf{I} = \sqrt{1-p}\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \quad \mathbf{E}_1 = \sqrt{p}\,\mathbf{Z} = \sqrt{p}\begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}. \tag{2.90}$$

- **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:

$$\mathbf{E}_0 = \sqrt{1-p}\,\mathbf{I} = \sqrt{1-p}\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \qquad \mathbf{E}_1 = \sqrt{p}\,\mathbf{Y} = \sqrt{p}\begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}. \tag{2.91}$$

- **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:

$$\mathbf{E}_0 = \sqrt{1-p}\,\mathbf{I} = \sqrt{1-p}\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \qquad \mathbf{E}_1 = \sqrt{\frac{p}{3}}\,\mathbf{X} = \sqrt{\frac{p}{3}}\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix},$$
$$\mathbf{E}_2 = \sqrt{\frac{p}{3}}\,\mathbf{Z} = \sqrt{\frac{p}{3}}\begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}, \qquad \mathbf{E}_3 = \sqrt{\frac{p}{3}}\,\mathbf{Y} = \sqrt{\frac{p}{3}}\begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}. \tag{2.92}$$

The depolarizing channel of Eq. (2.92) 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 [200, 201].

- **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:

$$\mathbf{E}_0 = \begin{pmatrix} 1 & 0 \\ 0 & \sqrt{1-\gamma} \end{pmatrix}, \qquad \mathbf{E}_1 = \begin{pmatrix} 0 & \sqrt{\gamma} \\ 0 & 0 \end{pmatrix}. \tag{2.93}$$

According to Eq. (2.93), 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:

$$\mathbf{E}_0 = \begin{pmatrix} 1 & 0 \\ 0 & \sqrt{1-\lambda} \end{pmatrix}, \qquad \mathbf{E}_1 = \begin{pmatrix} 0 & 0 \\ 0 & \sqrt{\lambda} \end{pmatrix}, \tag{2.94}$$

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. (2.92) [202, 152, 135, 203] and we we will focus our attention on the depolarizing channel model.

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.	$\begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix}$
Contr.-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 2.1: Summary of the quantum unitary operators of Section 2.1.

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 2.2: Summary of the quantum channel models of Section 2.7.

2.8 Summary and Conclusions

This chapter provides a brief introduction to fundamental quantum information processing. We have highlighted the basic properties of the qubits along with the quantum evolutions acting on single-qubit and multi-qubit systems. We have categorized the quantum evolution into reversible

evolution represented by the unitary transformations and the irreversible evolution constituted by quantum measurements. We have also introduced several quantum gates widely utilized in the development of this treatise. Additionally, we have presented various quantum channel models used for simulating the quantum decoherence imposed on the quantum information. Additionally, we have presented one of the fundamental limitations in the quantum domain, namely the no-cloning theorem, which states that we cannot replicate the quantum state of a qubit in an arbitrary superposition state. Finally, we have described one of the distinctive features of quantum information, namely the ability to create quantum entanglement. The rest of Part 1 will be developed based on the knowledge presented in this chapter.

Chapter 3

From Classical to Quantum Coding

3.1 Introduction

In Chapter 2, we have discussed some of the basic quantum signal processing gates and stated that quantum information suffers from qubit errors. In the classical domain, the deleterious effects of errors imposed by the channel can be mitigated using error correction codes by attaching redundancy to the information part. However, the laws of quantum mechanics prevent us from directly transplanting the classical error correction codes into the quantum domain owing to the following obstacles:

(a) **No-Cloning Theorem**. In the classical domain, the simplest technique of protecting the information bits is based on repetition coding, copying the same information bits several times and then using majority decisions for finding the most likely transmitted codeword. By contrast, in the quantum domain, this simple approach cannot be implemented, since no unitary quantum transformation is capable of performing this specific task. Formally, this is stated by the no-cloning theorem.

(b) **The quantum bit collapses into the corresponding classical bit upon measurement**. In the classical domain, the error correction decoders are typically fed by the bits received at the output of the demodulator. In the quantum domain, measuring the qubits represented by the superposition of the classical states will collapse the superposition into a single classical post-measurement state and consequently, we lose the original quantum information.

(c) **QECCs have to handle not only bit-flip errors but also phase-flip errors, as well as the simultaneous bit-flip and phase-flip errors**. By contrast, in the classical

domain, we deal with a single type of error, which is the bit-flip error. In the quantum domain, the nature of quantum decoherence is continuous and it can be modelled as a linear combination of bit-flip errors ($\mathbf{X}$), phase-flip errors ($\mathbf{Z}$), or both bit-flip and phase-flip errors ($i\mathbf{XZ} = \mathbf{Y}$). However, thanks to the beneficial effect of the stabilizer measurement, the continuous nature of quantum decoherence can be treated as a discrete set of independent errors imposed on the physical qubits.

Albeit all of the aforementioned obstacles hinder the development of QECC schemes, the invention of QSC formulation succeeded in circumventing these problems.

In this chapter, we present the classical-to-quantum isomorphism of the QSCs, demonstrating the duality between the classical and quantum domain. We will also show how to transplant the well-known syndrome-based decoding of classical codes into quantum codes. The rest of this chapter is organized as follows. We commence with a brief review of classical syndrome-based decoding in Section 3.2. This is followed by Section 3.3, where we present the similarities between classical syndrome-based decoding and the quantum stabilizer codes (QSCs). In Section 3.4, we provide the examples for the QSC constructions protecting a single qubit using 1/3-rate quantum repetition code, Shor's code, Steane's code and the perfect 5-qubit code. Finally, we conclude this chapter in Section 3.5.

3.2 A Brief Review of Classical Syndrome-based Decoding

As mentioned earlier, the problems revolving around the QECCs are effectively circumvented by QSCs, which essentially constitute the syndrome-based decoding version of QECCs. Hence, for the sake of shedding some light onto the parallelism between the classical and quantum regime, we proceed with the classical syndrome-based decoding first.

In the classical domain a $\mathcal{C}(n, k)$ code maps k information bits into n coded bits, where $k < n$. The objective of attaching $(n - k)$ redundant bits is to facilitate error detection or even error correction. Let us refer to Fig. 3.1 and consider the classical $\mathcal{C}(7, 4)$ Hamming code, which maps 4 information bits into 7 coded bits and hence becomes capable of correcting a single error. In general, the mapping of the k information bits is performed by multiplying the information row vector $\mathbf{x}$ consisting of k elements by the generator matrix $\mathbf{G}$ having $(k \times n)$ elements. Explicitly, the mapping can be formulated as

$$\mathbf{y} = \mathbf{x} * \mathbf{G}, \tag{3.1}$$

where the resultant codeword $\mathbf{y}$ is a row vector having n elements, while the notation of $*$ represents the matrix multiplication over modulo-2. For instance, the generator matrix of the

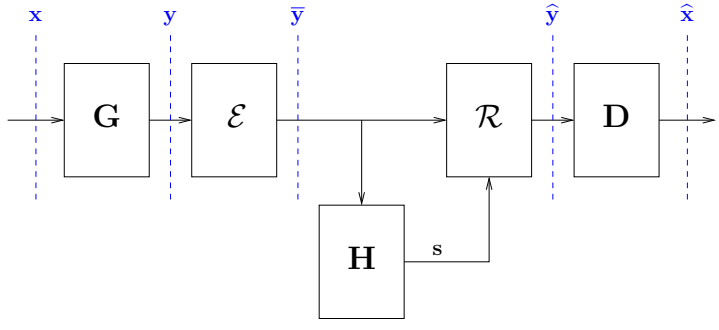


Figure 3.1: The basic model of classical error correction codes invoking syndrome-based decoding. The **G** denotes the generator matrix, which maps the k information bits $\mathbf{x}$ to the n coded bits $\mathbf{y}$. The channel $\mathcal{E}$ inflicts an error vector $\mathbf{e} \in \{0,1\}^n$ to the codeword $\mathbf{y}$ resulting the corrupted received bits $\overline{\mathbf{y}}$. The receiver calculates the syndrome vector $\mathbf{s}$ based on the PCM $\mathbf{H}$ and received bits $\overline{\mathbf{y}}$ to predict the number and the position of errors contained in the received bits $\overline{\mathbf{y}}$. The error recovery $\mathcal{R}$ generates the error recovery vector $\mathbf{r}$, which is applied to the received bits $\overline{\mathbf{y}}$. It collapses the received bits $\overline{\mathbf{y}}$ to one of the legitimate codedword $\mathbf{y}$ yielding the predicted codeword $\widehat{\mathbf{y}}$. Finally, we can readily determine the predicted information bits $\widehat{\mathbf{x}}$ from the predicted codeword $\widehat{\mathbf{y}}$. ©Chandra *et al.* [174]

$\mathcal{C}(7,4)$ Hamming code is defined by

$$\mathbf{G}_{\text{Hamming}} = \begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{pmatrix}. \tag{3.2}$$

From Eq. (3.1) and (3.2) we can generate the code space mapping shown in Table 3.1, where $\mathbf{x}_i$ denotes all the possible combinations of the information bits and $\mathbf{y}_i$ represents the associated legitimate codeword bits.

The generator matrix $\mathbf{G}$ can be arranged into a systematic form as

$$\mathbf{G} = (\mathbf{I}_k | \mathbf{P}), \tag{3.3}$$

where $\mathbf{I}_k$ is a $(k \times k)$ identity matrix and $\mathbf{P}$ is a matrix having $k \times (n-k)$ elements. The form given in Eq. (3.3) generates a systematic codeword $\mathbf{y}$ consisting of the k-bit information word $\mathbf{x}$ followed by $(n-k)$ parity bits.

As an example, the encoder has to carry out the following operations for the 4-bit original

Table 3.1: The code space mapping of the $\mathcal{C}(7,4)$ classical Hamming code.

i	$\mathbf{x}_i$	$\mathbf{y}_i$
1	0 0 0 0	0 0 0 0 0 0 0
2	0 0 0 1	0 0 0 1 1 1 1
3	0 0 1 0	0 0 1 0 0 1 1
4	0 0 1 1	0 0 1 1 1 0 0
5	0 1 0 0	0 1 0 0 1 0 1
6	0 1 0 1	0 1 0 1 0 1 0
7	0 1 1 0	0 1 1 0 1 1 0
8	0 1 1 1	0 1 1 1 0 0 1
9	1 0 0 0	1 0 0 0 1 1 0
10	1 0 0 1	1 0 0 1 0 0 1
11	1 0 1 0	1 0 1 0 1 0 1
12	1 0 1 1	1 0 1 1 0 1 0
13	1 1 0 0	1 1 0 0 0 1 1
14	1 1 0 1	1 1 0 1 1 0 0
15	1 1 1 0	1 1 1 0 0 0 0
16	1 1 1 1	1 1 1 1 1 1 1

information segment for generating the 7-bit encoded codeword:

$$\mathbf{y} = \mathbf{x} \cdot \mathbf{G} = [0001] \cdot \left(\begin{array}{cccc|ccc} 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{array} \right). \tag{3.4}$$

Explicitly, we write down the element-wise product as a sum and then take the mod-2, as detailed below:

- 0+0+0+0 (mod 2)=0
- 0+0+0+0 (mod 2)=0
- 0+0+0+0 (mod 2)=0
- 0+0+0+1 (mod 2)=1
- 0+0+0+1 (mod 2)=1
- 0+0+0+1 (mod 2)=1
- 0+0+0+1 (mod 2)=1

A generator matrix $\mathbf{G}$ is associated with an $(n-k) \times n$-element parity-check matrix (PCM) $\mathbf{H}$, which is defined as

$$\mathbf{H} = \left(\mathbf{P}^T | \mathbf{I}_{n-k} \right). \tag{3.5}$$

As an example, the generator matrix of the classical $\mathcal{C}(7,4)$ Hamming code of Eq. (3.2) is associated with the following PCM:

$$\mathbf{H}_{\text{Hamming}} = \begin{pmatrix} 1 & 1 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 1 \end{pmatrix}. \tag{3.6}$$

The PCM of $\mathbf{H}$ is constructed for ensuring that a valid codeword $\mathbf{y}$ satisfies the following requirement:

$$\mathbf{y} * \mathbf{H}^T = \mathbf{0}. \tag{3.7}$$

A received word $\overline{\mathbf{y}}$ may be contaminated by an error vector $\mathbf{e} \in \{0,1\}^n$ due to channel impairments, which is denoted by $\mathcal{E}$ in Fig. 3.1. More explicitly, the resultant received words corrupted by the additive noise $\mathcal{E}$ can be formulated as

$$\overline{\mathbf{y}} = \mathbf{y} + \mathbf{e}. \tag{3.8}$$

The error syndrome $\mathbf{s}$ is a row vector having $(n-k)$ elements obtained by the following calculation:

$$\begin{aligned} \mathbf{s} = \overline{\mathbf{y}} * \mathbf{H}^T &= (\mathbf{y} + \mathbf{e}) * \mathbf{H}^T \\ &= \mathbf{y} * \mathbf{H}^T + \mathbf{e} * \mathbf{H}^T \\ &= \mathbf{0} + \mathbf{e} * \mathbf{H}^T \\ &= \mathbf{e} * \mathbf{H}^T. \end{aligned} \tag{3.9}$$

The syndrome vector $\mathbf{s}$ contains the information related to the error pattern imposed by the channel. To elaborate, we have 2^k legitimate codewords generated by all the possible combinations of the k information bits, 2^n possible received bit patterns of $\widehat{\mathbf{y}}$ and $2^{(n-k)}$ syndromes $\mathbf{s}$, each unambiguously identifying one of the $2^{(n-k)}$ error patterns, including the error-free scenario.

Hence, for the classical $\mathcal{C}(7,4)$ Hamming code, the syndrome vector $\mathbf{s}_i$ can uniquely and unambiguously detect and correct one of the seven single-bit error patterns, plus the error-free scenario, as specified in Table 3.2.

For example, for the error pattern of $e^T = [0000001]$ the syndrom decoder carries out the

following operations:

$$\mathbf{s} = \mathbf{e} \cdot \mathbf{H}^T = [0000001] \cdot \begin{pmatrix} \mathbf{1} & 1 & 0 \\ \mathbf{1} & 0 & 1 \\ \mathbf{0} & 1 & 1 \\ \mathbf{1} & 1 & 1 \\ \mathbf{1} & 0 & 0 \\ \mathbf{0} & 1 & 0 \\ \mathbf{0} & 0 & 1 \end{pmatrix} = [001]. \tag{3.10}$$

The error recovery $\mathbf{r}_i$ is determined based on the most likely error pattern. After obtaining the syndrome vector, the recovery vector $\mathbf{r}_i$ is applied to the received words to obtain the predicted codeword $\widehat{\mathbf{y}} = \overline{\mathbf{y}} + \mathbf{r}$, as depicted in Fig. 3.1. The application of the recovery operator to the received word always collapses it into one of the legitimate codewords $\mathbf{y}$, hence the predicted codeword $\widehat{\mathbf{y}}$ can be finally demapped in order to obtain the predicted information bits $\widehat{\mathbf{x}}$ using Table 3.1, as illustrated in Fig. 3.1. For linear systematic codes, this process can be simply performed by chopping the last $(n-k)$ bits, namely the redundant bits.

Table 3.2: The look-up table to determine the most likely error pattern $\mathbf{e}_i \in \mathcal{E}$ that corresponds to the syndrome value $\mathbf{s}_i$, which is created based on Eq. (3.6) and (3.9).

i	$\mathbf{s}_i$	$\mathbf{e}_i$
1	0 0 0	0 0 0 0 0 0 0
2	0 0 1	0 0 0 0 0 0 1
3	0 1 0	0 0 0 0 0 1 0
4	0 1 1	0 0 1 0 0 0 0
5	1 0 0	0 0 0 0 1 0 0
6	1 0 1	0 1 0 0 0 0 0
7	1 1 0	1 0 0 0 0 0 0
8	1 1 1	0 0 0 1 0 0 0

For another detailed example let us consider k information bits of $\mathbf{x} = (1\ 1\ 0\ 1)$. The information bits are encoded using the classical $\mathcal{C}(7,4)$ Hamming code employing the generator matrix of Eq. (3.1), yielding the coded bits of $\mathbf{y} = (1\ 1\ 0\ 1\ 1\ 0\ 0)$. Let us assume that the channel corrupts the legitimate codeword $\mathbf{y}$ by imposing an error pattern of $\mathbf{e} = (1\ 0\ 0\ 0\ 0\ 0\ 0)$ resulting in the received word of $\overline{\mathbf{y}} = (0\ 1\ 0\ 1\ 1\ 0\ 0)$. Next, the received word is fed to the syndrome calculation block, which contains the PCM of Eq. (3.6). Based on Eq. (3.9), the received word $\overline{\mathbf{y}} = (0\ 1\ 0\ 1\ 1\ 0\ 0)$ generates the syndrome vector of $\mathbf{s} = (1\ 1\ 0)$. Utilizing the look-up table of Table 3.2, the error recovery vector becomes $\mathbf{r} = (1\ 0\ 0\ 0\ 0\ 0\ 0)$. Upon applying the error recovery vector, the received word $\overline{\mathbf{y}}$ is collapsed to one of the legitimate codewords $\mathbf{y}$ in Table 3.1, which is $\widehat{\mathbf{y}} = (1\ 1\ 0\ 1\ 1\ 0\ 0)$. Assuming that the predicted codeword $\widehat{\mathbf{y}}$ is valid,

the demapper decides to translate the predicted codeword $\widehat{\mathbf{y}} = (1\ 1\ 0\ 1\ 1\ 0\ 0)$ to the predicted information bits as $\widehat{\mathbf{x}} = (1\ 1\ 0\ 1)$. Hence, the original information is successfully recovered. The whole process of syndrome calculation, error recovery and demapping jointly form the *decoding* process. It is important to note that in practice, the syndrome calculation, recovery operator and demapper are amalgamated into a single *decoder* block.

Let us now assume that the channel imposes an error pattern beyond the error correction capability of the classical $\mathcal{C}(7,4)$ Hamming code. For example, assume that we send the identical k information bits of $\mathbf{x} = (1\ 1\ 0\ 1)$ while the channel inflicts an error pattern of $\mathbf{e} = (1\ 1\ 0\ 0\ 0\ 0\ 0)$. As a result, we have the received codeword bits of $\overline{\mathbf{y}} = (0\ 0\ 0\ 1\ 1\ 0\ 0)$. Based on the received codeword, we have the syndrome vector of $\mathbf{s} = (0\ 1\ 1)$. Based on the syndrome vector, the error recovery of $\mathbf{r} = (0\ 0\ 1\ 0\ 0\ 0\ 0)$ is chosen. Consequently, the error recovery vector collapses the received word to the incorrect legitimate codeword, which is $\widehat{\mathbf{y}} = (0\ 0\ 1\ 1\ 1\ 0\ 0)$, instead of the correct codeword of $\mathbf{y} = (1\ 1\ 0\ 1\ 1\ 0\ 0)$. Since the demapper assumes that the error recovery completes the task perfectly, the demapper decides that the predicted information bits are $\mathbf{x} = (0\ 0\ 1\ 1)$. As a result of this erroneous decoding action we end up having three, rather than two error. This example demonstrates that the classical $\mathcal{C}(7,4)$ Hamming code is unable to operate flawlessly beyond its error correction capability.

3.3 A Brief Review of Quantum Stabilizer Codes

The formulation of QSCs is capable of detecting both the number and the position of errors without actually observing the state of physical qubits, which is vitally important, since otherwise the quantum state will collapse to classical bits upon measurement. This was achieved by amalgamating the classical syndrome-based decoding with the QECCs. Similar to classical error correction codes, QSCs also rely on attaching redundant qubits to the information qubits for invoking error correction. The basic model of QSCs is depicted in Fig. 3.2, which will be contrasted to its classical pair in Fig. 3.1. To generate the codespace $\mathcal{C}$, the redundancy is constituted by $(n-k)$ auxiliary qubits. Next, a unitary transformation $\mathcal{V}$ transforms the k qubits in the state of $|\psi\rangle$ and the $(n-k)$ auxiliary qubits into n qubits in the state of $|\overline{\psi}\rangle$. The unitary transformation of $\mathcal{V}$ represents the action of the *quantum encoder*. Explicitly, the mapping of the *logical qubits* constituting the state of $|\psi\rangle \in \mathbb{C}^{2^k}$ to the *physical qubits* forming the state of $|\overline{\psi}\rangle \in \mathbb{C}^{2^n}$ by the encoder $\mathcal{V}$ of Fig. 3.2 can be mathematically formulated as follows:

$$\mathcal{C} = \{|\overline{\psi}\rangle = \mathcal{V}(|\psi\rangle \otimes |0\rangle^{\otimes(n-k)})\}. \tag{3.11}$$

The QSCs rely on the stabilizer operators $S_i \in \mathcal{S}$ for identifying the type, the number and also the position of the qubit errors. A stabilizer operator S_i is an n-tuple Pauli operator, which preserves the state of physical qubits as defined below:

$$S_i|\overline{\psi}\rangle = |\overline{\psi}\rangle. \tag{3.12}$$

The quantum channel inflicts errors represented by n-tuple Pauli operators $P \in \mathcal{P}_n$, which

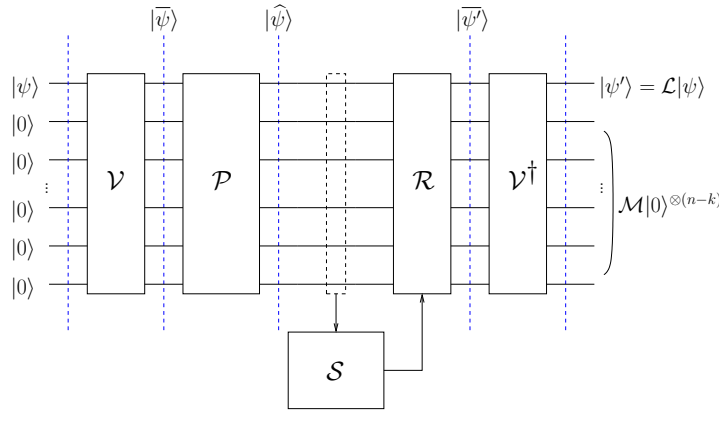


Figure 3.2: The basic model of QSCs implementation over the quantum depolarizing channel. The k logical qubits is mapped into n physical qubits with the aid of $(n-k)$ redundant/auxiliarry qubits (*ancillas*) to provide protection from the quantum decoherence. It is similar to the classical error correction model where $(n-k)$ redundant bits are added to k information bits in order to provide error correction. The quantum encoder $\mathcal{V}$ serves the same purpose with $\mathbf{G}$ of classical error correction codes in Fig. 3.1. The quantum encoder $\mathcal{V}$ transforms the state of k logical qubits $|\psi\rangle$ into the state of n physical qubits $|\overline{\psi}\rangle$ with the aid of $(n-k)$ ancillas. The quantum depolarizing channel imposes vector rerpesented by n-tupple Pauli operator $\mathcal{P} \in \mathcal{P}_n$. The syndrome operators $S_i \in \mathcal{S}$ generate eigenvalues of ± 1, which are analogue to the value 0 and 1 of classical syndrome vector, which is provided by the PCM $\mathbf{H}$ in Fig. 3.1. The error recovery $\mathcal{R}$ applies the correction according to the syndrome values provided by the syndrome measurements. Finally, the quantum inverse encoder $\mathcal{V}^\dagger$ transforms the predicted state of physical qubits $|\overline{\psi'}\rangle$ back to the predicted state of logical qubits $|\psi'\rangle$, which bears the same responsibility as the demapper $\mathbf{D}$ in classical syndrome-based decoding of Fig. 3.1. ©Chandra *et al.* [174]

transforms the encoded physical qubits that were originally in the state of $|\overline{\psi}\rangle$ to the potentially corrupted physical qubits in the state of $|\widehat{\psi}\rangle$, as seen in Fig. 3.2. More explicitly, this process can be described as follows:

$$|\widehat{\psi}\rangle = P|\overline{\psi}\rangle. \tag{3.13}$$

The stabilizer operators act similarly to the syndrome calculations routinely used in classical error correction codes. To elaborate a little further, a stabilizer operator will return an eigenvalue of $+1$, when an error operator P commutes with the stabilizer operator, while we arrive at the eigenvalue of -1, if it anti-commutes. The eigenvalues of $+1$ and -1 are analogous to the classic syndrome bit of 0 and 1, respectively, which can be defined as follows:

$$S_i|\widehat{\psi}\rangle = \begin{cases} |\widehat{\psi}\rangle & , \quad S_iP = PS_i \\ -|\widehat{\psi}\rangle & , \quad S_iP = -PS_i. \end{cases} \tag{3.14}$$

Therefore, the stabilizer operators naturally have to inherit the commutative property. Consequently, the product between the stabilizer operators S_i yields another legitimate stabilizer

operator. Furthermore, the commutativity property implies that

$$S_i|\overline{\psi}\rangle = S_j|\overline{\psi}\rangle = S_iS_j|\overline{\psi}\rangle = |\overline{\psi}\rangle, \forall S_{i,j} \in \mathcal{S}, \tag{3.15}$$

suggesting that the stabilizer group $\mathcal{S}$ is closed under multiplication.

Based on the syndrome measurement by the stabilizer operators S_i, a recovery operator constituted by the n-tupple Pauli operator of $\mathcal{R} \in \mathcal{P}_n$ seen in Fig. 3.2 is applied to the corrupted physical qubit state $|\widehat{\psi}\rangle$, yielding the predicted state of the original encoded logical qubit $|\overline{\psi'}\rangle$, which is formulated as

$$|\overline{\psi'}\rangle = \mathcal{R}|\widehat{\psi}\rangle. \tag{3.16}$$

Finally, the inverse encoder $\mathcal{V}^\dagger$ of Fig. 3.2 performs the following transformation[1]:

$$\begin{aligned} \mathcal{V}^\dagger|\overline{\psi'}\rangle &= \mathcal{V}^\dagger\mathcal{R}|\widehat{\psi}\rangle \\ &= \mathcal{V}^\dagger\mathcal{R}\mathcal{P}|\overline{\psi}\rangle \\ &= \mathcal{V}^\dagger\mathcal{R}\mathcal{P}\mathcal{V}(|\psi\rangle \otimes |0\rangle^{\otimes(n-k)}) \\ &= (\mathcal{L}|\psi\rangle) \otimes (\mathcal{M}|0\rangle^{\otimes(n-k)}), \end{aligned} \tag{3.17}$$

where we have $\mathcal{V}^\dagger\mathcal{R}\mathcal{P}\mathcal{V} \equiv \mathcal{L} \otimes \mathcal{M}$ and $\mathcal{L} \in \mathcal{P}_k$ represents the error inflicted on the logical qubits according to $|\psi'\rangle = \mathcal{L}|\psi\rangle$, while $\mathcal{M} \in \mathcal{P}_{n-k}$ represents the residual errors that remained in the $(n-k)$ auxiliary qubits after the error correction procedure. In the case of $\mathcal{R} = \mathcal{P}$, we arrive at $\mathcal{R}\mathcal{P} = \mathbf{I}^{\otimes n}$, where $\mathbf{I}^{\otimes n}$ denotes an n-fold tensor product Pauli-$\mathbf{I}$ matrix. Another possibility is to arrive at $\mathcal{R}\mathcal{P} = S_i$. In either of these cases, the state of the physical qubits is not altered, since we have $\mathcal{R}\mathcal{P}|\overline{\psi}\rangle = |\overline{\psi}\rangle$. Therefore, the decoding procedure of Fig. 3.2 successfully recovers the original quantum state constituted by the logical qubits, yielding $|\psi'\rangle = |\psi\rangle$.

The stabilizer operators can be translated into the classical PCM $\mathbf{H}$ by mapping the Pauli matrices $\mathbf{I}$, $\mathbf{X}$, $\mathbf{Y}$ and $\mathbf{Z}$ onto $(\mathbb{F}_2)^2$ as follows:

$$\begin{aligned} \mathbf{I} &\rightarrow \left(\begin{array}{c|c} 0 & 0 \end{array} \right), \\ \mathbf{X} &\rightarrow \left(\begin{array}{c|c} 0 & 1 \end{array} \right), \\ \mathbf{Y} &\rightarrow \left(\begin{array}{c|c} 1 & 1 \end{array} \right), \\ \mathbf{Z} &\rightarrow \left(\begin{array}{c|c} 1 & 0 \end{array} \right). \end{aligned} \tag{3.18}$$

This concept is also known as the *Pauli-to-binary isomorphism*. By exploiting the Pauli-to-binary isomorphism, the stabilizer operators of any QSC can be represented as a pair of PCMs $\mathbf{H}_z$ and $\mathbf{H}_z$, where $\mathbf{H}_z$ is invoked for handling the phase-flip ($\mathbf{Z}$) errors and $\mathbf{H}_x$ for handling the bit-flip ($\mathbf{X}$) errors. Explicitly, the classical PCM representation of the QSC stabilizer operators may be written as follows:

$$\mathbf{H} = (\mathbf{H}_z|\mathbf{H}_x). \tag{3.19}$$

[1]The inverse encoder $\mathcal{V}^\dagger$ is the Hermitian transpose of encoder $\mathcal{V}$. It is referred to as the *inverse*, since it satisfies the unitary requirement of $\mathcal{V}^\dagger\mathcal{V} = \mathbf{I}$, as the inverse of the matrix does.

The classical representation of the stabilizer operators gives the advantage of predicting and evaluating the performances of QSCs by treating them similarly to classical error correction codes. Additionally, it allows us to transform a pair of classical PCMs into the correponding quantum counterpart. However, to ensure that the commutative property is preserved in the quantum domain, a pair of classical PCMs have to satisfy the so-called *symplectic criterion* [64] given by

$$\mathbf{H}_z \cdot \mathbf{H}_x^T + \mathbf{H}_x \cdot \mathbf{H}_z^T = 0. \tag{3.20}$$

A special class of QSCs, namely the family of Calderbank-Shore-Steane (CSS) codes, treats the phase-flip ($\mathbf{Z}$) and bit-flip ($\mathbf{X}$) errors as two separate entities. More specifically, this can be interpreted as having the PCMs of $\mathbf{H}_z$ and $\mathbf{H}_x$ in Eq. (3.19) formulated as $\mathbf{H}_z = \begin{pmatrix} \mathbf{H}'_z \\ 0 \end{pmatrix}$ and $\mathbf{H}_x = \begin{pmatrix} 0 \\ \mathbf{H}'_x \end{pmatrix}$, respectively. Therefore, the binary PCM $\mathbf{H}$ can be expressed as follows:

$$\mathbf{H} = \left(\begin{array}{c|c} \mathbf{H}'_z & \mathbf{0} \\ \mathbf{0} & \mathbf{H}'_x \end{array} \right). \tag{3.21}$$

Consequently, the symplectic criterion given in Eq. (3.20) can be reduced to the following criterion:

$$\mathbf{H}'_z \cdot \mathbf{H}'^{\,T}_x = 0. \tag{3.22}$$

Furthermore, we can formulate a CSS code by using a PCM of $\mathbf{H}'_z = \mathbf{H}'_x$ and the resultant quantum code may be referred to as a dual-containing quantum CSS code or self-orthogonal quantum CSS code. For dual-containing CSS codes, the symplectic criterion can be further simplified to $\mathbf{H}'_z \mathbf{H}'^{\,T}_z = 0$.

Again, the classical code constructions can be readily transformed into their quantum version provided that they satisfy the symplectic criterion of Eq. (3.20). In other words, unless the symplectic condition is satisfied by them, some of the popular classical codes cannot be 'transplanted' into the quantum domain. However, fortunately, this limitation can be relaxed by utilizing the family of EA-QSCs [68, 69]. To elaborate a little further, transforming a classical codes into a QSC does not come without cost. Invoking the above-mentioned EA-QSC construction requires so-called preshared maximally-entangled qubits before the encoding procedure, as detailed in [69]. However, the mechanism of presharing the maximally-entangled qubits allows us to transform a set of non-symplectic QSCs into their symplectic counterpart. In closing this section we note that the classification of QSCs is summarized at a glance in Fig. 3.3.

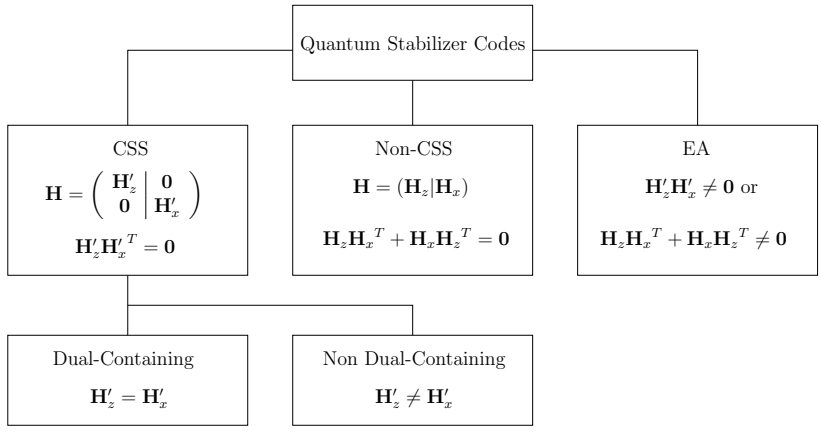


Figure 3.3: The classification and characterization of QSCs, where CSS stands for *Calderbank-Shor-Steane* and EA for *entanglement-assisted*. ©Chandra *et al.* [204]

3.4 Protecting A Single Qubit: Design Examples

In Chapter 1, we have already mentioned the three pioneering contributions on QSCs, which are only capable of handling a single qubit error, while in Section 3.3, we briefly highlighted the different types of QSC constructions. In this section, we will link up both ideas in a more concrete context.

3.4.1 Classical and Quantum 1/3-rate Repetition Codes

Before we delve deeper into the aforementioned QSCs, let us commence with a simple 1/3-rate classical repetition codes, which maps a binary digit of "0" or "1" into a vector that contains three replicas of each binary digit as

$$\begin{aligned} 0 &\xrightarrow{\mathbf{G}} \begin{pmatrix} 0 & 0 & 0 \end{pmatrix}, \\ 1 &\xrightarrow{\mathbf{G}} \begin{pmatrix} 1 & 1 & 1 \end{pmatrix}. \end{aligned} \tag{3.23}$$

To elaborate briefly, we have created two legitimate codewords, which have a Hamming distance of three. Given the three-bit codewords, we can have eight different received words. Two of these are legitimate error-free codewords and six of them are erroneous. However, even if one of the bits is corrupted, majority logic allows us to correct it. By contrast, if two bits are corrupted, the majority logic decision would opt for the wrong legitimate codeword, because more errors were imposed by the channel than the error correction capability of code. This situation is often referred to as the code being overloaded by an excessive number of errors.

Let us now briefly portray the task of the decoder from a syndrom decoding - rather than majority logic - perspective. If the first and second bit are different, we know that one of them is corrupted, bit it is not clear at this stage, which one. So this is the first symptom of having

an error. This simple calculation provides us with the first syndrom bit, which is a logical one. Similarly, if the first and third bit are different, it is not clear, which of these two bits is corrupted, but we have a second syndrom bit. Simple logic dictates that this is only possible, if the first bit was corrupted. This can hence be inverted for correcting this 3-bit codeword. But again, if there are two or three errors, this 1/3-rate code having a minimum distance of three between the legitimate codewordsbecome overwhelmed. The above pair of syndrom bits allow us to distinguish four possible scenarios, namely an error-free one and having a single error in any of the three positions.

Hence, from the brief description of the basic classical codes given in Section 3.2, the mapping in Eq. (3.23) can be encapsulated into a generator matrix $\mathbf{G}$ as given below:

$$\mathbf{G} = \begin{pmatrix} 1 & 1 & 1 \end{pmatrix}. \tag{3.24}$$

From the generator matrix $\mathbf{G}$ given in Eq. (3.24) and the PCM formulation given in Eq. (3.3), we obtain the PCM $\mathbf{H}$ for a 1/3-rate classical repetition code encapsulated by

$$\mathbf{H} = \begin{pmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \end{pmatrix}, \tag{3.25}$$

where the first row returns the first bit of the two-bit syndrome and acccordingly the second row evaluates the second bit. Thus, it can be easily checked by using the syndrome computation of Eq. (3.9) that the syndrome value of (0 0) is obtained if the received word $\widehat{\mathbf{y}}$ is equal to the valid codeword, either (0 0 0) or (1 1 1). The syndrome computation yields a syndrome vector with $(n-k)$-element and in this case for a 1/3-rate classical repetition code, it generates a syndrome vector with two elements. Therefore, again, there are four possible outcomes from the syndrome computation and one of them indicates the error-free received word, which is the (0 0) syndrome. Since a 1/3-rate classical repetition code may be viewed as a short block code, the syndrome computation and the associated error pattern is readily checked using a look-up table, namely Table. 3.3.

Table 3.3: Syndrome computation and the associated error pattern for a 1/3-rate classical repetition code.

Syndrome ($\mathbf{s}$)	Error Pattern ($\mathbf{e}$)	Index of Corrupted Bit
(0 0)	(0 0 0)	-
(0 1)	(0 0 1)	3
(1 0)	(0 1 0)	2
(1 1)	(1 0 0)	1

Next, we proceed with with a simple 1/3-rate quantum repetition code that is capable of recovering a single bit-flip, but no phase-flip. Let us assume that we have a quantum state $|\psi\rangle = \alpha_0|0\rangle + \alpha_1|1\rangle$. As a consequence of the *No Cloning Theorem* of quantum mechanics, there is no unitary transformation U capable of mapping an *arbitrary* quantum state $|\psi\rangle$ onto a state

of $|\overline{\psi}\rangle = |\psi\rangle^{\otimes 3}$. However, we are still allowed to make a copy of the *orthogonal* states. Hence, the mapping of the *orthogonal* quantum states $|0\rangle$ and $|1\rangle$ may be carried out by a unitary transformation U as follows:

$$\begin{aligned} |0\rangle_L &= U\left(|0\rangle \otimes |0\rangle^{\otimes 2}\right) = |000\rangle, \\ |1\rangle_L &= U\left(|1\rangle \otimes |0\rangle^{\otimes 2}\right) = |111\rangle. \end{aligned} \tag{3.26}$$

In a more general scenario, the mapping of a single logical qubit to n physical qubits is arranged by a unitary transformation carried out by a quantum encoder $\mathcal{V}$, which can be formulated as follows:

$$|\overline{\psi}\rangle = \mathcal{V}\left(|\psi\rangle \otimes |0\rangle^{\otimes(n-1)}\right) = \alpha_0|0\rangle_L + \alpha_1|1\rangle_L, \tag{3.27}$$

where $|0\rangle_L$ denotes the encoded state of the logical qubit $|0\rangle$, $|1\rangle_L$ denotes the encoded state of the logical qubit $|1\rangle$, while $|0\rangle^{\otimes n-1}$ represents the auxiliary or redundant qubits (*also referred to as ancillas*), and the superscript of $\otimes(n-1)$ represents $(n-1)$-fold tensor products. Hence, for 1/3-rate quantum repetition codes, the state of the logical qubit $|\psi\rangle$ corresponds to the state of the physical qubit $|\overline{\psi}\rangle$ as given by

$$|\overline{\psi}\rangle = \mathcal{V}\left((\alpha_0|0\rangle + \alpha_1|1\rangle) \otimes |0\rangle^{\otimes 2}\right) = \alpha_0|000\rangle + \alpha_1|111\rangle, \tag{3.28}$$

where the string $|000\rangle$ defines the encoded logical qubit $|0\rangle_L$ and $|111\rangle$ defines $|1\rangle_L$. Again, it is important to bear in mind that the state of $|\overline{\psi}\rangle = \alpha_0|000\rangle + \alpha_1|111\rangle$ is not equal to $|\overline{\psi}\rangle = |\psi\rangle^{\otimes 3}$. More explicitly, this relationship can also be expressed as $|\overline{\psi}\rangle = \alpha_0|000\rangle + \alpha_1|111\rangle \neq |\psi\rangle^{\otimes 3}$. The state of the physical qubits of the 1/3-rate quantum repetition code is stabilized, or synonymously 'parity-checked' by the pair of stabilizer operators $S_1 = \mathbf{ZZI}$ and $S_2 = \mathbf{ZIZ}$. A valid codeword or a valid encoded state, which is not affected by the stabilizer operators S_1 and S_2, has an input state of $|\overline{\psi}\rangle$ and returns the state of $|\overline{\psi}\rangle$, hence it yields the so-called eigenvalues of $+1$, and more explicitly, it is described below:

$$\begin{aligned} S_1|\overline{\psi}\rangle &= \alpha_0|000\rangle + \alpha_1|111\rangle \equiv |\overline{\psi}\rangle, \\ S_2|\overline{\psi}\rangle &= \alpha_0|000\rangle + \alpha_1|111\rangle \equiv |\overline{\psi}\rangle. \end{aligned} \tag{3.29}$$

By contrast, if the stabilizer operators g_1 and g_2 are applied to the corrupted states $|\widehat{\psi}\rangle$, they both yield eigenvalues that are not in the all-one state. For instance, let us assume that we received a corrupted state having a bit-flip error imposed on the first qubit of $|\overline{\psi}\rangle$ yielding $|\widehat{\psi}\rangle = \alpha_0|100\rangle + \alpha_1|011\rangle$. Then, upon applying the stabilizer operators $S_1 = \mathbf{ZZI}$ and $S_2 = \mathbf{ZIZ}$ to the state of $|\widehat{\psi}\rangle$, it may be readily shown after few steps that we arrive at the following

eigenvalues:

$$
\begin{aligned}
S_1|\widehat{\psi}\rangle &= \mathbf{ZZI}(\alpha_0|100\rangle + \alpha_1|011\rangle) \\
&= \begin{pmatrix}
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1
\end{pmatrix} \cdot \begin{pmatrix} 0 \\ 0 \\ 0 \\ \alpha_1 \\ \alpha_0 \\ 0 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ -\alpha_1 \\ -\alpha_0 \\ 0 \\ 0 \\ 0 \end{pmatrix} \\
&\equiv -\alpha_0|100\rangle - \alpha_1|011\rangle \equiv -|\widehat{\psi}\rangle, \qquad (3.30)
\end{aligned}
$$

$$
\begin{aligned}
S_2|\widehat{\psi}\rangle &= \mathbf{ZIZ}(\alpha_0|100\rangle + \alpha_1|011\rangle) \\
&= \begin{pmatrix}
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1
\end{pmatrix} \cdot \begin{pmatrix} 0 \\ 0 \\ 0 \\ \alpha_1 \\ \alpha_0 \\ 0 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ -\alpha_1 \\ -\alpha_0 \\ 0 \\ 0 \\ 0 \end{pmatrix} \\
&\equiv -\alpha_0|100\rangle - \alpha_1|011\rangle \equiv -|\widehat{\psi}\rangle. \qquad (3.31)
\end{aligned}
$$

The resultant eigenvalues of ± 1 act similarly to the syndrome vector of classical codes, where the eigenvalue $+1$ is associated with the classical syndrome value 0 and the eigenvalue -1 with the classical syndrome value 1. More explicitly, the single-qubit error patterns imposed on the 1/3-rate quantum repetition codes and the associated eigenvalues are portrayed in Table. 3.4. However, this specific construction is only capable of detecting and correcting a single bit-flip error imposed by the Pauli channel on the physical qubits, but no phase-flips.

Since the physical qubits may experience not only bit-flip errors, but also phase-flip errors as well as both bit-flip and phase-flip errors, a different mapping is necessitated to protect the

Table 3.4: Single qubit bit-flip errors along with the associated eigenvalues in 1/3-rate quantum repetition where the eigenvalues act similarly with the syndrome values in classical linear block codes.

Received States $\|\widehat{\psi}\rangle$	Eigenvalue $g_1\|\widehat{\psi}\rangle$	Eigenvalue $g_2\|\widehat{\psi}\rangle$	Syndrome ($\mathbf{s}$)	Index of Corrupted Qubit
$\alpha_0\|000\rangle + \alpha_1\|111\rangle$	$+1$	$+1$	(0 0)	-
$\alpha_0\|001\rangle + \alpha_1\|110\rangle$	$+1$	-1	(0 1)	3
$\alpha_0\|010\rangle + \alpha_1\|101\rangle$	-1	$+1$	(1 0)	2
$\alpha_0\|100\rangle + \alpha_1\|011\rangle$	-1	-1	(1 1)	1

physical qubits from phase-flip error. In order to protect the physical qubits from a phase-flip error, we may require a different basis but we can still invoke a similar approach. To elaborate further, the Hadamard transformation ($\mathbf{H}$) maps the computational basis of $\{|0\rangle, |1\rangle\}$ onto the Hadamard basis of $\{|+\rangle, |-\rangle\}$, where the states of $|+\rangle$ and $|-\rangle$ are defined as

$$|+\rangle \equiv \mathbf{H}|0\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle), \tag{3.32}$$

$$|-\rangle \equiv \mathbf{H}|1\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle), \tag{3.33}$$

and the unitary Hadamard transformation $\mathbf{H}$, which acts on a single qubit state, is given by

$$\mathbf{H} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}. \tag{3.34}$$

A phase-flip error defined over the Hadamard basis of $\{|+\rangle, |-\rangle\}$ acts similarly to the bit-flip error defined over the computational basis of $\{|0\rangle, |1\rangle\}$. Hence, for the handling of a single phase-flip error, the encoder of the 1/3-rate quantum repetition code protecting against phase-flips carries out:

$$\begin{aligned} |0\rangle_L &= U\left(|0\rangle \otimes |0\rangle^{\otimes 2}\right) = |++\!+\rangle, \\ |1\rangle_L &= U\left(|1\rangle \otimes |0\rangle^{\otimes 2}\right) = |--\!-\rangle. \end{aligned} \tag{3.35}$$

Therefore, the logical qubit of $|\psi\rangle$ corresponding to the physical qubits $|\overline{\psi}\rangle$ is given by

$$|\overline{\psi}\rangle = \mathcal{V}\left((\alpha_0|0\rangle + \alpha_1|1\rangle) \otimes |0\rangle^{\otimes 2}\right) = \alpha_0|++\!+\rangle + \alpha_1|--\!-\rangle. \tag{3.36}$$

The state of physical qubits given in Eq. (3.36) can be stabilized by the operators $S_1 = \mathbf{XXI}$ and $S_2 = \mathbf{XIX}$. The detection and correction of a phase flip error can be carried out in analogy with the 1/3-rate quantum repetition code for handling the bit-flip error. The stabilizer operators can be derived from the classical PCM $\mathbf{H}$ by mapping the Pauli matrices $\mathbf{I}$, $\mathbf{X}$, $\mathbf{Y}$ and $\mathbf{Z}$ onto $(\mathbb{F}_2)^2$ using one of the Pauli mappings given in Eq. 3.18. Each row of $\mathbf{H}$ is associated with a stabilizer operator $S_i \in \mathcal{S}$, where the i-th column of both $\mathbf{H}_z$ and $\mathbf{H}_x$ corresponds to the i-th

qubit and the binary 1 locations represent the **Z** and **X** positions in the PCMs $\mathbf{H}_z$ and $\mathbf{H}_x$, respectively. For instance, for the 1/3-rate quantum repetition code, which is stabilized by the operators $S_1 = \mathbf{ZZI}$ and $S_2 = \mathbf{ZIZ}$, the PCM $\mathbf{H}$ is given as follows:

$$\mathbf{H} = \left(\begin{array}{ccc|ccc} 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 \end{array}\right). \tag{3.37}$$

Since the 1/3-rate quantum repetition code in this example can only correct a bit-flip (**X**) error, which is stabilized by the **Z** operators, the PCM $\mathbf{H}_x$ contains only zero elements. The same goes for a 1/3-rate quantum repetition code conceived for handling a phase-flip (**Z**) error, which is stabilized by the operators $S_1 = \mathbf{XXI}$ and $S_2 = \mathbf{XIX}$. The PCM $\mathbf{H}$ corresponding to this particular QSC is defined as follows:

$$\mathbf{H} = \left(\begin{array}{ccc|ccc} 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 \end{array}\right). \tag{3.38}$$

It is clearly shown in Eq. (3.37) and (3.38) that the PCM of a 1/3-rate quantum repetition code is similar to that of the 1/3-rate classical repetition code given in Eq. (3.25).

In order to encode the logical qubits into physical qubits, we require the unitary transformation $\mathcal{V}$ acting as the quantum encoder. To represent the quantum encoding circuit, one of the essential components is the controlled-NOT (CNOT) quantum gate, which has been described in Subsection 2.3.1.4. A logical qubit in the superimposed state of $|\psi\rangle = \alpha_0|0\rangle + \alpha_1|1\rangle$ and a qubit in the pure state of $|0\rangle$ are manipulated by the quantum CNOT gate into following state:

$$\begin{aligned} \mathbf{CNOT}(|\psi\rangle, |0\rangle) &= \mathbf{CNOT}(\alpha_0|0\rangle + \alpha_1|1\rangle, |0\rangle) \\ &= \mathbf{CNOT}(\alpha_0|00\rangle + \alpha_1|10\rangle) \\ &= \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \cdot \begin{pmatrix} \alpha_{00} \\ 0 \\ \alpha_{10} \\ 0 \end{pmatrix} = \begin{pmatrix} \alpha_0 \\ 0 \\ 0 \\ \alpha_1 \end{pmatrix} \\ &\equiv \alpha_0|00\rangle + \alpha_1|11\rangle. \end{aligned} \tag{3.39}$$

Similarly, we may also use the CNOT definition given in Eq. (2.33) to determine the resultant state as described below:

$$\begin{aligned} \mathbf{CNOT}(|\psi\rangle, |0\rangle) &= \mathbf{CNOT}(\alpha_0|0\rangle + \alpha_1|1\rangle, |0\rangle) \\ &= \mathbf{CNOT}(\alpha_0|00\rangle + \alpha_1|10\rangle) \\ &= \alpha_0|0, (0 \oplus 0)\rangle + \alpha_1|1, (1 \oplus 0)\rangle \\ &= \alpha_0|00\rangle + \alpha_1|11\rangle. \end{aligned} \tag{3.40}$$

For the sake of creating the encoded state of a 1/3-rate quantum repetition code, we require a single logical qubit and two ancillas prepared in the pure state of $|0\rangle$, as described in Eq. (3.28). In the first step, the CNOT unitary transformation is performed between the logical qubit and the first ancilla, in which the logical qubit acts as the control qubit and the ancilla as the target qubit. The same step is repeated during the second stage between the logical qubit and the second ancilla, where the second ancilla is also preserved as the target qubit. Therefore, the encoding circuit of the 1/3-rate quantum repetition code can be represented as in Fig 3.4, which was designed for protecting the physical qubits from a single bit-flip error, as also seen in the mapping given in Eq. (3.26).

For its 1/3-rate quantum repetition code counterpart protecting the physical qubits from a phase-flip error, we require the Hadamard transformation to obtain the mapping given in Eq. (3.35). The Hadamard transformation is required for changing the base of quantum repetition codes from the computational basis $\{|0\rangle, |1\rangle\}$ to the Hadamard basis $\{|+\rangle, |-\rangle\}$. Hence, we can readily create the encoding circuit for a 1/3-rate quantum repetition code for protecting the physical qubits from a phase-flip error by placing the Hadamard gates after the second stage as portrayed in Fig. 3.5.

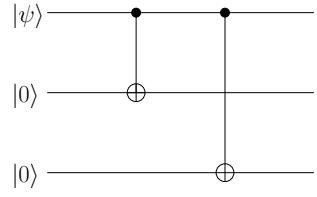


Figure 3.4: The encoding circuit of the 1/3-rate quantum repetition code protecting the physical qubits from a bit-flip error.

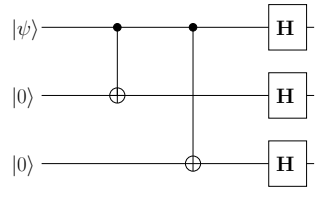


Figure 3.5: The encoding circuit of the 1/3-rate quantum repetition code protecting the physical qubits from a phase-flip error.

3.4.2 Shor's 9-Qubit Code

Since we have elaborated briefly on the construction of QSCs along with the Pauli to binary isomorphism, we may now proceed with the corresponding examples of different QSC constructions conceived for protecting the physical qubits from any type of a single qubit error. Firstly, we commence with Shor's code [58]. In order to protect the qubits from any type of single qubit error, a logical qubit is mapped onto nine physical qubits. This code may also be viewed as a concatenated version of two 1/3-rate quantum repetition codes, where the first stage is dedicated

to the protection of the physical qubits from phase-flip errors, while the second stage is invoked for handling the bit-flip errors. To elaborate further, at the first stage of Shor's code, the state of a logical qubit is encoded by using the following mapping: $|0\rangle \rightarrow |+++\rangle$, $|1\rangle \rightarrow |---\rangle$. At the second stage, we encode each of the states of $|+\rangle$ to the state of $(|000\rangle + |111\rangle)/\sqrt{2}$, while the state of $|-\rangle$ is mapped to the state of $(|000\rangle - |111\rangle)/\sqrt{2}$. Therefore, the final state of the encoded logical qubits $|0\rangle_L$ and $|1\rangle_L$ as follows:

$$\begin{aligned}
|0\rangle_L &= \frac{1}{\sqrt{2}}(|000\rangle + |111\rangle) \otimes \frac{1}{\sqrt{2}}(|000\rangle + |111\rangle) \otimes \frac{1}{\sqrt{2}}(|000\rangle + |111\rangle) \\
&= \frac{1}{2\sqrt{2}}(|000000000\rangle + |000000111\rangle + |000111000\rangle + |000111111\rangle \\
&+ |111000000\rangle + |111000111\rangle + |111111000\rangle + |111111111\rangle), \qquad (3.41) \\
|1\rangle_L &= \frac{1}{\sqrt{2}}(|000\rangle - |111\rangle) \otimes \frac{1}{\sqrt{2}}(|000\rangle - |111\rangle) \otimes \frac{1}{\sqrt{2}}(|000\rangle - |111\rangle) \\
&= \frac{1}{2\sqrt{2}}(|000000000\rangle - |000000111\rangle - |000111000\rangle + |000111111\rangle \\
&- |111000000\rangle + |111000111\rangle + |111111000\rangle - |111111111\rangle). \qquad (3.42)
\end{aligned}$$

Based on the above description, the encoding circuit of Shor's code is portrayed in Fig. 3.6. The state determined by the nine physical qubits of Shor's code, where the latter are defined in Eq. (3.41) and (3.42), is stabilized by the eight stabilizer operators which are listed in Table 3.5.

Table 3.5: The eight stabilizer operators of Shor's 9-qubit code, which stabilizes a single logical qubit with the aid of eight auxiliarry qubits.

S_i	Stabilizer Operator
S_1	**ZZIIIIIII**
S_2	**IZZIIIIII**
S_3	**IIIZZIIII**
S_4	**IIIIZZIII**
S_5	**IIIIIIZZI**
S_6	**IIIIIIIZZ**
S_7	**XXXXXXIII**
S_8	**IIIXXXXXX**

To elaborate a little further, Shor's code is a member of the class of non-dual-containing CSS codes. Explicitly, it belongs to the class of CSS codes because the stabilizer formalism of Shor's code implies that the code handles the **Z** error and the **X** error separately, whilst it is a non-dual-containing code because the PCMs $\mathbf{H}_z$ and $\mathbf{H}_x$ are not identical. Based on the list of stabilizer operators given in Table 3.5, the PCM **H** of Shor's code is given in Eq. (3.43), where each row of the PCM corresponds to each of the stabilizer operators listed in Table 3.5.

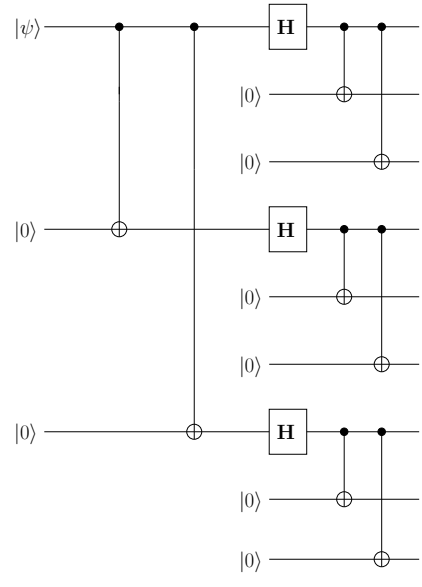


Figure 3.6: The encoding circuit $\mathcal{V}$ of Shor's 9-qubit code.

$$\mathbf{H}_{Shor} = \left(\begin{array}{ccccccccc|ccccccccc} 1&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\ 0&1&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\ 0&0&0&1&1&0&0&0&0&0&0&0&0&0&0&0&0&0\\ 0&0&0&0&1&1&0&0&0&0&0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&1&1&0&0&0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&1&1&0&0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0&0&1&1&1&1&1&1&0&0&0\\ 0&0&0&0&0&0&0&0&0&0&0&0&1&1&1&1&1&1 \end{array}\right). \tag{3.43}$$

The quantum coding rate (r_Q) of a quantum code $\mathcal{C}[n,k]$ is defined by the ratio of the number of logical qubits k to the number of physical qubits n, which can be formulated as

$$r_Q = \frac{k}{n}. \tag{3.44}$$

Hence again, for Shor's 9-qubit code the quantum coding rate is $r_Q = 1/9$.

3.4.3 Steane's 7-Qubit Code

Steane's code was proposed for protecting a single qubit from any type of error by mapping a logical qubit onto seven physical qubits, instead of nine qubits. In contrast to Shor's code, Steane's code is a dual-containing CSS code, since the PCMs $\mathbf{H}_z$ and $\mathbf{H}_x$ are based on that of

the clasical Hamming code $\mathbf{H}_{Ham}$, which is repeated here for convenience:

$$\mathbf{H}_{Ham} = \begin{pmatrix} 1 & 1 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 1 \end{pmatrix}. \tag{3.45}$$

It can be confirmed that the classical Hamming code is a dual-containing code, because it satisfies the condition $\mathbf{H}_{Ham}.\mathbf{H}_{Ham}^T = 0$. Therefore, the PCM $\mathbf{H}$ of Steane's code is defined as:

$$\mathbf{H}_{Steane} = \left(\begin{array}{c|c} \mathbf{H}_{Ham} & \mathbf{0} \\ \mathbf{0} & \mathbf{H}_{Ham} \end{array}\right) = \left(\begin{array}{ccccccc|ccccccc} 1 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 1 \end{array}\right). \tag{3.46}$$

Since Steane's code is a member of the dual-containing CSS codes, the encoded state of the logical qubit $|0\rangle_L$ and $|1\rangle_L$ may be determined from its classical code counterpart. Let $\mathcal{C}_1(7,4)$ be the Hamming code and $\mathcal{C}_2(7,3)$ be its dual. Both of the codes are capable of corrrecting a single bit error. Hence, the resultant CSS quantum code derived from these codes, namely the $\mathcal{C}[n, k_1 - k_2] = \mathcal{C}[7,1]$, is also capable of correcting a single qubit error. For Steane's code the states of the encoded logical qubit of $|0\rangle_L$ and $|1\rangle_L$ are defined as follows:

$$|0\rangle_L = \frac{1}{\sqrt{|\mathcal{C}_2|}} \sum_{x \in \mathcal{C}_1, \mathcal{C}_2} |x\rangle, \tag{3.47}$$

$$|1\rangle_L = \frac{1}{\sqrt{|\mathcal{C}_2|}} \sum_{x \in \mathcal{C}_1, x \notin \mathcal{C}_2} |x\rangle. \tag{3.48}$$

Since $\mathcal{C}_2$ is the dual of $\mathcal{C}_1$, the PCM of $\mathcal{C}_2$, denoted by $\mathbf{H}(\mathcal{C}_2)$ is by definition the generator matrix of $\mathcal{C}_1$, which is denoted by $\mathbf{G}(\mathcal{C}_1)$. Hence, the parity-check matrix of $\mathcal{C}_2$ can be written as

$$\mathbf{H}(\mathcal{C}_2) = \mathbf{G}(\mathcal{C}_1) = \begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{pmatrix}. \tag{3.49}$$

Based on the PCM given in Eq. (3.45) and (3.49), we can define the code space of $\mathcal{C}_1$ and $\mathcal{C}_2$, which is described in Table 3.6. Finally, upon using Eq. (3.47), (3.48), and also the code space given in Table 3.6, the encoded states of the logical qubit $|0\rangle_L$ and $|1\rangle_L$ of Steane's code

Table 3.6: The code space of $\mathcal{C}_1$ and $\mathcal{C}_2$ for determining the encoded state of Steane's code.

$x \in \mathcal{C}_1, \mathcal{C}_2$	$x \in \mathcal{C}_1, x \notin \mathcal{C}_2$
0000000	1111111
0111001	1000110
1011010	0100101
1100011	0011100
1101100	0010011
1010101	0101010
0110110	1001001
0001111	1110000

may be formulated as:

$$|0\rangle_L = \frac{1}{2\sqrt{2}}(|0000000\rangle + |0111001\rangle + |1011010\rangle + |1100011\rangle + |1101100\rangle + |1010101\rangle + |0110110\rangle + |0001111\rangle), \tag{3.50}$$

$$|1\rangle_L = \frac{1}{2\sqrt{2}}(|1111111\rangle + |1000110\rangle + |0100101\rangle + |0011100\rangle + |0010011\rangle + |0101010\rangle + |1001001\rangle + |1110000\rangle). \tag{3.51}$$

Again, it can be readily seen that the quantum coding rate of Steane's 7-qubit code is 1/7. Its encoding circuit is portrayed in Fig. 3.7, while the corresponding set of stabilizers is listed in Table 3.7. Observe in the table that in the positions of the logical ones seen in Eq. 3.45 we have either Z or X, while the logical zeros correspond to I. Furthermore, the first three rows in the table are mirrored by the bottom three rows upon replacing Z by x. Steane's code is clearly more efficient than Shor's code, because it requires seven - rather than nine - quibits for protecting a single one. However, the separate treatment of bit-flips and phase-flips is still associated with some extra redundancy, as it becomes explicit by listing the number of possible erroneous scenarios. Explicitly, each of the seven qubits might be corrupted in three possible ways by X, Z and Y errors, which results in 21 possibilites, plus we have an error-free scenario.Therefore we would require five stabilizers, which would allow us to differentiate up to 32 situations. But given the six stabilizers of Table 3.6 we could potentially distinguish as many as 64 scenarios. Therefore we surmize that there exists a more efficient code. Following the same logic, let us consider a five-qubit code, where again, each qubit may be corrupted in three different ways, yielding a total of 15 possible cases, plus an error-free scenario. These 16 cases would require four stabilizers for uniquely and unambiguously distinguishing and eventually correcting all single-qubit errors. This is why Laflamme's 1/5-rate code has the fond connotation of being a 'perfect' code, which will be discussed in the next section.

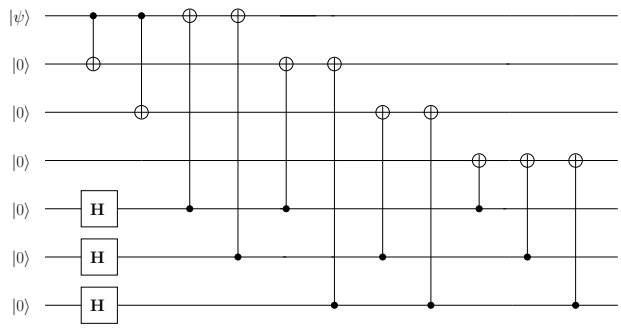


Figure 3.7: The encoding circuit $\mathcal{V}$ of Steane's 7-qubit code.

Table 3.7: The stabilizer operators of Steane's 7-qubit code.

S_i	Stabilizer Operator
S_1	**ZZIZZII**
S_2	**ZIZZIZI**
S_3	**IZZZIIZ**
S_4	**XXIXXII**
S_5	**XIXXIXI**
S_6	**IXXXIIX**

3.4.4 Laflamme's 5-Qubit Code - The Perfect Code

Again, Laflamme's code maps a single logical qubit onto five physical qubits, which is also referred to as the "perfect code", because as alluded to in the previous sections, in order to protect a logical qubit, the lowest number of physical qubits required is five [61, 62]. This 5-qubit scheme is a non-CSS code, since the stabilizer formalism is designed to handle the **Z** errors and **X** errors simultaneously. There are several existing designs related to the perfect 5-qubit code [60, 181] and in this treatise, we use the PCM formulation given in [181]. Explicitly, its non-CSS characteristics can be readily observed from the PCM $\mathbf{H}_{perfect}$, which is specified as follows:

$$\mathbf{H}_{perfect} = \left(\begin{array}{ccccc|ccccc} 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \end{array}\right). \tag{3.52}$$

Hence, the stabilizer operators of the 5-qubit code may be explicitly formulated as in Table 3.8.

In general, the encoded state of a QSC which encodes a single-logical qubit into n physical

Table 3.8: The stabilizer formalism of the perfect 5-qubit code.

S_i	Stabilizer Operator
S_1	**XZZXI**
S_2	**IXZZX**
S_3	**XIXZZ**
S_4	**ZXIXZ**

qubits can defined as follows [188]:

$$|0\rangle_L = \prod_{S_i \in \mathcal{S}} \frac{1}{\sqrt{N}} \left(\mathbf{I}^{\otimes n} + S_i\right) |0\rangle^{\otimes N}, \tag{3.53}$$

$$|1\rangle_L = \prod_{S_i \in \mathcal{S}} \frac{1}{\sqrt{N}} \left(\mathbf{I}^{\otimes n} + S_i\right) |1\rangle^{\otimes N}, \tag{3.54}$$

where the factor N is introduced to preserve the unitary constraint. Therefore, based on the stabilizer operators as well as on Eq. (3.53) and (3.54), the encoded state of Laflamme's 5-qubit code can be defined as follows:

$$\begin{aligned} |0\rangle_L = \frac{1}{\sqrt{N}} &\left(\mathbf{I}^{\otimes 5} + \mathbf{XZZXI}\right)\left(\mathbf{I}^{\otimes 5} + \mathbf{XZZXI}\right)\left(\mathbf{I}^{\otimes 5} + \mathbf{XZZXI}\right) \\ &\left(\mathbf{I}^{\otimes 5} + \mathbf{XZZXI}\right)\left(\mathbf{I}^{\otimes 5} + \mathbf{XZZXI}\right)|0\rangle^{\otimes 5}, \end{aligned} \tag{3.55}$$

$$\begin{aligned} |1\rangle_L = \frac{1}{\sqrt{N}} &\left(\mathbf{I}^{\otimes 5} + \mathbf{XZZXI}\right)\left(\mathbf{I}^{\otimes 5} + \mathbf{XZZXI}\right)\left(\mathbf{I}^{\otimes 5} + \mathbf{XZZXI}\right) \\ &\left(\mathbf{I}^{\otimes 5} + \mathbf{XZZXI}\right)\left(\mathbf{I}^{\otimes 5} + \mathbf{XZZXI}\right)|1\rangle^{\otimes 5}. \end{aligned} \tag{3.56}$$

Finally, after a few further steps we arrive at:

$$\begin{aligned} |0\rangle_L = \frac{1}{4}(&|00000\rangle - |00011\rangle + |00101\rangle - |00110\rangle + |01001\rangle + |01010\rangle - |01100\rangle - |01111\rangle \\ &- |10001\rangle + |10010\rangle + |10100\rangle - |10111\rangle - |11000\rangle - |11011\rangle - |11101\rangle - |11110\rangle), \end{aligned} \tag{3.57}$$

$$\begin{aligned} |1\rangle_L = \frac{1}{4}(&|11111\rangle - |11100\rangle + |11010\rangle - |11001\rangle + |10110\rangle + |10101\rangle - |10011\rangle - |10000\rangle \\ &- |01110\rangle + |01101\rangle + |01011\rangle - |01000\rangle - |00111\rangle - |00100\rangle - |00010\rangle - |00001\rangle). \end{aligned} \tag{3.58}$$

The same method can be utilized for determining the encoded state of a logical qubit for Shor's code and Steane's code. However, both Shor's code and Steane's code offer a more simplistic approach for determining their corresponding encoded states. The design of an efficient encoding circuit $\mathcal{V}$ conceived for the 1/5-rate code of Laflamme can be found in [60, 205, 206].

Based on the aforementioned constructions, we evaluated the performance of the above three QSCs by simulation in the context of quantum depolarizing channels. The performance of the 9-qubit Shor-code, 7-qubit Steane-code and 5-qubit Laflamme-code is portrayed in Fig. 3.8 in terms of the qubit error rate (QBER) *vs.* the depolarizing probability (p). Observed that the performances of all three QSCs are quite similar, which is not unexpected, because they are all

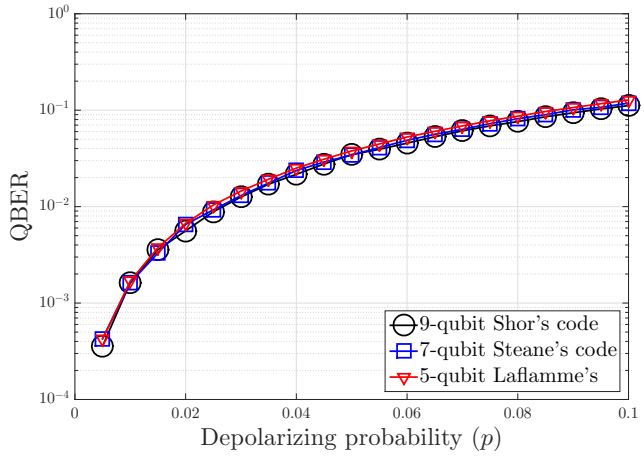


Figure 3.8: QBER performance of the QSCs protecting a single qubit, namely Shor's 9-qubit code, Steane's 7-qubit code and the perfect 5-qubit code, recorded for the quantum depolarizing channel. The similarity of performances is because all of the QSCs rely on hard-decision syndrome decoding and they all have the same error correction capabilities.

capable of correcting a single-qubit error.

3.5 Summary and Conclusions

The purpose of this chapter is to provide the similarities between the classical and quantum error correction codes. More specifically, the QSCs can be viewed as the syndrome-based version of QECCs, as exemplified by the 1/3-rate quantum repetition code. By using the classical-to-quantum Pauli isomorphism, we can observe that a QSC can be treated as a syndrome-based classical error-correction code, whose PCM can be separated into two independent PCMs, one for dealing with bit-flip ($\mathbf{X}$) errors, which is the PCM $\mathbf{H}_x$, and one for handling the phase-flip ($\mathbf{Z}$) errors, which is the PCM $\mathbf{H}_z$. Based on how to mitigate the $\mathbf{X}$ and the $\mathbf{Z}$ errors, a QSC can be classified either as a CSS code or a non-CSS code. The main difference between the two is the CSS codes handle the $\mathbf{X}$ and the $\mathbf{Z}$ errors independently, as exemplified by the PCMs of Shor's code and Steane's code, while the non-CSS codes treat them simultaneously, as exemplified by the PCM of the Laflamme's code.

We have simulated the QBER performance for Shor's, Steane's and Laflamme's code over the quantum depolarizing channel. The simulation results show a similar QBER performance for all of the aforementioned codes. The main reason is all of Shor's, Steane's and Laflamme's code exhibit the same minimum distance ($d = 3$), which translates directly to the error correction capability of the code ($t = 1$) despite of their differences in code type, codeword length and and quantum coding rate. Therefore, we believe that the trade-off between conflicting parameters in QSCs such as minimum distance, quantum coding rate and codeword length is a pivotal subject

to investigate. This specific line of investigation will be carried out in Chapter 6.

Chapter 4

Revisiting Classical Syndrome Decoding

4.1 Introduction

In Chapters 3 we kept the level of treatment relatively simple for readers who are exposed to the quantum coding concepts for the first time, including the syndrom decoding basics relying on the $\mathcal{C}(7,4)$ Hamming code. A rudimentary introduction to quantum stabilizer codes was also provided.

In Chapters 7 and **??**, we then offered somewhat deeper 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 **??** may be expanded, as shown in Figure 4.1. The process begins with the computation of the syndrome of the received sequence $\left|\hat{\psi}\right\rangle$ using the stabilizer generators, which collapse to a binary 0 or 1 upon measurement. For estimating the equivalent channel error $\tilde{P}$ (or $\tilde{\hat{P}}$ 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. (**??**) (or quaternary-to-Pauli mapping of Eq. (**??**)). The classical syndrome decoder component used in the middle of Figure 4.1 bears some similarity to that of a conventional classical code, but it also exhibits some differences:

(a) In contrast to the syndrome of a classical code, which is computed as the product of the

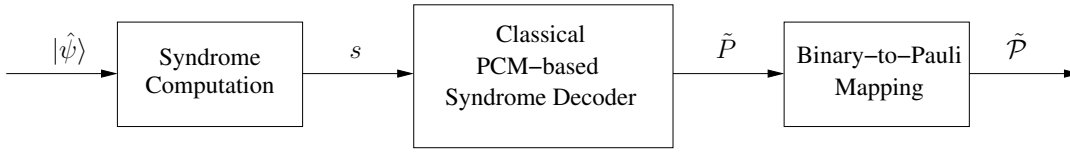


Figure 4.1: Syndrome processing block seen in Figure **??**.

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

(b) The conventional classical decoding aims for finding the most likely error vector, given the syndrome observed, 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 **??**.

In this chapter, we will focus our attention on the classical syndrome decoding techniques conceived for conventional classical codes transmitted over a classical channel, thereby ignoring the above-mentioned pair of differences. The concepts developed in this chapter together with those of Chapter **??** 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.

Again, 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 errors by harnessing a pre-computed LUT [207]. 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.* [208] 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 [209], whereby the method of constructing the syndrome-based code trellis for linear block codes was presented. Wolf [209] also proved that in contrast to a brute force ML decoding of a linear block code $C(n, k)$ over $\mathrm{GF}(q)$, which would require q^k evaluations, a code trellis requires only $q^{\min\{k,n-k\}}$ states. The ideas presented in [208, 209] 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 [210, 211, 212] conceived the idea of a

[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.

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 [213] and [214], 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 [215].

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 [216, 217] 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 [218]. The concept of BSD was further extended to turbo codes in [219], 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 [220] for designing an adaptive low-complexity decoding approach for turbo equalization. Some other applications of BSD are dealt with in [221, 222, 223].

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 [220] together with the BSD of [219] to Turbo Trellis Coded Modulation (TTCM) for the sake of reducing its decoding complexity [224]. 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 4.2, 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 4.3. More particularly, Section 4.3.1 focuses on the construction of the syndrome-based trellis of linear block codes, while in Section 4.3.2 we extend this syndrome-based trellis formalism to convolutional codes. Section 4.4 deals with the BSD. More specifically, in Section 4.4.1, we lay out the general BSD formalism, while our proposed block-based syndrome decoder designed for TTCM is presented in Section 4.4.2. We then evaluate the performance of our proposed decoder in Section 4.5. Finally, we summarize the chapter in Section 4.6.

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

4.2 Look-Up Table-Based Syndrome Decoding

The PCM-based syndrome decoding technique discussed in Section **??** is in fact the LUT-based syndrome-based decoder, which is based on Table **??**. In this section, we will detail the construction of the LUT.

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:

- 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 for ensuring 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 formulated as:

$$\begin{pmatrix} \overline{x}_1 = 0 & \overline{x}_2 & \ldots & \overline{x}_i & \ldots & \overline{x}_{2^k} \\ e_2 & e_2 + \overline{x}_2 & \ldots & e_2 + \overline{x}_i & \ldots & e_2 + \overline{x}_{2^k} \\ e_3 & e_3 + \overline{x}_2 & \ldots & e_3 + \overline{x}_i & \ldots & e_3 + \overline{x}_{2^k} \\ \vdots & & & & & \vdots \\ e_{2^{n-k}} & e_{2^{n-k}} + \overline{x}_2 & \ldots & e_{2^{n-k}} + \overline{x}_i & \ldots & e_{2^{n-k}} + \overline{x}_{2^k} \end{pmatrix}, \tag{4.1}$$

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. (4.1) is a unique coset, whose coset leader is the minimum weight vector given in the first column. We may notice in Eq. (4.1) 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 **??** LUT-based syndrome decoder, let us construct the associated standard array. Recall from Section **??** 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. (4.1) and this set of valid codewords, we get the following standard array:

$$\begin{pmatrix} \overline{x}_1 & \overline{x}_2 \\ e_2 & e_2 + \overline{x}_2 \\ e_3 & e_3 + \overline{x}_2 \\ e_4 & e_4 + \overline{x}_2 \end{pmatrix} = \begin{pmatrix} 000 & 111 \\ 001 & 110 \\ 010 & 101 \\ 100 & 011 \end{pmatrix}. \tag{4.2}$$

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 in Eq. (4.1). Its syndrome may be formulated as:

$$s = (e_j + \overline{x}_i)H^T = e_j H^T + \overline{x}_i H^T = e_j H^T. \tag{4.3}$$

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:

$$\begin{pmatrix} e_1 = 0 & 0 \\ e_2 & e_2 H^T \\ e_3 & e_3 H^T \\ e_4 & e_4 H^T \end{pmatrix} = \begin{pmatrix} 000 & 00 \\ 001 & 01 \\ 010 & 10 \\ 100 & 11 \end{pmatrix}. \tag{4.4}$$

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

4.3 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. (4.1)) corresponding to that syndrome. When the syndrome is zero, which is equivalent to the first coset of Eq. (4.1), 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 [225] or the Bahl-Cocke-Jelinek-Raviv (BCJR) decoding (also called MAP decoding) [208] 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

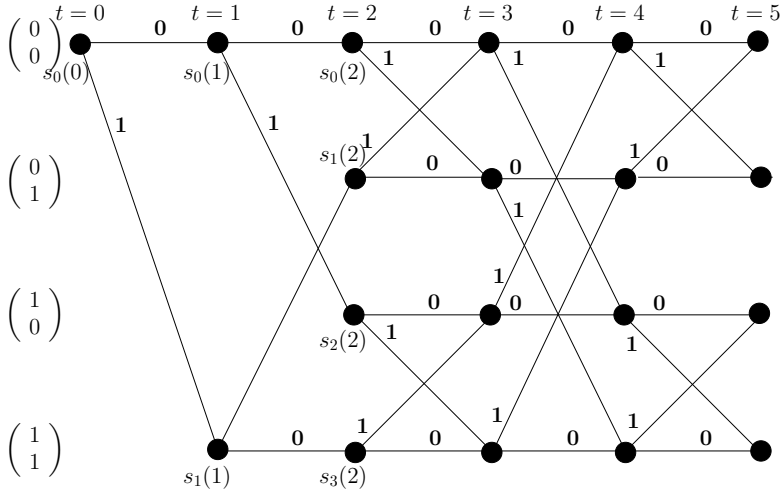


Figure 4.2: Syndrome-based trellis for a linear block code $C(5,3)$ constructed over GF(2).

block codes and convolutional codes in Section 4.3.1 and 4.3.2, respectively. Finally, the MAP algorithm will be presented later in Section 4.4.2.2.

4.3.1 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 4.2 for a binary code $C(5,3)$, whose PCM is given by,

$$H=\begin{pmatrix}1&1&0&1&0\\1&0&1&0&1\end{pmatrix}=\begin{pmatrix}h_1&h_2&h_3&h_4&h_5\end{pmatrix}. \tag{4.5}$$

The trellis of Figure 4.2 can be constructed as follows:

(a) The trellis emerges from the trellis-state $(0,0)$ at $t=0$, i.e. we have $s_0(0)=(0,0)$.

(b) 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:

$$s_l(t)=s_i(t-1)+\alpha_j h_t,\quad \text{for } j=0,1,\ldots,(q-1), \tag{4.6}$$

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. (4.6) i.e. $\alpha_0 = 0$ and $\alpha_1 = 1$. Substituting these values of α_j into Eq. (4.6) at $t = 1$ yields:

$$s_0(1) = s_0(0) + \alpha_0 h_1 = \begin{pmatrix} 0 \\ \\ 0 \end{pmatrix} + 0. \begin{pmatrix} 1 \\ \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ \\ 0 \end{pmatrix}, \tag{4.7}$$

$$s_1(1) = s_0(0) + \alpha_1 h_1 = \begin{pmatrix} 0 \\ \\ 0 \end{pmatrix} + 1. \begin{pmatrix} 1 \\ \\ 1 \end{pmatrix} = \begin{pmatrix} 1 \\ \\ 1 \end{pmatrix}. \tag{4.8}$$

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 4.6. The corresponding edges are labeled by 0 and 1 for $\alpha_0 = 0$ and $\alpha_1 = 1$, respectively.

Eq. (4.7) and Eq. (4.8) 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,

$$s = yH^T. \tag{4.9}$$

At $t = 1$, we receive only the first element of y i.e. $y = [y_1 0000]^T$. Therefore, Eq. (4.7) and Eq. (4.8) can also be formulated as:

$$s_l(1) = \begin{pmatrix} 1 & 1 & 0 & 1 & 0 \\ \\ 1 & 0 & 1 & 0 & 1 \end{pmatrix} . \begin{pmatrix} y_1 \\ \\ 0 \\ \\ 0 \\ \\ 0 \\ \\ 0 \end{pmatrix} \quad \text{where } y_1 \in \{\alpha_0, \alpha_1\}. \tag{4.10}$$

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

$$\begin{aligned} s_l(2) &= s_0(1) + \alpha_j h_2 \\ s_l(2) &= s_1(1) + \alpha_j h_2, \end{aligned} \tag{4.11}$$

for $j = \{0, 1\}$. If the alternate approach of Eq. (4.10) is adopted, Eq. (4.11) may be

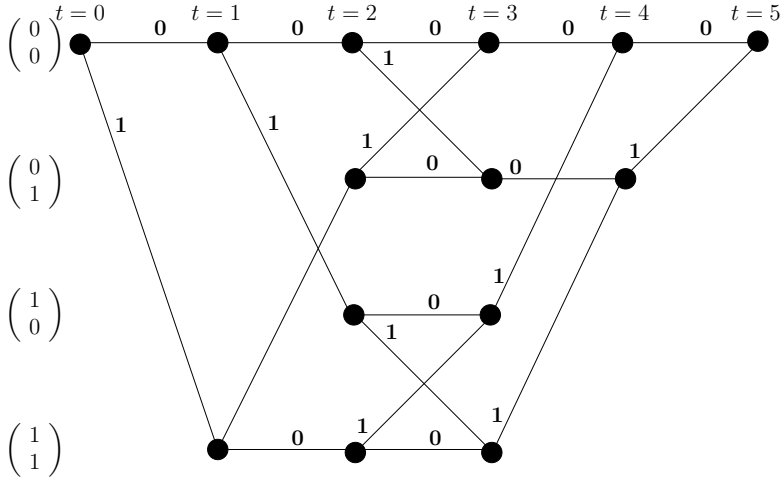


Figure 4.3: Expurgated syndrome-based code trellis for the binary code $C(5, 3)$.

written as:

$$s_l(2) = \begin{pmatrix} 1 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} y_1 \\ y_2 \\ 0 \\ 0 \\ 0 \end{pmatrix}, \quad \text{where } y_1, y_2 \in \{\alpha_0, \alpha_1\}, \tag{4.12}$$

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 $t = 5$.

(c) 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 4.3. 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:

(a) 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 $(n - k)$.

(b) 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 Eq. (4.6). 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.

(c) 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.

4.3.2 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 [226] for illustrating the trellis construction procedure. We use a rate-2/3 convolutional code, whose generator $G(D)$ is given by:

$$G(D) = \begin{pmatrix} 1+D & D & 1+D \\ D & 1 & 1 \end{pmatrix}. \tag{4.13}$$

The corresponding syndrome former is as follows:

$$H^T(D) = \begin{pmatrix} 1 \\ 1+D^2 \\ 1+D+D^2 \end{pmatrix}. \tag{4.14}$$

The circuit of $G(D)$ for CC(n,k) is realized in Figure 4.4(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:

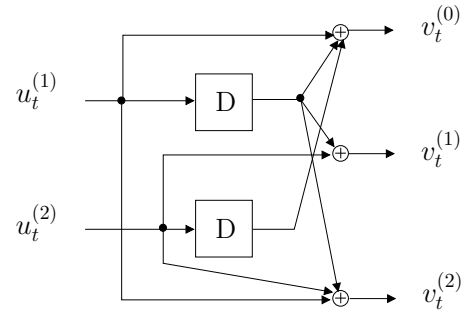
$$y_t^j = v_t^j \oplus e_t^j. \tag{4.15}$$

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

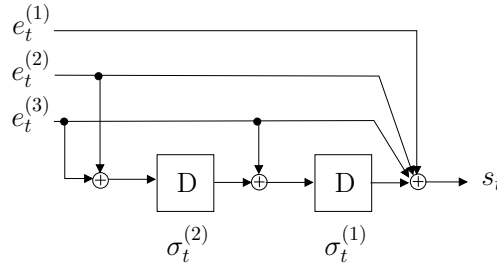
$$s = yH^T = (v+e)H^T = eH^T. \tag{4.16}$$

This realization of the syndrome former is shown in Figure 4.4(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. (4.14). Consequently, we may construct the trellis for

[4]Recall from Section **??** 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. (4.14) is defined later in Eq. (4.20).



(a) Encoder G



(b) Syndrome Former H^T

Figure 4.4: Realization of the encoder and syndrome former given in Eq. (4.13) and Eq. (4.14), respectively.

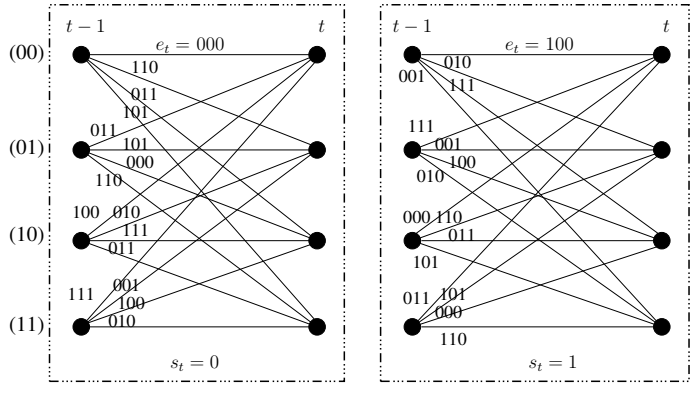


Figure 4.5: Syndrome-based error sub-trellises for H^T corresponding to $s_t = 0$ and $s_t = 1$.

the syndrome former of Figure 4.4(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 4.5. 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:

$$\{y_t\}_{t=1}^{3} = \{011 \;\; 011 \;\; 111\}. \tag{4.17}$$

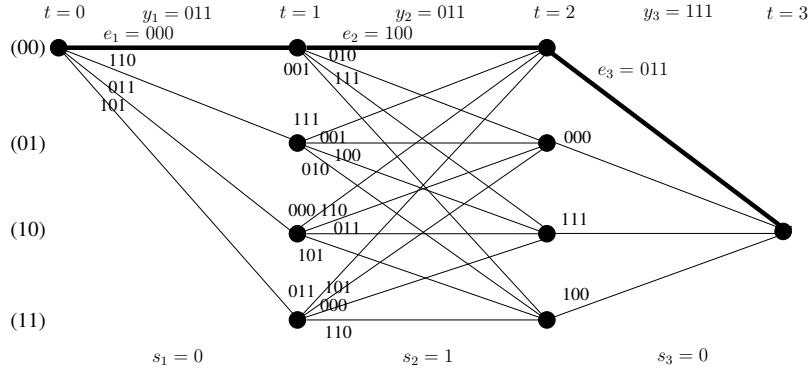


Figure 4.6: Syndrome-based error trellis for H^T corresponding to the received sequence of Eq. (4.17), which is constructed by concatenating the sub-trellises of Figure 4.5.

Since the received sequence y_t and the inflicted channel error e_t yield the same syndrome according to Eq. (4.16), we feed y_t into the circuit of Figure 4.4(b) for computing the syndrome. Consequently, using the syndrome former of Figure 4.4(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 4.5 based on the value of s_t. More specifically, for the syndrome values of $\{s_t\}_{t=1}^{3} = \{0, 1, 0\}$, 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 4.6 for the received sequence of Eq. (4.17), which is initialized to the state $\sigma_0 = (00)$ and terminates at $\sigma_3 = (10)$.

The error trellis of Figure 4.6 is equivalent to the conventional code trellis generated using the generator G, which is shown in Figure 4.7. 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 4.6 corresponds to a path of Figure 4.7 and this correspondence is a one-to-one relationship [226]. Consider an arbitrary path $\tilde{e} = \{000 \;\; 100 \;\; 011\}$ of Figure 4.6 (marked with a thick line). Since the received sequence is $y = \{011 \;\; 011 \;\; 111\}$, the estimated transmitted code sequence $\tilde{v}$ is computed as:

$$\tilde{v}_t^{(j)} = \tilde{e}_t^{(j)} \oplus y_t^{(j)} \; . \tag{4.18}$$

Hence, we have $\tilde{v} = \{011 \;\; 111 \;\; 100\}$, which is a path of the trellis seen in Figure 4.7 (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 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. (4.18), yielding the most likely transmitted code sequence $\tilde{v}$. It is then passed through

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

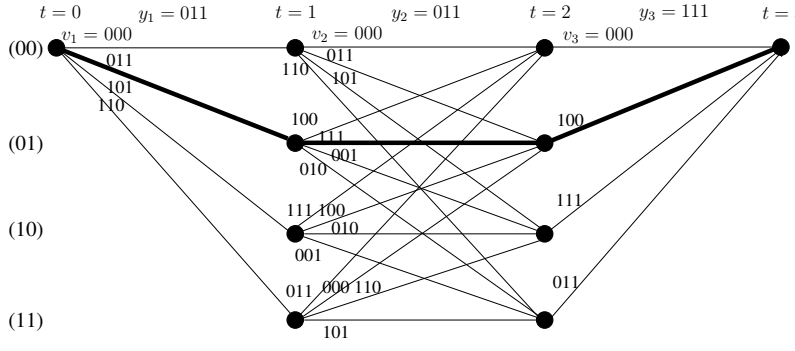


Figure 4.7: Conventional code trellis, generated using the encoder G of Figure 4.4(a), corresponding to Figure 4.6. Each path of Figure 4.6 can be mapped onto a path of the code trellis using Eq. (4.18).

the inverse encoder G^{-1} for estimating the most likely information sequence $\tilde{u}$ as follows:

$$\tilde{u} = \tilde{v}G^{-1}. \tag{4.19}$$

The syndrome-based error trellis of Figure 4.6 has the same complexity as the conventional code trellis of Figure 4.7 for soft-decision decoding, which is $\sim nq^k q^m$ for CC(n, k, m) and grows exponentially upon increasing the value of k (increasing the coding rate R). By contrast, Sidorenko *et al.* [214] 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 [209] conceived for linear block codes, which was discussed in Section 4.3.1. 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 [214] of constructing the syndrome-based trellis divides each branch of Figure 4.6 into n stages.

Let us now continue the same example as in Eq. (4.14). Recall from Section **??** that a convolutional code is equivalent to a semi-infinite linear block code. For the syndrome former of Eq. (4.14), the associated semi-infinite parity check matrix can be constructed as:

$$H = \begin{pmatrix} 111 & & & \\ 001 & 111 & & \\ 011 & 001 & 111 & \\ & 011 & 001 & 111 \\ & \ddots & & \ddots \end{pmatrix}, \tag{4.20}$$

[6]The complexity of a code trellis is the number of operations (selections and additions) required to decode a block [214].

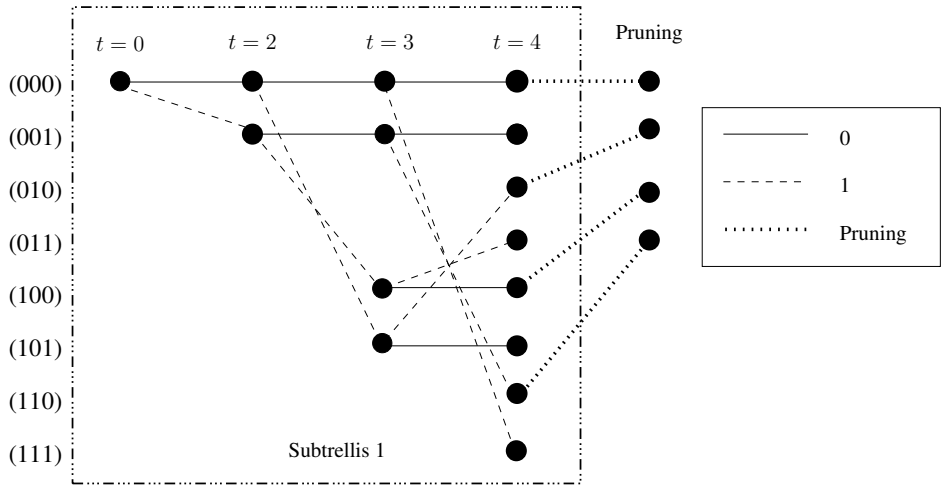


Figure 4.8: Syndrome-based code trellis for the PCM of Eq. (4.20).

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

$$H_b = \begin{pmatrix} 111 \\ 001 \\ 011 \end{pmatrix}. \tag{4.21}$$

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 4.8 shows the first sub-trellis of the syndrome-based code trellis for the PCM H of Eq. (4.20), which has $(2^3 = 8)$ states, since the associated H_b has 3 rows (syndromes). The trellis of Figure 4.8 is constructed as follows:

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

(b) The first sub-trellis is constructed using Eq. (4.21). Here each state is labeled by three syndromes, namely $s_3 s_2 s_1$[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:

$$s_l(t) = s_i(t-1) + \alpha_j h_t, \quad \text{for } j = 0, 1, \ldots, (q-1). \tag{4.22}$$

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

[7] In Section 4.3.1, state was represented by $s_1 s_2 s_3$. Both representations are equivalent.

value of α_j. $s_l(t)$ from $s_i(t-1)$. At $t=1$, we have:

$$s_0(1) = s_0(0) + \alpha_0 h_1 = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} + 0. \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}, \tag{4.23}$$

$$s_1(1) = s_0(0) + \alpha_1 h_1 = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} + 1. \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}. \tag{4.24}$$

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:

$$s_0(2) = s_0(1) + \alpha_0 h_2 = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} + 0. \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}, \tag{4.25}$$

$$s_1(2) = s_0(1) + \alpha_1 h_2 = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} + 1. \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix}. \tag{4.26}$$

$$s_2(2) = s_1(1) + \alpha_0 h_2 = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} + 0. \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}, \tag{4.27}$$

$$s_3(2) = s_1(1) + \alpha_1 h_2 = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} + 1. \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}. \tag{4.28}$$

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 $t=3$.

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

$$s = yH^T. \tag{4.29}$$

Since in Eq. (4.20) only the first three ($n=3$) elements of the first row are non-zero, the first element of s in Eq. (4.29), 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 no longer have to represent this bit in the trellis. 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 [227]. 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).

(d) Step 2 is repeated in order 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 the sake of interconnecting them is repeated until the length of trellis becomes equivalent to that of the received bit stream.

(e) 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 [228]. For the error trellis, the pruning is carried out on the basis of the syndrome of the received sequence, rather than the zero syndrome.

4.4 Block Syndrome Decoding

4.4.1 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, leading to the Block Syndrom Decoding (BSD) concept of [216, 217].

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. This must be long enough to ensure that the performance of BSD approaches 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, the choice of $L_{\min}$ hast to strike 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

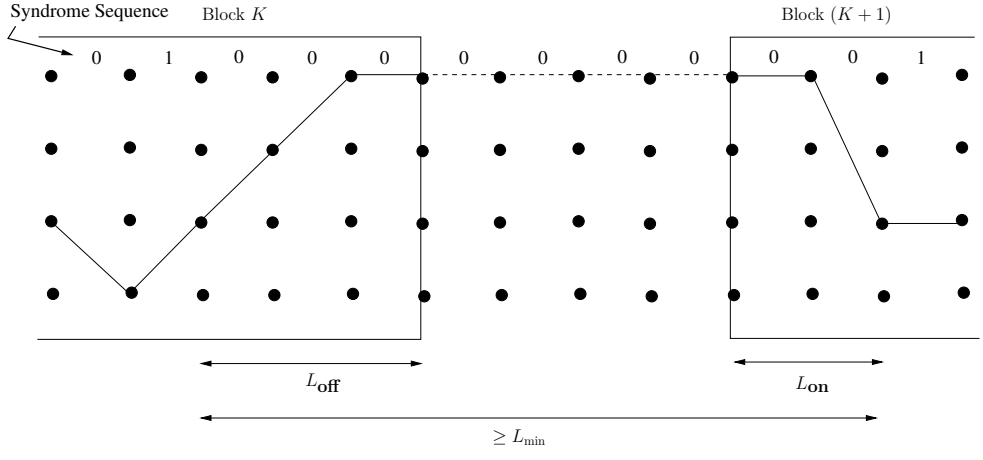


Figure 4.9: Design parameters of BSD [216].

L_{on}, where we have $L_{\text{min}} = (L_{\text{off}} + L_{\text{on}} + 1)$ [216], as shown in Figure 4.9. Here, L_{off} denotes the number of consecutive zero syndromes after which the decoder can be safely switched off. Therefore this 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 to be error-free. This ensures that the trellis of the erroneous sub-blocks starts from and terminates at the all-zero state.

4.4.2 Block Syndrome Decoder for TTCM

Turbo Trellis-Coded Modulation (TTCM) [229] constitutes a bandwidth-efficient near-capacity joint modulation/coding solution, which relies on the classic turbo coding architecture, but involves the bandwidth-efficient Trellis-Coded Modulation (TCM) [230] instead of the constituent convolutional codes. More explicitly, the constituent TCM codes, which can be optimally designed using EXtrinsic Information Transfer (EXIT) charts [231], 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 [216, 219] for TTCM.

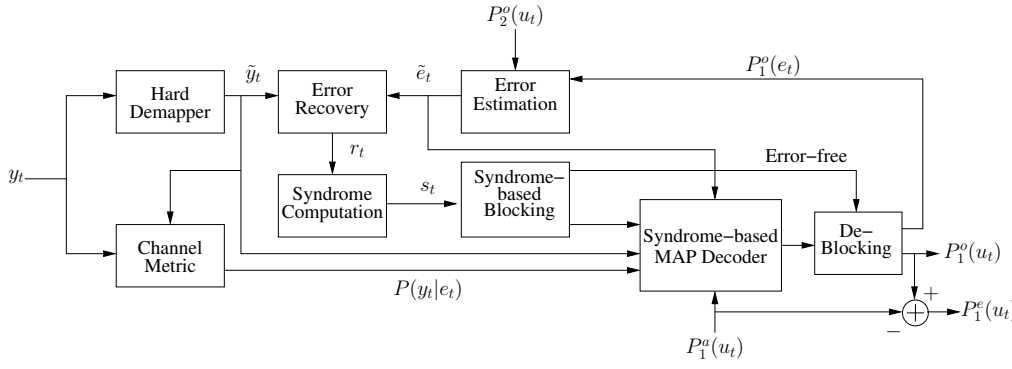


Figure 4.10: Schematic of the proposed BSD-TTCM Decoder. *Only one constituent decoder is shown here.* $P_i^a[.]$, $P_i^e[.]$ *and* $P_i^o[.]$ *are the a-priori, extrinsic and a-posteriori probabilities related to the* i*th decoder;* e_t *is the channel error on the transmitted symbol and* u_t *is the information part of the* t*th channel error* e_t*.* ©Babar *et al.* [224]

4.4.2.1 System Model

Figure 4.10 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:

$$\tilde{y}_t = \arg\min_i (y_t - h_t x_i), \tag{4.30}$$

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

$$y_t = h_t x_t + n_t. \tag{4.31}$$

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:

$$x_t = \mu(c_t), \tag{4.32}$$

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 [229]. Consequently, the parity bits of the corresponding hard-demapped punctured symbols are set to zero [229] in the 'Hard Demapper' block of Figure 4.10. 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:

$$s = rH^T, \tag{4.33}$$

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:

$$r_t^{(j)} = \tilde{y}_t^{(j)} \oplus \tilde{e}_t^{(j)}, \tag{4.34}$$

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 the 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.

4.4.2.2 Syndrome-Based MAP Decoder

We have invoked the syndrome-based MAP decoder of [220] in the BSD-TTCM of Figure 4.10. 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 [210, 226]. 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 [226]. The classic MAP algorithm [208] 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:

$$\mathrm{P}(y_t|e_t) = \frac{1}{2\pi\sigma^2} \cdot e^{-\frac{|y_t - h_t\tilde{x}_t|^2}{2\sigma^2}}, \tag{4.35}$$

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

$$\tilde{x}_t = \mu\left(\tilde{c}_t\right), \tag{4.36}$$

for,

$$\tilde{c}_t^{(j)} = \tilde{y}_t^{(j)} \oplus e_t^{(j)}. \tag{4.37}$$

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:

$$\mathrm{P}^o(u_t) = \sum_{\substack{(\hat{\tau},\tau) \Rightarrow \\ u_t = m}} \gamma_t(\hat{\tau}, \tau) \cdot \alpha_{t-1}(\hat{\tau}) \cdot \beta_t(\tau), \tag{4.38}$$

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:

$$\begin{aligned} \gamma_t(\hat{\tau}, \tau) &= \mathrm{P}^a(u_t) \; . \; \mathrm{P}(y_t|e_t), \\ \alpha_t(\tau) &= \sum_{\text{all } \hat{\tau}} \gamma_t(\hat{\tau}, \tau) \; . \; \alpha_{t-1}(\hat{\tau}), \\ \beta_{t-1}(\hat{\tau}) &= \sum_{\text{all } \tau} \gamma_t(\hat{\tau}, \tau) \; . \; \beta_t(\tau), \end{aligned} \tag{4.39}$$

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. $\mathrm{P}^a(u_t) = 1/M$.

4.4.2.3 Error Estimation

Similar to the bit-wise pre-correction sequence proposed in [219] 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 throughout the ongoing iterations. While the extrinsic information was used in [219] for the estimating the pre-correction sequence, we have improved the estimation by using the APP instead of the extrinsic information. 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 4.11 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 12 000 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 4.11. Observe that the differences decrease at each successive iteration, eventually approaching zero at the 6th iteration. Furthermore, the Hamming weight w_h of the syndrome closely follows the same trend.

4.4.2.4 Syndrome-Based Blocking

The Hamming weight of the syndrome sequence of Eq. (4.33) tends to decrease upon increasing the SNR, 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 either the number of iterations or the SNR, the syndrome tends to exhibit longer sequences of zeros, which indicates near-error-free transmission. This fact can be exploited to partition the received blocks into error-free and error-infested segments, as proposed in [216, 219]. This is achieved by heuristically

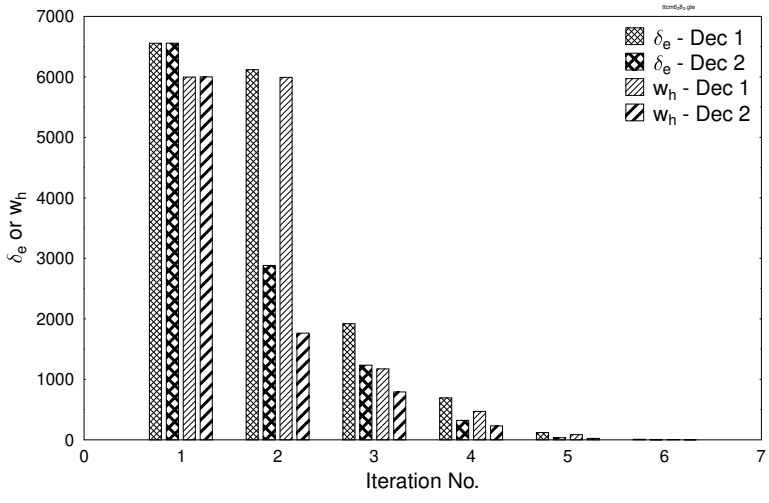


Figure 4.11: Variation in the number of differences (δ_e) between the actual and estimated error and Hamming weight (w_h) of the syndrome with increasing iterations at $E_b/N_0 = 3.8$ dB. ©Babar *et al.* [224]

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 [216, 219]. 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.

Please bear in mind in this context that the design parameter $L_{\min}$ must be chosen by striking 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, albeit at the expense of an increased decoding complexity.

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

Table 4.1: 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 4.2: Optimum $L_{\min}$ for the TTCM of Table 4.1 operating over an AWGN channel. ©Babar *et al.* [224]

4.5 Results and Discussions

4.5.1 Performance of BSD-TTCM over AWGN Channel

In order to quantify the decoding complexity reduction achieved by the BSD-TTCM scheme, we have analyzed the performance of TTCM in the context of an AWGN channel using the parameters of Table 4.1. Furthermore, we have heuristically optimized the design parameter $L_{\min}$ while ensuring that the BSD-TTCM attains a similar BER to 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 4.2. 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 4.1 as well as on the channel type. The BER performance of our BSD-TTCM based on the design parameter $L_{\min}$ of Table 4.2 is compared to that of the conventional TTCM decoder in Figure 4.12. Both decoding schemes exhibit a similar performance. The corresponding reduction in the decoding complexity is quantified in Figure 4.13 and Figure 4.14 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.

Observe in Figure 4.13 that 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 increases for each iteration, reaching a maximum of 45% for the 6th iteration at 3.5 dB. When we further increase $L_{\min}$ 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:

(a) An increased $L_{\min}$ would reduce the number of error-free blocks.

(b) An increased E_b/N_0 would decrease the Hamming weight of the syndrome sequence and, therefore, increase the number of error-free blocks.

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

Figure 4.14 quantifies the decoding complexity in terms of the equivalent number of decoding iterations. We may observe in Figure 4.14 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 [219] in Figure 4.14. The scheme considered outperforms ET by at high SNRs. The complexity may be further reduced by increasing $L_{\min}$. However, as discussed before, this will incur a BER performance degradation.

4.5.2 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 4.1. The corresponding optimum design parameter $L_{\min}$ for increasing the SNR in the high-SNR region (i.e. for $E_b/N_0 \geq 6.2$) are listed in Table 4.3. Figure 4.15 compares the resultant BER performance of the proposed BSD-TTCM to that of the conventional TTCM. As seen from Figure 4.15, 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 4.16 and Figure 4.17 in terms of the 'percentage of no-decoding' and 'equivalent number of iterations', respectively.

In Figure 4.16, 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

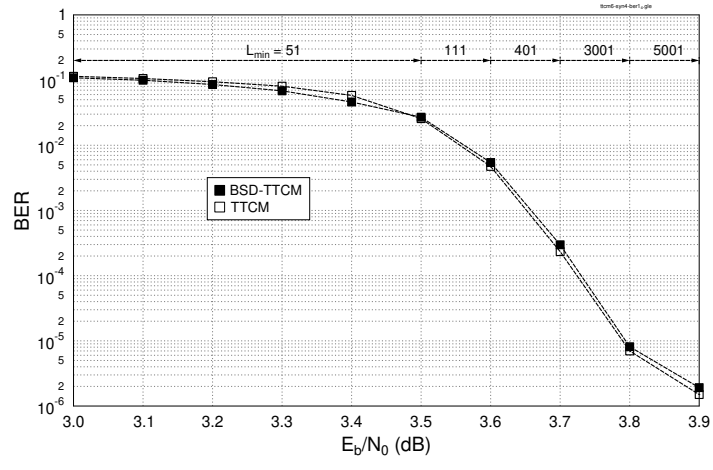


Figure 4.12: Comparison of the BER performance curve of BSD-TTCM with the conventional TTCM decoding over an AWGN channel. The corresponding TTCM parameters are summarized in Table 4.1, while the optimized L_{min} are listed in Table 4.2. ©Babar *et al.* [224]

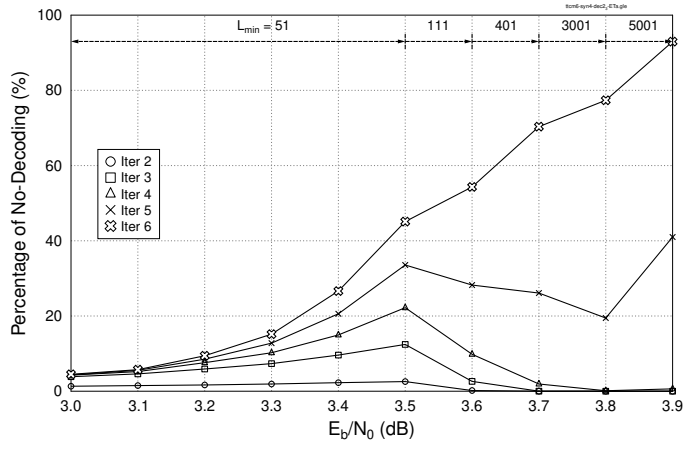


Figure 4.13: Reduction in decoding complexity of the BSD-TTCM of Figure 4.12, quantified in terms of the 'percentage of no-decoding'. ©Babar *et al.* [224]

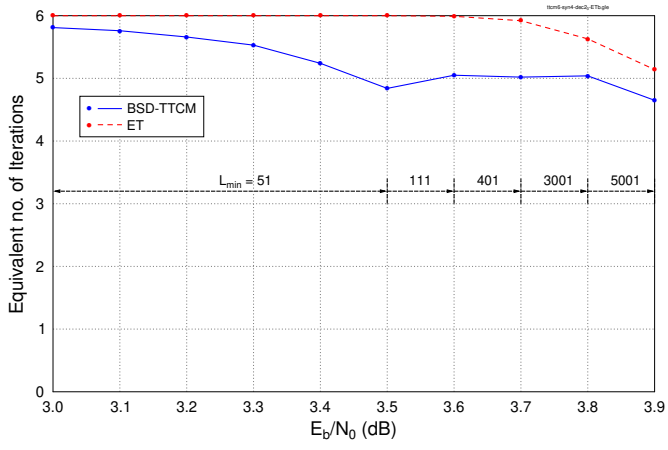


Figure 4.14: Reduction in decoding complexity of the BSD-TTCM of Figure 4.12, quantified in terms of the 'equivalent number of iterations'. ©Babar *et al.* [224]

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 4.3: Optimum $L_{\min}$ for the TTCM of Table 4.1 operating over an uncorrelated Rayleigh fading channel.

during the iterations spanning from 2 to 4, while it increases during the iterations 5 and 6. This behaviour is similar to that observed in Figure 4.13. We may also notice in Figure 4.16 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 in the context of an AWGN channel in Figure 4.13.

Figure 4.17 plots the corresponding decoding complexity in terms of the effective number of iterations. Increasing 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 remains the same. Hence, BSD-TTCM yields a decoding complexity reduction of around 12% in the high-SNR region, when operating in uncorrelated Rayleigh fading channels, which is slightly less than the 17% reduction observed for the AWGN channel in Figure 4.13. 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,

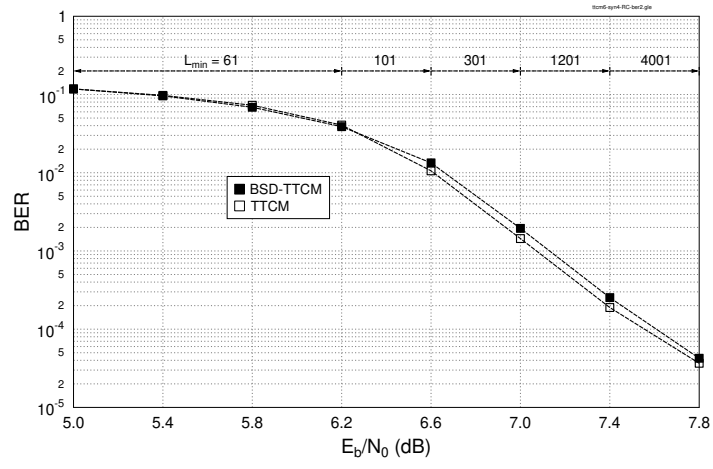


Figure 4.15: Comparison of the BER performance curve of BSD-TTCM with the conventional TTCM decoding over an uncorrelated Rayleigh fading channel. The corresponding TTCM parameters are summarized in Table 4.1, while the optimized $L_{\min}$ are listed in Table 4.3.

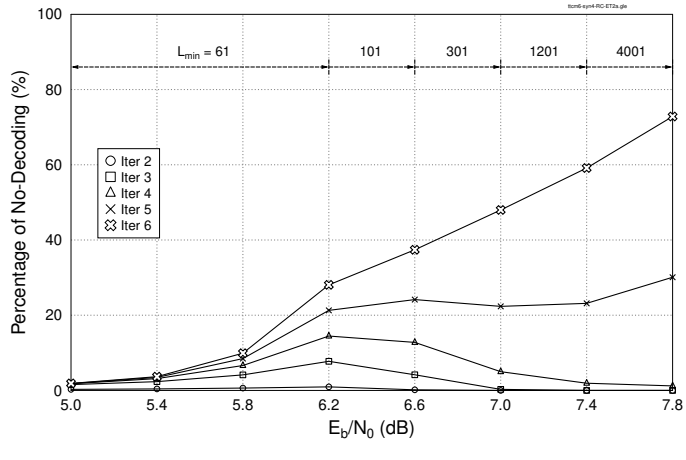


Figure 4.16: Reduction in decoding complexity of the BSD-TTCM of Figure 4.15, quantified in terms of the 'percentage of no-decoding'.

but the attainable complexity savings are still quite significant. In Figure 4.17, we have also benchmarked the performance of our proposed BSD-TTCM decoder against the conventional hard-decision aided ET criterion of [219]. We may observe in Figure 4.17 that BSD-TTCM outperforms ET for all SNRs.

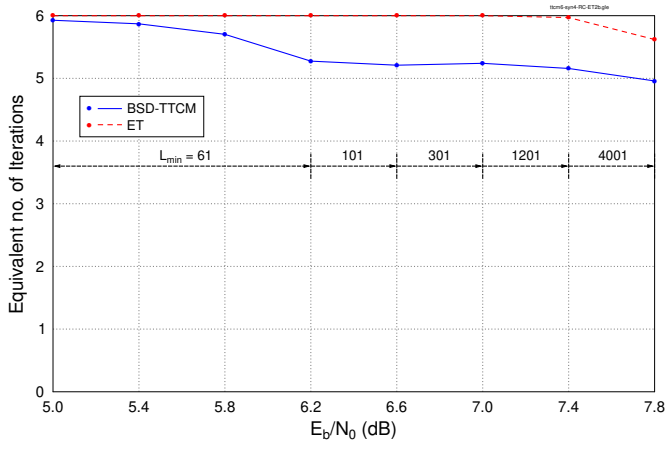


Figure 4.17: Reduction in decoding complexity of the BSD-TTCM of Figure 4.15, quantified in terms of the 'equivalent number of iterations'.

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 4.4: 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 4.1.

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

The high-SNR ET scheme, which we have used in Section 4.5.1 and 4.5.2 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 this hypothesis, 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 4.4.

The design parameter $L_{\min}$ of Table 4.4 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 4.18. The corresponding decoding complexity is analyzed in Figure 4.19 and Figure 4.20. Reducing the frame length from $12,000$ to 500 symbols increases the percentage of non-decoded symbols at each iteration, as we can observe by comparing Figure 4.19 with Figure 4.13. Consequently,

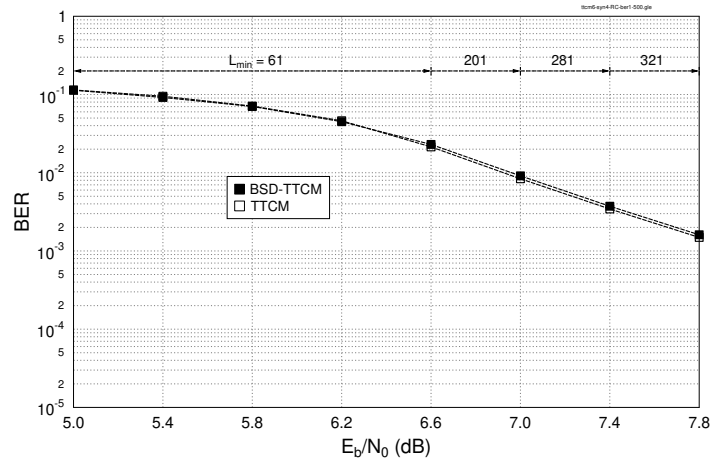


Figure 4.18: Comparison of the BER performance curve of BSD-TTCM with the conventional TTCM decoding over an uncorrelated Rayleigh fading channel using the TTCM parameters of Table 4.1 but with a reduced frame length of 500. The corresponding optimized $L_{\min}$ are listed in Table 4.3.

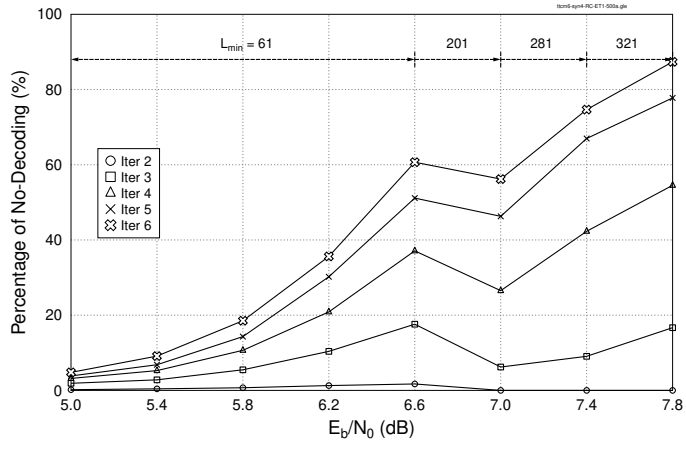


Figure 4.19: Reduction in decoding complexity of the BSD-TTCM of Figure 4.18, quantified in terms of the 'percentage of no-decoding'.

BSD-TTCM still out performs the ET technique, as shown in Figure 4.20.

4.6 Summary and Conclusions

Since quantum codes invoke the classical syndrome decoding based approach, as illustrated in Figure 4.1, in this chapter we discussed these decoding techniques operating in a classical chan-

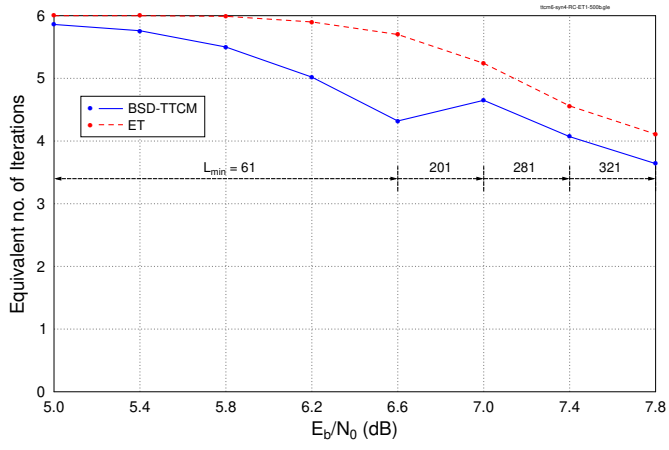


Figure 4.20: Reduction in decoding complexity of the BSD-TTCM of Figure 4.18, quantified in terms of the 'equivalent number of iterations'.

nel. In contrast to the widely used codeword based 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 4.2, 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 on 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. (4.2), while the corresponding standard array of Eq. (4.1) 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 4.3.1 and 4.3.2, respectively. It was shown in Figure 4.6 and Figure 4.7 that the conventional code trellis and the syndrome-based error trellis are equivalent. More explicitly, each 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 4.4, we detailed the reduced-complexity BSD approach. In particular, we conceived a syndrome-based block decoding approach for TTCM in Section 4.4.2. 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 4.5.

In Section 4.5.1, 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 4.12 that the design parameter $L_{\min}$ may be heuristically optimized upon increasing the SNR values for the sake of achieving the same BER performance as that of

Channel	SNR Range	No-Decoding		$\mathtt{I}_{\mathbf{avg}}$	Reduction w.r.t. Early Termination	Reference Figures
		$\mathtt{I}=5$	$\mathtt{I}=6$			
AWGN	≥ 3.5	$\geq 20\%$	$\geq 45\%$	≤ 5	≥ 0.5 iteration	Figure 4.13 and Figure 4.14
Rayleigh	≥ 6.2	$\geq 20\%$	$\geq 30\%$	≤ 5.3	≥ 0.5 iteration	Figure 4.16 and Figure 4.17

Table 4.5: Summary of the achieved decoding complexity reduction, when BSD-TTCM is invoked using the simulation parameters of Table 4.1. Decoding complexity is quantified in terms of the percentage of no-decoding as well as the equivalent number of iterations $\mathtt{I}_{\mathrm{avg}}$. Complexity reduction is also compared with the high-SNR early termination technique.

a classic TTCM decoder. We further quantified the complexity reductions attained in terms of the percentage of non-decoded blocks at each iteration index as well as the number of effective decoding iterations in Figure 4.13 and Figure 4.14, respectively. Quantitatively, we demonstrated in Figure 4.13 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 4.14 that reduced-complexity technique advocated outperforms the ET scheme at all SNRs. These results are summarized in Table 4.5[8].

We further extended our analysis to the performance of BSD-TTCM over an uncorrelated Rayleigh fading channel in Section 4.5.2. 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 4.15. It was observed in Figure 4.16 and Figure 4.17 and that 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 4.16 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 4.5.3, we furthermore evaluated the performance of our BSD-TTCM scheme for a short frame length of 500 symbols. It was observed in Figure 4.20 that BSD-TTCM outperforms the popular ET technique, regardless of the frame length. These results are also tabulated in Table 4.5.

The classical-quantum communication system, which we conceived in Chapter 5, 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 isomorphism of Chapter **??** and on the classical syndrome decoding approach of Chapter 4.

[8] 'With respect to' is abbreviated as 'w.r.t.' in Table 4.5.

Bibliography

[1] E. Schrödinger, "Die gegenwärtige Situation in der Quantenmechanik," *Naturwissenschaften*, vol. 23, no. 49, pp. 823–828, 1935.

[2] J. D. Trimmer, "The present situation in quantum mechanics: A translation of Schrödinger's "Cat Paradox" paper," *Proceedings of the American Philosophical Society*, vol. 124, no. 5, pp. 323–338, 1980.

[3] S. Imre and F. Balazs, *Quantum Computing and Communications: An Engineering Approach*. John Wiley & Sons, 2005.

[4] 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. [Online]. Available: http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=6515077

[5] 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.

[6] 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.

[7] 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: IEEE Computer Society, 1994, pp. 124–134. [Online]. Available: http://portal.acm.org/citation.cfm?id=1398518.1399018

[8] L. K. Grover, "A fast quantum mechanical algorithm for database search," in *Proceedings of the Twenty-eighth Annual ACM Symposium on Theory of Computing*, ser. STOC '96. New York, NY, USA: ACM, 1996, pp. 212–219. [Online]. Available: http://doi.acm.org/10.1145/237814.237866

[9] "Moore's Law - The Number of transistors on integrated circuit chips (1971-2018)," https://ourworldindata.org/technological-progress, accessed: 2019-08-21.

[10] G. E. Moore, "Cramming more components onto integrated circuits," *Electronics Magazine*, pp. 114–117, 1965.

[11] M. M. Waldrop, "The chips are down for Moore's law," *Nature News*, vol. 530, pp. 144–147, 2016.

[12] 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 (Special Centennial Issue)*, vol. 100, pp. 1853–1888, 2012.

[13] R. P. Feynman, "Simulating physics with computers," *International Journal of Theoretical Physics*, vol. 21, no. 6, pp. 467–488, 1982.

[14] D. E. Deutsch, "Quantum theory, the Church–Turing principle and the universal quantum computer," *Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, vol. 400, no. 1818, pp. 97–117, 1985.

[15] P. W. Shor, "Algorithms for quantum computation: Discrete logarithms and factoring," in *Proceedings of 35th Annual Symposium on Foundations of Computer Science (FOCS)*, 1994, pp. 124–134.

[16] ——, "Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer," *SIAM Review*, vol. 41, no. 2, pp. 303–332, 1999.

[17] L. K. Grover, "A fast quantum mechanical algorithm for database search," in *Proceedings of 28th Annual ACM Symposium on Theory of Computing*, 1996, pp. 212–219.

[18] M. Boyer, G. Brassard, P. Høyer, and A. Tapp, "Tight bounds on quantum searching," *arXiv preprint quant-ph/9605034*, 1996.

[19] C. Durr and P. Høyer, "A quantum algorithm for finding the minimum," *arXiv preprint quant-ph/9607014*, 1996.

[20] L. K. Grover, "Quantum mechanics helps in searching for a needle in a haystack," *Physical Review Letters*, vol. 79, no. 2, 1997.

[21] C. Zalka, "Grover's quantum searching algorithm is optimal," *Physical Review A*, vol. 60, no. 4, 1999.

[22] G. Brassard, F. Dupuis, S. Gambs, and A. Tapp, "An optimal quantum algorithm to approximate the mean and its application for approximating the median of a set of points over an arbitrary distance," *arXiv preprint arXiv:1106.4267*, 2011.

[23] 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.

[24] ——, "Fixed-complexity quantum-assisted multi-user detection for CDMA and SDMA," *IEEE Transactions on Communications*, vol. 62, no. 3, pp. 990–1000, 2014.

[25] 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.

[26] 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.

[27] P. Botsinis, D. Alanis, Z. Babar, S. X. Ng, and L. Hanzo, "Noncoherent quantum multiple symbol differential detection for wireless systems," *IEEE Access*, vol. 3, pp. 569–598, 2015.

[28] ——, "Iterative quantum-assisted multi-user detection for multi-carrier interleave division multiple access systems," *IEEE Transactions on Communications*, vol. 63, no. 10, pp. 3713–3727, 2015.

[29] D. Alanis, P. Botsinis, Z. Babar, S. X. Ng, and L. Hanzo, "Non-dominated quantum iterative routing optimization for wireless multihop networks," *IEEE Access*, vol. 3, pp. 1704–1728, 2015.

[30] P. Botsinis, D. Alanis, Z. Babar, H. V. Nguyen, D. Chandra, S. X. Ng, and L. Hanzo, "Quantum search algorithms for wireless communications," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 2, pp. 1209–1242, 2018.

[31] C. H. Bennett, "Quantum cryptography: Public key distribution and coin tossing," in *Proceedings on IEEE International Conference on Computer System and Signal Processing*, 1984, pp. 175–179.

[32] A. K. Ekert, "Quantum cryptography based on Bell's theorem," *Physical Review Letters*, vol. 67, no. 6, 1991.

[33] C. H. Bennett and S. J. Wiesner, "Communication via one-and two-particle operators on Einstein-Podolsky-Rosen states," *Physical Review Letters*, vol. 69, no. 20, 1992.

[34] 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," *Physical Review Letters*, vol. 70, no. 13, 1993.

[35] F.-G. Deng, G. L. Long, and X.-S. Liu, "Two-step quantum direct communication protocol using the Einstein-Podolsky-Rosen pair block," *Physical Review A*, vol. 68, no. 4, 2003.

[36] F.-G. Deng and G. L. Long, "Secure direct communication with a quantum one-time pad," *Physical Review A*, vol. 69, no. 5, 2004.

[37] N. L. Piparo and M. Razavi, "Long-distance trust-free quantum key distribution," *IEEE Journal of Selected Topics in Quantum Electronics*, vol. 21, no. 3, pp. 123–130, 2014.

[38] C. Panayi, M. Razavi, X. Ma, and N. Lütkenhaus, "Memory-assisted measurement-device-independent quantum key distribution," *New Journal of Physics*, vol. 16, no. 4, 2014.

[39] M. Razavi, "Multiple-access quantum key distribution networks," *IEEE Transactions on communications*, vol. 60, no. 10, pp. 3071–3079, 2012.

[40] A. S. Cacciapuoti, M. Caleffi, R. Van Meter, and L. Hanzo, "When entanglement meets classical communications: Quantum teleportation for the quantum internet," *arXiv preprint arXiv:1907.06197*, 2019.

[41] G. De Lange, Z. Wang, D. Riste, V. Dobrovitski, and R. Hanson, "Universal dynamical decoupling of a single solid-state spin from a spin bath," *Science*, vol. 330, no. 6000, pp. 60–63, 2010.

[42] T. Van der Sar, Z. Wang, M. Blok, H. Bernien, T. Taminiau, D. Toyli, D. A. Lidar, D. Awschalom, R. Hanson, and V. Dobrovitski, "Decoherence-protected quantum gates for a hybrid solid-state spin register," *Nature*, vol. 484, no. 7392, pp. 82–86, 2012.

[43] J. L. O'Brien, G. Pryde, A. Gilchrist, D. James, N. Langford, T. Ralph, and A. White, "Quantum process tomography of a controlled-NOT gate," *Physical Review Letters*, vol. 93, no. 8, 2004.

[44] A. Politi, M. J. Cryan, J. G. Rarity, S. Yu, and J. L. O'Brien, "Silica-on-silicon waveguide quantum circuits," *Science*, vol. 320, no. 5876, pp. 646–649, 2008.

[45] A. Politi, J. C. Matthews, and J. L. O'brien, "Shor's quantum factoring algorithm on a photonic chip," *Science*, vol. 325, no. 5945, pp. 1221–1221, 2009.

[46] L. DiCarlo, J. M. Chow, J. M. Gambetta, L. Bishop, B. Johnson, D. Schuster, J. Majer, A. Blais, L. Frunzio, S. Girvin, *et al.*, "Demonstration of two-qubit algorithms with a superconducting quantum processor," *Nature*, vol. 460, no. 7252, p. 240, 2009.

[47] E. Martín-López, A. Laing, T. Lawson, R. Alvarez, X.-Q. Zhou, and J. L. O'Brien, "Experimental realization of Shor's quantum factoring algorithm using qubit recycling," *Nature Photonics*, vol. 6, no. 11, pp. 773–776, 2012.

[48] J. M. Chow, J. M. Gambetta, A. Córcoles, S. T. Merkel, J. A. Smolin, C. Rigetti, S. Poletto, G. A. Keefe, M. B. Rothwell, J. Rozen, *et al.*, "Universal quantum gate set approaching fault-tolerant thresholds with superconducting qubits," *Physical Review Letters*, vol. 109, no. 6, 2012.

[49] M. Veldhorst, J. Hwang, C. Yang, A. Leenstra, B. De Ronde, J. Dehollain, J. Muhonen, F. Hudson, K. M. Itoh, A. Morello, *et al.*, "An addressable quantum dot qubit with fault-tolerant control-fidelity," *Nature Nanotechnology*, vol. 9, no. 12, pp. 981–985, 2014.

[50] M. Veldhorst, C. Yang, J. Hwang, W. Huang, J. Dehollain, J. Muhonen, S. Simmons, A. Laucht, F. Hudson, K. M. Itoh, *et al.*, "A two-qubit logic gate in silicon," *Nature*, vol. 526, no. 7573, pp. 410–414, 2015.

[51] S. Weidt, J. Randall, S. Webster, K. Lake, A. Webb, I. Cohen, T. Navickas, B. Lekitsch, A. Retzker, and W. K. Hensinger, "Trapped-ion quantum logic with global radiation fields," *Physical review letters*, vol. 117, no. 22, 2016.

[52] B. Lekitsch, S. Weidt, A. G. Fowler, K. Mølmer, S. J. Devitt, C. Wunderlich, and W. K. Hensinger, "Blueprint for a microwave trapped ion quantum computer," *Science Advances*, vol. 3, no. 2, 2017.

[53] D. P. DiVincenzo, "The physical implementation of quantum computation," *Fortschritte der Physik*, vol. 48, no. 9-11, pp. 771–783, 2000.

[54] P. A. Dirac, "A new notation for quantum mechanics," *Proceedings of the Cambridge Philosophical Society*, vol. 35, pp. 416–418, 1939.

[55] S. Aaronson and D. Gottesman, "Improved simulation of stabilizer circuits," *Physical Review A*, vol. 70, no. 5, 2004.

[56] J. Preskill, "Reliable quantum computers," *Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, vol. 454, no. 1969, pp. 385–410, 1998.

[57] C. E. Shannon, "A mathematical theory of communication," *Bell System Technical Journal*, vol. 27, no. 3, pp. 379–423, 1948.

[58] P. W. Shor, "Scheme for reducing decoherence in quantum computer memory," *Physical Review A*, vol. 52, no. 4, 1995.

[59] A. M. Steane, "Error correcting codes in quantum theory," *Physical Review Letters*, vol. 77, no. 5, 1996.

[60] R. Laflamme, C. Miquel, J. P. Paz, and W. H. Zurek, "Perfect quantum error correcting code," *Physical Review Letters*, vol. 77, no. 1, 1996.

[61] D. Gottesman, "Class of quantum error-correcting codes saturating the quantum Hamming bound," *Physical Review A*, vol. 54, no. 3, 1996.

[62] A. K. Ekert and C. Macchiavello, "Quantum error correction for communication," *Physical Review Letters*, vol. 77, no. 12, 1996.

[63] D. Gottesman, "Stabilizer codes and quantum error correction," Ph.D. dissertation, California Institute of Technology, 1997.

[64] A. R. Calderbank, E. M. Rains, P. W. Shor, and N. J. A. Sloane, "Quantum error correction via codes over GF(4)," *IEEE Transactions on Information Theory*, vol. 44, no. 4, pp. 1369–1387, 1998.

[65] M. Grassl and T. Beth, "Quantum BCH codes," in *Proceedings of International Symposium Theoretical Electrical Engineering*, 1999, pp. 207–212.

[66] M. Grassl, W. Geiselmann, and T. Beth, "Quantum Reed-Solomon codes," in *Applied Algebra, Algebraic Algorithms and Error-correcting Codes*. Springer Berlin Heidelberg, 1999, pp. 231–244.

[67] A. M. Steane, "Quantum Reed-Muller codes," *IEEE Transactions on Information Theory*, vol. 45, no. 5, pp. 1701–1703, 1999.

[68] G. Bowen, "Entanglement required in achieving entanglement-assisted channel capacities," *Physical Review A*, vol. 66, no. 5, 2002.

[69] T. A. Brun, I. Devetak, and M.-H. Hsieh, "Correcting quantum errors with entanglement," *Science*, vol. 314, no. 5798, pp. 436–439, 2006.

[70] H. Ollivier and J.-P. Tillich, "Description of a quantum convolutional code," *Physical Review Letters*, vol. 91, no. 17, 2003.

[71] M. M. Wilde and T. A. Brun, "Entanglement-assisted quantum convolutional coding," *Physical Review A*, vol. 81, no. 4, 2010.

[72] M. S. Postol, "A proposed quantum low density parity check code," *arXiv preprint quant-ph/0108131*, 2001.

[73] D. J. MacKay, G. Mitchison, and P. L. McFadden, "Sparse-graph codes for quantum error correction," *IEEE Transactions on Information Theory*, vol. 50, no. 10, pp. 2315–2330, 2004.

[74] T. Camara, H. Ollivier, and J.-P. Tillich, "Constructions and performance of classes of quantum LDPC codes," *arXiv preprint quant-ph/0502086*, 2005.

[75] ——, "A class of quantum LDPC codes: Construction and performances under iterative decoding," in *Proceedings of IEEE International Symposium on Information Theory (ISIT)*, 2007, pp. 811–815.

[76] M.-H. Hsieh, T. A. Brun, and I. Devetak, "Entanglement-assisted quantum quasicyclic low-density parity-check codes," *Physical Review A*, vol. 79, no. 3, 2009.

[77] Z. Babar, P. Botsinis, D. Alanis, S. X. Ng, and L. Hanzo, "Fifteen years of quantum LDPC coding and improved decoding strategies," *Access, IEEE*, vol. 3, pp. 2492–2519, 2015.

[78] D. Poulin, J.-P. Tillich, and H. Ollivier, "Quantum serial turbo codes," *IEEE Transactions on Information Theory*, vol. 55, no. 6, pp. 2776–2798, 2009.

[79] M. M. Wilde and M.-H. Hsieh, "Entanglement boosts quantum turbo codes," in *Proceedings of IEEE International Symposium on Information Theory (ISIT)*, 2011, pp. 445–449.

[80] M. M. Wilde, M.-H. Hsieh, and Z. Babar, "Entanglement-assisted quantum turbo codes," *IEEE Transactions on Information Theory*, vol. 60, no. 2, pp. 1203–1222, 2014.

[81] 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.

[82] J. M. Renes, F. Dupuis, and R. Renner, "Efficient polar coding of quantum information," *Physical Review Letters*, vol. 109, no. 5, 2012.

[83] M. M. Wilde and J. M. Renes, "Quantum polar codes for arbitrary channels," in *Proceedings of IEEE International Symposium on Information Theory (ISIT)*, 2012, pp. 334–338.

[84] M. Grassl, T. Beth, and T. Pellizzari, "Codes for the quantum erasure channel," *Physical Review A*, vol. 56, no. 1, 1997.

[85] H. F. Chau, "Quantum convolutional error-correcting codes," *Physical Review A*, vol. 58, no. 2, 1998.

[86] D. A. Lidar and K. B. Whaley, "Decoherence-free subspaces and subsystems," in *Irreversible Quantum Dynamics*. Springer, 2003, pp. 83–120.

[87] A. Shabani and D. A. Lidar, "Theory of initialization-free decoherence-free subspaces and subsystems," *Physical Review A*, vol. 72, no. 4, 2005.

[88] D. Lidar and T. Brun, *Quantum Error Correction*. Cambridge University Press, 2013. [Online]. Available: http://books.google.co.uk/books?id=XV9sAAAAQBAJ

[89] E. Knill and R. Laflamme, "Theory of quantum error-correcting codes," *Physical Review A*, vol. 55, no. 2, 1997.

[90] A. R. Calderbank and P. W. Shor, "Good quantum error-correcting codes exist," *Physical Review A*, vol. 54, no. 2, 1996.

[91] D. Gottesman, "Theory of fault-tolerant quantum computation," *Physical Review A*, vol. 57, no. 1, 1998.

[92] E. Knill, R. Laflamme, and W. H. Zurek, "Resilient quantum computation," *Science*, vol. 279, no. 5349, pp. 342–345, 1998.

[93] ——, "Resilient quantum computation: Error models and thresholds," *Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, vol. 454, no. 1969, pp. 365–384, 1998.

[94] D. Gottesman, "Fault-tolerant quantum computation with constant overhead," *Quantum Information & Computation*, vol. 14, no. 15-16, pp. 1338–1372, 2014.

[95] J. Akhtman, R. Maunder, N. Bonello, and L. Hanzo, "Closed-form approximation of maximum free distance for binary block codes," in *Proceedings of IEEE 70th Vehicular Technology Conference Fall (VTC-Fall)*, 2009, pp. 1–3.

[96] A. M. Steane, "Multiple-particle interference and quantum error correction," *Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, vol. 452, no. 1954, pp. 2551–2577, 1996.

[97] P. W. Shor, "Fault-tolerant quantum computation," in *Proceedings of 37th Annual Symposium on Foundations of Computer Science (FOCS)*, 1996, pp. 56–65.

[98] A. R. Calderbank, E. M. Rains, P. W. Shor, and N. J. A. Sloane, "Quantum error correction and orthogonal geometry," *Physical Review Letters*, vol. 78, no. 3, 1997.

[99] "IBM Quantum Experience," https://www.research.ibm.com/ibm-q, accessed: 2017-10-01.

[100] A. Y. Kitaev, "Quantum computations: Algorithms and error correction," *Russian Mathematical Surveys*, vol. 52, no. 6, pp. 1191–1249, 1997.

[101] ——, "Fault-tolerant quantum computation by anyons," *Annals of Physics*, vol. 303, no. 1, pp. 2–30, 2003.

[102] M. H. Freedman, D. A. Meyer, and F. Luo, "Z2-systolic freedom and quantum codes," *Mathematics of Quantum Computation*, pp. 287–320, 2002.

[103] S. Bravyi and A. Y. Kitaev, "Quantum codes on a lattice with boundary," *arXiv preprint quant-ph/9811052*, 1998.

[104] C. Horsman, A. G. Fowler, S. J. Devitt, and R. Van Meter, "Surface code quantum computing by lattice surgery," *New Journal of Physics*, vol. 14, no. 12, 2012.

[105] H. Bombin and M. A. Martin-Delgado, "Topological quantum distillation," *Physical Review Letters*, vol. 97, no. 18, 2006.

[106] J. Haah, "Local stabilizer codes in three dimensions without string logical operators," *Physical Review A*, vol. 83, no. 4, 2011.

[107] G. Zémor, "On Cayley graphs, surface codes, and the limits of homological coding for quantum error correction," in *Coding and Cryptology*. Springer, 2009, pp. 259–273.

[108] A. Couvreur, N. Delfosse, and G. Zemor, "A construction of quantum LDPC codes from Cayley graphs," *IEEE Transactions on Information Theory*, vol. 59, no. 9, pp. 6087–6098, 2013.

[109] N. Delfosse, "Tradeoffs for reliable quantum information storage in surface codes and color codes," in *Proceedings of IEEE International Symposium on Information Theory (ISIT)*, 2013, pp. 917–921.

[110] J.-P. Tillich and G. Zémor, "Quantum LDPC codes with positive rate and minimum distance proportional to the square root of the blocklength," in *Proceedings of IEEE International Symposium on Information Theory (ISIT)*, 2009, pp. 799–803.

[111] A. A. Kovalev and L. P. Pryadko, "Improved quantum hypergraph-product LDPC codes," in *Proceedings of IEEE International Symposium on Information Theory (ISIT)*, 2012, pp. 348–352.

[112] J.-P. Tillich and G. Zémor, "Quantum LDPC codes with positive rate and minimum distance proportional to the square root of the blocklength," *IEEE Transactions on Information Theory*, vol. 60, no. 2, pp. 1193–1202, 2014.

[113] S. Bravyi and M. B. Hastings, "Homological product codes," in *Proceedings of the 46th Annual ACM Symposium on Theory of Computing*, 2014, pp. 273–282.

[114] S. Bravyi and J. Haah, "Quantum self-correction in the 3D cubic code model," *Physical Review Letters*, vol. 111, no. 20, 2013.

[115] L. Guth and A. Lubotzky, "Quantum error correcting codes and 4-dimensional arithmetic hyperbolic manifolds," *Journal of Mathematical Physics*, vol. 55, no. 8, 2014.

[116] H. Bombín, "Dimensional jump in quantum error correction," *New Journal of Physics*, vol. 18, no. 4, 2016.

[117] W. Zeng and L. P. Pryadko, "Higher-dimensional quantum hypergraph-product codes with finite rates," *Physical Review Letters*, vol. 122, no. 23, 2019.

[118] P. Piret, *Convolutional Codes: An Algebraic Approach*. MIT Press, Cambridge, MA, 1988.

[119] H. F. Chau, "Quantum convolutional error-correcting codes," *Phys. Rev. A*, vol. 58, pp. 905–909, Aug 1998. [Online]. Available: http://link.aps.org/doi/10.1103/PhysRevA.58.905

[120] ——, "Good quantum-convolutional error-correction codes and their decoding algorithm exist," *Phys. Rev. A*, vol. 60, pp. 1966–1974, Sep 1999. [Online]. Available: http://link.aps.org/doi/10.1103/PhysRevA.60.1966

[121] H. Ollivier and J.-P. Tillich, "Description of a quantum convolutional code," *Phys. Rev. Lett.*, vol. 91, p. 177902, Oct 2003. [Online]. Available: http://link.aps.org/doi/10.1103/PhysRevLett.91.177902

[122] ——, "Quantum convolutional codes: fundamentals," *quant-ph/0401134*, 2004.

[123] 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. [Online]. Available: http://link.aps.org/doi/10.1103/PhysRevLett.77.198

[124] A. Aido De Almeida and J. Palazzo, R., "A concatenated [(4, 1, 3)] quantum convolutional code," in *IEEE Information Theory Workshop*, Oct. 2004, pp. 28 – 33.

[125] G. Forney and S. Guha, "Simple rate-1/3 convolutional and tail-biting quantum error-correcting codes," in *International Symposium on Information Theory*, Sept. 2005, pp. 1028 –1032.

[126] G. D. Forney, M. Grassl, and S. Guha, "Convolutional and tail-biting quantum error-correcting codes," *IEEE Transactions on Information Theory*, vol. 53, no. 3, pp. 865–880, March 2007.

[127] M. Grassl and M. Rotteler, "Quantum block and convolutional codes from self-orthogonal product codes," in *International Symposium on Information Theory*, Sep. 2005, pp. 1018 –1022.

[128] ——, "Constructions of quantum convolutional codes," in *IEEE International Symposium on Information Theory*, June 2007, pp. 816–820.

[129] S. Aly, M. Grassl, A. Klappenecker, M. Rotteler, and P. Sarvepalli, "Quantum convolutional BCH codes," in *10th Canadian Workshop on Information Theory*, June 2007, pp. 180 –183.

[130] 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*, June 2007, pp. 821 –825.

[131] E. Pelchat and D. Poulin, "Degenerate Viterbi decoding," *IEEE Transactions on Information Theory*, vol. 59, no. 6, pp. 3915–3921, 2013.

[132] 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.

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

[134] M. S. Postol, "A proposed quantum low density parity check code," *arXiv:quant-ph/0108131v1*, 2001.

[135] D. MacKay, G. Mitchison, and P. McFadden, "Sparse-graph codes for quantum error correction," *IEEE Transactions on Information Theory*, vol. 50, no. 10, pp. 2315–2330, Oct 2004.

[136] 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.

[137] A. Couvreur, N. Delfosse, and G. Zemor, "A construction of quantum LDPC codes from cayley graphs," in *IEEE International Symposium on Information Theory Proceedings*, 31 2011-aug. 5 2011, pp. 643 –647.

[138] ——, "A construction of quantum LDPC codes from cayley graphs," *IEEE Transactions on Information Theory*, vol. 59, no. 9, pp. 6087–6098, Sept 2013.

[139] S. Aly, "A class of quantum LDPC codes constructed from finite geometries," in *IEEE Global Telecommunications Conference*, 30 2008-dec. 4 2008, pp. 1 –5.

[140] I. Djordjevic, "Quantum LDPC codes from balanced incomplete block designs," *IEEE Communications Letters*, vol. 12, no. 5, pp. 389 –391, may 2008.

[141] 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*, june 2005, pp. 1043 – 1047.

[142] ——, "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.

[143] M. Hagiwara and H. Imai, "Quantum quasi-cyclic LDPC codes," in *IEEE International Symposium on Information Theory*, june 2007, pp. 806 –810.

[144] K. Kasai, M. Hagiwara, H. Imai, and K. Sakaniwa, "Non-binary quasi-cyclic quantum LDPC codes," in *IEEE International Symposium on Information Theory Proceedings*, 31 2011-aug. 5 2011, pp. 653 –657.

[145] ——, "Quantum error correction beyond the bounded distance decoding limit," *IEEE Transactions on Information Theory*, vol. 58, no. 2, pp. 1223 –1230, feb. 2012.

[146] M. Hagiwara, K. Kasai, H. Imai, and K. Sakaniwa, "Spatially coupled quasi-cyclic quantum LDPC codes," in *IEEE International Symposium on Information Theory Proceedings*, 31 2011-aug. 5 2011, pp. 638 –642.

[147] 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*, June 2007, pp. 811–815.

[148] P. Tan and J. Li, "Efficient quantum stabilizer codes: LDPC and LDPC-convolutional constructions," *IEEE Transactions on Information Theory*, vol. 56, no. 1, pp. 476 –491, jan. 2010.

[149] N. Delfosse, "Tradeoffs for reliable quantum information storage in surface codes and color codes," in *IEEE International Symposium on Information Theory Proceedings*, July 2013, pp. 917–921.

[150] 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. [Online]. Available: http://link.aps.org/doi/10.1103/PhysRevA.88.012311

[151] 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, no. 2, pp. 1193–1202, Feb 2014.

[152] D. Poulin and Y. Chung, "On the iterative decoding of sparse quantum codes," *Quantum Info. Comput.*, vol. 8, no. 10, pp. 987–1000, Nov. 2008. [Online]. Available: http://dl.acm.org/citation.cfm?id=2016985.2016993

[153] 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, no. 2, pp. 1231 –1241, feb. 2012.

[154] D. Poulin, J.-P. Tillich, and H. Ollivier, "Quantum serial turbo-codes," in *IEEE International Symposium on Information Theory*, July 2008, pp. 310–314.

[155] ——, "Quantum serial turbo codes," *IEEE Transactions on Information Theory*, vol. 55, no. 6, pp. 2776–2798, June 2009.

[156] 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.

[157] M. M. Wilde, M.-H. Hsieh, and Z. Babar, "Entanglement-assisted quantum turbo codes," *IEEE Transactions on Information Theory*, vol. 60, no. 2, pp. 1203–1222, Feb 2014.

[158] G. Bowen, "Entanglement required in achieving entanglement-assisted channel capacities," *Phys. Rev. A*, vol. 66, p. 052313, Nov 2002. [Online]. Available: http://link.aps.org/doi/10.1103/PhysRevA.66.052313

[159] T. A. Brun, I. Devetak, and M.-H. Hsieh, "Correcting quantum errors with entanglement," *Science*, vol. 314, no. 5798, oct. 2006.

[160] ——, "General entanglement-assisted quantum error-correcting codes," in *IEEE International Symposium on Information Theory*, june 2007, pp. 2101 –2105.

[161] M.-H. Hsieh, I. Devetak, and T. Brun, "General entanglement-assisted quantum error-correcting codes," *Phys. Rev. A*, vol. 76, p. 062313, Dec 2007. [Online]. Available: http://link.aps.org/doi/10.1103/PhysRevA.76.062313

[162] 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. [Online]. Available: http://link.aps.org/doi/10.1103/PhysRevA.79.032340

[163] M. M. Wilde and T. A. Brun, "Entanglement-assisted quantum convolutional coding," *Phys. Rev. A*, vol. 81, p. 042333, Apr 2010. [Online]. Available: http://link.aps.org/doi/10.1103/PhysRevA.81.042333

[164] M. M. Wilde and M.-H. Hsieh, "Entanglement boosts quantum turbo codes," in *IEEE International Symposium on Information Theory Proceedings*, Aug. 2011, pp. 445 – 449.

[165] M. M. Wilde and J. Renes, "Quantum polar codes for arbitrary channels," in *IEEE International Symposium on Information Theory Proceedings*, july 2012, pp. 334 –338.

[166] C. Ballance, T. Harty, N. Linke, M. Sepiol, and D. Lucas, "High-fidelity quantum logic gates using trapped-ion hyperfine qubits," *Physical review letters*, vol. 117, no. 6, 2016.

[167] H. V. Nguyen, Z. Babar, D. Alanis, P. Botsinis, D. Chandra, S. X. Ng, and L. Hanzo, "EXIT-chart aided quantum code design improves the normalised throughput of realistic quantum devices," *IEEE Access*, vol. 4, pp. 10 194–10 209, 2016.

[168] S. Bravyi and A. Y. Kitaev, "Universal quantum computation with ideal Clifford gates and noisy ancillas," *Physical Review A*, vol. 71, no. 2, 2005.

[169] D. Aharonov and M. Ben-Or, "Fault-tolerant quantum computation with constant error rate," *SIAM Journal on Computing*, vol. 38, no. 4, pp. 1207–1282, 2008.

[170] H. Bombín, "Single-shot fault-tolerant quantum error correction," *Physical Review X*, vol. 5, no. 3, 2015.

[171] O. Fawzi, A. Grospellier, and A. Leverrier, "Constant overhead quantum fault-tolerance with quantum expander codes," in *Proceedings of 59th IEEE Annual Symposium on Foundations of Computer Science (FOCS)*, 2018, pp. 743–754.

[172] E. Campbell, "A theory of single-shot error correction for adversarial noise," *Quantum Science and Technology*, vol. 4, no. 2, 2019.

[173] T. Jochym-O'Connor, "Fault-tolerant gates via homological product codes," *Quantum*, vol. 3, 2019.

[174] D. Chandra, Z. Babar, H. V. Nguyen, D. Alanis, P. Botsinis, S. X. Ng, and L. Hanzo, "Quantum topological error correction codes: The classical-to-quantum isomorphism perspective," *IEEE Access*, vol. 6, pp. 13 729–13 757, 2018.

[175] S. Bravyi and B. Terhal, "A no-go theorem for a two-dimensional self-correcting quantum memory based on stabilizer codes," *New Journal of Physics*, vol. 11, no. 4, 2009.

[176] S. Bravyi, D. Poulin, and B. Terhal, "Tradeoffs for reliable quantum information storage in 2D systems," *Physical Review Letters*, vol. 104, no. 5, 2010.

[177] Y. Fujiwara, "Ability of stabilizer quantum error correction to protect itself from its own imperfection," *Physical Review A*, vol. 90, no. 6, 2014.

[178] A. Ashikhmin, C.-Y. Lai, and T. A. Brun, "Robust quantum error syndrome extraction by classical coding," in *Proceedings of IEEE International Symposium on Information Theory (ISIT)*, 2014, pp. 546–550.

[179] ——, "Correction of data and syndrome errors by stabilizer codes," in *Proceedings of IEEE International Symposium on Information Theory (ISIT)*, 2016, pp. 2274–2278.

[180] W. Zeng, A. Ashikhmin, M. Woolls, and L. P. Pryadko, "Quantum convolutional data-syndrome codes," *arXiv preprint arXiv:1902.07395*, 2019.

[181] M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information*. Cambridge University Press, 2000.

[182] D. Gottesman, "The Heisenberg representation of quantum computers," *arXiv preprint quant-ph/9807006*, 1998.

[183] T. Toffoli, "Reversible computing," in *International Colloquium on Automata, Languages, and Programming*. Springer, 1980, pp. 632–644.

[184] P. Botsinis, Z. Babar, D. Alanis, D. Chandra, H. V. Nguyen, S. X. Ng, and L. Hanzo, "Quantum error correction protects quantum search algorithms against decoherence," *Scientific Reports*, vol. 6, 2016.

[185] A. Barenco, C. H. Bennett, R. Cleve, D. P. DiVincenzo, N. Margolus, P. W. Shor, T. Sleator, J. A. Smolin, and H. Weinfurter, "Elementary gates for quantum computation," *Physical Review A*, vol. 52, no. 5, 1995.

[186] U. Fano, "Description of states in quantum mechanics by density matrix and operator techniques," *Reviews of Modern Physics*, vol. 29, no. 1, 1957.

[187] G. M. D'Ariano, M. G. Paris, and M. F. Sacchi, "Quantum tomography," *Advances in Imaging and Electron Physics*, vol. 128, pp. 206–309, 2003.

[188] J. Roffe, "Quantum error correction: An introductory guide," *arXiv preprint arXiv:1907.11157*, 2019.

[189] A. Ashikhmin, "Remarks on bounds for quantum codes," *arXiv preprint quant-ph/9705037*, 1997.

[190] P. K. Sarvepalli and A. Klappenecker, "Degenerate quantum codes and the quantum Hamming bound," *Physical Review A*, vol. 81, no. 3, 2010.

[191] Z. Babar, D. Chandra, H. V. Nguyen, P. Botsinis, D. Alanis, S. X. Ng, and L. Hanzo, "Duality of Quantum and Classical Error Correction Codes: Design Principles and Examples," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 1, pp. 970–1010, 2018.

[192] F. Schmidt-Kaler, S. Gulde, M. Riebe, T. Deuschle, A. Kreuter, G. Lancaster, C. Becher, J. Eschner, H. Häffner, and R. Blatt, "The coherence of qubits based on single Ca+ ions," *Journal of Physics B: Atomic, Molecular and Optical Physics*, vol. 36, no. 3, p. 623, 2003.

[193] L. M. Vandersypen, M. Steffen, G. Breyta, C. S. Yannoni, M. H. Sherwood, and I. L. Chuang, "Experimental realization of Shor's quantum factoring algorithm using nuclear magnetic resonance," *Nature*, vol. 414, no. 6866, pp. 883–887, 2001.

[194] A. Tyryshkin, J. Morton, S. Benjamin, A. Ardavan, G. Briggs, J. Ager, and S. Lyon, "Coherence of spin qubits in silicon," *Journal of Physics: Condensed Matter*, vol. 18, no. 21, p. S783, 2006.

[195] J. Petta, A. C. Johnson, J. Taylor, E. Laird, A. Yacoby, M. D. Lukin, C. Marcus, M. Hanson, and A. Gossard, "Coherent manipulation of coupled electron spins in semiconductor quantum dots," *Science*, vol. 309, no. 5744, pp. 2180–2184, 2005.

[196] P. Bertet, I. Chiorescu, G. Burkard, K. Semba, C. Harmans, D. P. DiVincenzo, and J. Mooij, "Dephasing of a superconducting qubit induced by photon noise," *Physical Review Letters*, vol. 95, no. 25, 2005.

[197] L. Ioffe and M. Mézard, "Asymmetric quantum error-correcting codes," *Physical Review A*, vol. 75, no. 3, 2007.

[198] P. W. Shor, "Scheme for reducing decoherence in quantum computer memory," *Physical Review A*, vol. 52, no. 4, pp. R2493–R2496, Oct. 1995. [Online]. Available: http://dx.doi.org/10.1103/PhysRevA.52.R2493

[199] M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information*. Cambridge University Press, 2000.

[200] L. Ioffe and M. Mézard, "Asymmetric quantum error-correcting codes," *Phys. Rev. A*, vol. 75, p. 032345, Mar 2007. [Online]. Available: http://link.aps.org/doi/10.1103/PhysRevA.75.032345

[201] P. Sarvepalli, A. Klappenecker, and M. Rotteler, "Asymmetric quantum LDPC codes," in *IEEE International Symposium on Information Theory*, july 2008, pp. 305 –309.

[202] A. R. Calderbank and P. W. Shor, "Good quantum error-correcting codes exist," *Phys. Rev. A*, vol. 54, no. 2, pp. 1098–1105, Aug 1996.

[203] Y. Fujiwara and V. Tonchev, "A characterization of entanglement-assisted quantum low-density parity-check codes," *IEEE Transactions on Information Theory*, vol. 59, no. 6, pp. 3347–3353, June 2013.

[204] D. Chandra, Z. Babar, H. V. Nguyen, D. Alanis, P. Botsinis, S. X. Ng, and L. Hanzo, "Quantum coding bounds and a closed-form approximation of the minimum distance versus quantum coding rate," *IEEE Access*, vol. 5, pp. 11 557–11 581, 2017.

[205] R. Cleve and D. Gottesman, "Efficient computations of encodings for quantum error correction," *Physical Review A*, vol. 56, no. 1, 1997.

[206] I. Djordjevic, *Quantum Information Processing and Quantum Error Correction: An Engineering Approach*. Academic press, 2012.

[207] S. Lin and D. J. Costello, *Error Control Coding*. Pearson Education India. [Online]. Available: http://books.google.co.uk/books?id=BG08Zylo_7gC

[208] 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, no. 2, pp. 284 – 287, Mar. 1974.

[209] J. Wolf, "Efficient maximum likelihood decoding of linear block codes using a trellis," *IEEE Transactions on Information Theory*, vol. 24, no. 1, pp. 76 – 80, jan 1978.

[210] J. Schalkwijk and A. Vinck, "Syndrome decoding of convolutional codes," *IEEE Transactions on Communications*, vol. 23, no. 7, pp. 789 – 792, jul 1975.

[211] ——, "Syndrome decoding of binary rate-1/2 convolutional codes," *IEEE Transactions on Communications*, vol. 24, no. 9, pp. 977 – 985, sep 1976.

[212] J. Schalkwijk, A. Vinck, and K. Post, "Syndrome decoding of binary-rate k/n convolutional codes," *IEEE Transactions on Information Theory*, vol. 24, no. 5, pp. 553 – 562, sep 1978.

[213] M. Ariel and J. Snyders, "Soft syndrome decoding of binary convolutional codes," *IEEE Transactions on Communications*, vol. 43, no. 234, pp. 288 – 297, Feb./Mar./Apr. 1995.

[214] V. Sidorenko and V. Zyablov, "Decoding of convolutional codes using a syndrome trellis," *IEEE Transactions on Information Theory*, vol. 40, no. 5, pp. 1663 –1666, sep 1994.

[215] 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.

[216] 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.

[217] K. Hueske, J. Geldmacher, and J. Götze, "Adaptive decoding of convolutional codes," *Advances in Radio Science*, vol. 5, pp. 209–214, 2007. [Online]. Available: http://www.adv-radio-sci.net/5/209/2007/

[218] ——, "Multi core implementation of a trellis based syndrome decoder with adaptive complexity," in *International Symposium on Wireless Communication Systems*, sept. 2010, pp. 859 –863.

[219] 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.

[220] 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.

[221] 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*, oct. 2009, pp. 1 –7.

[222] 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*, nov. 2010, pp. 155 –159.

[223] 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*, nov. 2011, pp. 26 –30.

[224] Z. Babar, S. X. Ng, and L. Hanzo, "Reduced-complexity syndrome-based TTCM decoding," *IEEE Communications Letters*, vol. 17, no. 6, pp. 1220–1223, June 2013.

[225] A. Viterbi, "Error bounds for convolutional codes and an asymptotically optimum decoding algorithm," *IEEE Transactions on Information Theory*, vol. 13, no. 2, pp. 260–269, April 1967.

[226] 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, no. 9, pp. 1474 – 1484, sept. 2003.

[227] 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, no. 3, pp. 920 –935, sept. 2011.

[228] M. Tajima, K. Shibata, and Z. Kawasaki, "Modification of syndrome trellises for high-rate convolutional codes," *IEICE Technical Report*, vol. 102, no. 66, pp. 7 – 12, May 2002.

[229] P. Robertson and T. Worz, "Bandwidth-efficient turbo trellis-coded modulation using punctured component codes," *IEEE Journal on Selected Areas in Communications*, vol. 16, no. 2, pp. 206 – 218, Feb. 1998.

[230] G. Ungerboeck, "Channel coding with multilevel/phase signals," *IEEE Transactions on Information Theory*, vol. 28, no. 1, pp. 55 – 67, Jan. 1982.

[231] H. Chen and A. Haimovich, "Exit charts for turbo trellis-coded modulation," *IEEE Communications Letters*, vol. 8, no. 11, pp. 668 – 670, nov. 2004.

[232] 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. [Online]. Available: http://dx.doi.org/%7B10.1103/PhysRevLett.69.2881%7D

[233] 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. [Online]. Available: http://link.aps.org/doi/10.1103/PhysRevLett.83.3081

[234] A. Holevo, "On entanglement-assisted classical capacity," *J. Math. Phys.*, vol. 43, no. 9, pp. 4326–4333, 2002.

[235] 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, no. 12, pp. 4801–4807, December 2013.

[236] ——, "EXIT-chart aided code design for symbol-based entanglement-assisted classical communication over quantum channels," in *IEEE Vehicular Technology Conference (VTC Fall)*, Sept 2014, pp. 1–5.

[237] 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.

[238] 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. [Online]. Available: http://link.aps.org/doi/10.1103/PhysRevA.87.022333

[239] P. Kaye, R. Laflamme, and M. Mosca, *An introduction to quantum computing.* Oxford University Press, 2007. [Online]. Available: http://books.google.com/books?id=gLFQAAAAMAAJ

[240] 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.

[241] 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.

[242] C. E. Shannon, "A Mathematical Theory of Communication," *The Bell system Technical Journal*, vol. 27, no. 3, pp. 379–656, July 1948.

[243] J. Kliewer, S. X. Ng, and L. Hanzo, "Efficient computation of EXIT functions for non-binary iterative decoding," *IEEE Transactions on Communications*, vol. 54, no. 12, pp. 2133–2136, December 2006.

[244] 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, no. 12, pp. 2030 –2039, December 2008.

[245] S. ten Brink, "Convergence behaviour of iteratively decoded parallel concatenated codes," *IEEE Transactions on Communications*, vol. 49, no. 10, pp. 1727–1737, October 2001.

[246] M. El-Hajjar and L. Hanzo, "EXIT charts for system design and analysis," *IEEE Communications Surveys Tutorials*, pp. 1–27, 2013.

[247] M. Tüchler and J. Hagenauer, "EXIT charts of irregular codes," in *Proceedings of Conference on Information Science and Systems*, Princeton University, 20-22 March 2002, pp. 465–490.

[248] M. Tüchler, "Design of serially concatenated systems depending on the block length," *IEEE Transactions on Communications*, vol. 52, no. 2, pp. 209–218, February 2004.

[249] 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, no. 4, pp. 686–698, April 2001.

[250] A. Ashikhmin, G. Kramer, and S. ten Brink, "Extrinsic information transfer functions: model and erasure channel properties," *IEEE Transactions on Information Theory*, vol. 50, no. 11, pp. 2657–2673, November 2004.

[251] C. Weidmann and G. Lechner, "A fresh look at coding for q-ary symmetric channels," *IEEE Transactions on Information Theory*, vol. 58, no. 11, pp. 6959–6967, Nov 2012.

[252] R. C. Singleton, "Maximum distance Q-nary codes," *IEEE Transactions on Information Theory*, vol. 10, no. 2, pp. 116–118, 1964.

[253] R. W. Hamming, "Error detecting and error correcting codes," *Bell System Technical Journal*, vol. 29, no. 2, pp. 147–160, 1950.

[254] E. N. Gilbert, "A comparison of signalling alphabets," *Bell System Technical Journal*, vol. 31, no. 3, pp. 504–522, 1952.

[255] R. J. McEliece, E. R. Rodemich, H. Rumsey Jr, and L. R. Welch, "New upper bounds on the rate of a code via the Delsarte-MacWilliams inequalities," *IEEE Transactions on Information Theory*, vol. 23, no. 2, pp. 157–166, 1977.

[256] M. Plotkin, "Binary codes with specified minimum distance," *IRE Transactions on Information Theory*, vol. 6, no. 4, pp. 445–450, 1960.

[257] E. M. Rains, "Quantum shadow enumerators," *IEEE Transactions on Information Theory*, vol. 45, no. 7, pp. 2361–2366, 1999.

[258] A. Ashikhmin and S. Litsyu, "Upper bounds on the size of quantum codes," *IEEE Transactions on Information Theory*, vol. 45, no. 4, pp. 1206–1215, 1999.

[259] C.-Y. Lai and C.-C. Lu, "A construction of quantum stabilizer codes based on syndrome assignment by classical parity-check matrices," *IEEE Transactions on Information Theory*, vol. 57, no. 10, pp. 7163–7179, 2011.

[260] Y. Xie, J. Yuan, *et al.*, "Design of quantum stabilizer codes from quadratic residues sets," *arXiv preprint arXiv:1407.8249*, 2014.

[261] M. Grassl, "Bounds on the minimum distance of linear codes and quantum codes," Online available at http://www.codetables.de, 2007, accessed on 2016-05-14.

[262] C.-Y. Lai, "Linear programming bounds for entanglement-assisted quantum error-correcting codes by split weight enumerators," *arXiv preprint arXiv:1602.00413*, 2016.

[263] L. Guo and R. Li, "Linear Plotkin bound for entanglement-assisted quantum codes," *Physical Review A*, vol. 87, no. 3, 2013.

[264] C.-Y. Lai, T. A. Brun, and M. M. Wilde, "Dualities and identities for entanglement-assisted quantum codes," *Quantum Information P rocessing*, vol. 13, no. 4, pp. 957–990, 2014.

[265] L. Hanzo, T. Liew, B. Yeap, R. Tee, and S. X. Ng, *Turbo coding, turbo equalisation and space-time coding: EXIT-chart-aided near-capacity designs for wireless channels*. IEEE Press-John Wiley, 2011.

[266] H. Bombin and M. A. Martin-Delgado, "Homological error correction: Classical and quantum codes," *Journal of Mathematical Physics*, vol. 48, no. 5, 2007.

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

[268] F. R. Kschischang, B. J. Frey, and H.-A. Loeliger, "Factor graphs and the sum-product algorithm," *IEEE Transactions on Information Theory*, vol. 47, no. 2, pp. 498–519, 2001.

[269] R. Steele and L. Hanzo, *Mobile Radio Communications: Second and Third Generation Cellular and WATM Systems*. IEEE Press-John Wiley, 1999.

[270] C. H. Bennett, D. P. DiVincenzo, J. A. Smolin, and W. K. Wootters, "Mixed-state entanglement and quantum error correction," *Physical Review A*, vol. 54, no. 5, 1996.

[271] R. Jozsa, "Fidelity for mixed quantum states," *Journal of Modern Optics*, vol. 41, no. 12, pp. 2315–2323, 1994.

[272] B. Schumacher, "Quantum coding," *Physical Review A*, vol. 51, no. 4, 1995.

[273] J. Von Neumann, "Probabilistic logics and the synthesis of reliable organisms from unreliable components," *Automata Studies*, vol. 34, pp. 43–98, 1956.

[274] A. M. Steane, "Space, time, parallelism and noise requirements for reliable quantum computing," *Fortschritte der Physik*, vol. 46, no. 4-5, pp. 443–457, 1998.

[275] D. E. Deutsch, "Quantum computational networks," *Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, vol. 425, no. 1868, pp. 73–90, 1989.

[276] R. Raussendorf, D. E. Browne, and H. J. Briegel, "Measurement-based quantum computation on cluster states," *Physical Review A*, vol. 68, no. 2, 2003.

[277] G. Grössing and A. Zeilinger, "Quantum cellular automata," *Complex Systems*, vol. 2, no. 2, pp. 197–208, 1988.

[278] M. H. Freedman, A. Y. Kitaev, M. Larsen, and Z. Wang, "Topological quantum computation," *Bulletin of the American Mathematical Society*, vol. 40, no. 1, pp. 31–38, 2003.

[279] E. Farhi, J. Goldstone, S. Gutmann, and M. Sipser, "Quantum computation by adiabatic evolution," *arXiv preprint quant-ph/0001106*, 2000.

[280] M. Grassl, "Variations on encoding circuits for stabilizer quantum codes," in *Proceedings of International Conference on Coding and Cryptology*. Springer, 2011, pp. 142–158.

[281] S. J. Devitt, W. J. Munro, and K. Nemoto, "Quantum error correction for beginners," *Reports on Progress in Physics*, vol. 76, no. 7, 2013.

[282] J. Łodyga, P. Mazurek, A. Grudka, and M. Horodecki, "Simple scheme for encoding and decoding a qubit in unknown state for various topological codes," *Scientific Reports*, vol. 5, 2015.

[283] D. Chandra, Z. Babar, H. V. Nguyen, D. Alanis, P. Botsinis, S. X. Ng, and L. Hanzo, "Quantum topological error correction codes are capable of improving the performance of Clifford gates," *IEEE Access*, vol. 7, pp. 121 501–121 529, 2019.

[284] A. M. Steane, "Overhead and noise threshold of fault-tolerant quantum error correction," *Physical Review A*, vol. 68, no. 4, 2003.

[285] H. Bombin and M. A. Martin-Delgado, "Quantum measurements and gates by code deformation," *Journal of Physics A: Mathematical and Theoretical*, vol. 42, no. 9, 2009.

[286] H. Bombin, "Clifford gates by code deformation," *New Journal of Physics*, vol. 13, no. 4, 2011.

[287] J. T. Anderson, G. Duclos-Cianci, and D. Poulin, "Fault-tolerant conversion between the Steane and Reed-Muller quantum codes," *Physical review letters*, vol. 113, no. 8, 2014.

[288] C. Vuillot, L. Lao, B. Criger, C. G. Almudever, K. Bertels, and B. Terhal, "Code deformation and lattice surgery are gauge fixing," *New Journal of Physics*, vol. 21, no. 3, 2019.

[289] A. G. Fowler, A. M. Stephens, and P. Groszkowski, "High-threshold universal quantum computation on the surface code," *Physical Review A*, vol. 80, no. 5, 2009.

[290] D. S. Wang, A. G. Fowler, and L. C. Hollenberg, "Surface code quantum computing with error rates over 1%," *Physical Review A*, vol. 83, no. 2, 2011.

[291] A. G. Fowler, M. Mariantoni, J. M. Martinis, and A. N. Cleland, "Surface codes: Towards practical large-scale quantum computation," *Physical Review A*, vol. 86, no. 3, 2012.

[292] T. A. Brun, Y.-C. Zheng, K.-C. Hsu, J. Job, and C.-Y. Lai, "Teleportation-based fault-tolerant quantum computation in multi-qubit large block codes," *arXiv preprint arXiv:1504.03913*, 2015.

[293] A. J. Landahl, J. T. Anderson, and P. R. Rice, "Fault-tolerant quantum computing with color codes," *arXiv preprint arXiv:1108.5738*, 2011.

[294] F. J. MacWilliams and N. J. A. Sloane, *The theory of error-correcting codes*. Elsevier, 1977.

[295] J. I. Hall, *Notes on coding theory*. FreeTechBooks, 2003.

[296] M. M. Christiansen, K. R. Duffy, F. du Pin Calmon, and M. Médard, "Multi-user guesswork and brute force security," *IEEE Transactions on Information Theory*, vol. 61, no. 12, pp. 6876–6886, 2015.

[297] M. M. Christiansen and K. R. Duffy, "Guesswork, large deviations, and Shannon entropy," *IEEE Transactions on Information Theory*, vol. 59, no. 2, pp. 796–802, 2012.

[298] K. R. Duffy, J. Li, and M. Médard, "Capacity-achieving guessing random additive noise decoding," *IEEE Transactions on Information Theory*, vol. 65, no. 7, pp. 4023–4040, 2019.

[299] K. R. Duffy, M. Médard, and W. An, "Guessing random additive noise decoding with symbol reliability information (SRGRAND)," *IEEE Transactions on Communications*, vol. 70, no. 1, pp. 3–18, 2021.

[300] K. R. Duffy, E. P. Gabhart, and M. Médard, "Quantized guessing random additive noise decoding," *arXiv preprint arXiv:2203.13552*, 2022.

[301] A. Rényi, "On measures of entropy and information," in *Proceedings of the Fourth Berkeley Symposium on Mathematical Statistics and Probability: Contributions to the Theory of Statistics*, vol. 4. University of California Press, 1961, pp. 547–562.

[302] A. Riaz, V. Bansal, A. Solomon, W. An, Q. Liu, K. Galligan, K. R. Duffy, M. Medard, and R. T. Yazicigil, "Multi-code multi-rate universal maximum likelihood decoder using GRAND," in *Proceedings of IEEE 47th European Solid-State Circuits Conference (ESSCIRC)*. IEEE, 2021, pp. 239–246.

[303] D. Chandra, Z. B. K. Egilmez, Y. Xiong, S. X. Ng, R. G. Maunder, and L. Hanzo, "Universal decoding of quantum stabilizer codes via classical guesswork," *IEEE Access*, vol. 11, pp. 19 059–19 072, 2023.

[304] A. M. Steane, "Simple quantum error-correcting codes," *Physical Review A*, vol. 54, no. 6, 1996.

[305] M. Grassl and T. Beth, "Quantum BCH codes," *arXiv preprint quant-ph/9910060*, 1999.

[306] Z. Babar, Z. B. K. Egilmez, L. Xiang, D. Chandra, R. G. Maunder, S. X. Ng, and L. Hanzo, "Polar codes and their quantum-domain counterparts," *IEEE Communications Surveys & Tutorials*, vol. 22, no. 1, pp. 123–155, 2019.

[307] D. Chandra, Z. Babar, S. X. Ng, and L. Hanzo, "Near-hashing-bound multiple-rate quantum turbo short-block codes," *IEEE Access*, vol. 7, pp. 52 712–52 730, 2019.

[308] A. Solomon, K. R. Duffy, and M. Médard, "Soft maximum likelihood decoding using GRAND," in *Proceedings of IEEE International Conference on Communications (ICC)*. IEEE, 2020, pp. 1–6.

[309] K. R. Duffy, W. An, and M. Médard, "Ordered reliability bits guessing random additive noise decoding," *IEEE Transactions on Signal Processing*, vol. 70, pp. 4528–4542, 2022.

[310] D. M. Nguyen and S. Kim, "Quantum stabilizer codes construction from Hermitian self-orthogonal codes over GF (4)," *Journal of Communications and Networks*, vol. 20, no. 3, pp. 309–315, 2018.

[311] Z. Yi, Z. Liang, and X. Wang, "Quantum polar stabilizer codes based on polarization of pure quantum channel are bad stabilizer codes for quantum computing," *arXiv preprint arXiv:2204.11655*, 2022.

[312] W. K. Wootters and W. H. Zurek, "A single quantum cannot be cloned," *Nature*, vol. 299, no. 5886, pp. 802–803, Oct. 1982. [Online]. Available: http://dx.doi.org/10.1038/299802a0

[313] J. M. Renes, F. Dupuis, and R. Renner, "Efficient polar coding of quantum information," *Phys. Rev. Lett.*, vol. 109, p. 050504, Aug 2012. [Online]. Available: http://link.aps.org/doi/10.1103/PhysRevLett.109.050504

[314] D. Gottesman, "Class of quantum error-correcting codes saturating the quantum Hamming bound," *Phys. Rev. A*, vol. 54, no. 3, pp. 1862–1868, Sep 1996.

[315] ——, "Stabilizer codes and quantum error correction," Ph.D. dissertation, California Institute of Technology, 1997.

[316] 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)*.

[317] ——, "The road from classical to quantum codes: A hashing bound approaching design procedure," *IEEE Access*, vol. 3, pp. 146–176, 2015.

[318] R. Cleve, "Quantum stabilizer codes and classical linear codes," *Phys. Rev. A*, vol. 55, pp. 4054–4059, Jun 1997. [Online]. Available: http://link.aps.org/doi/10.1103/PhysRevA.55.4054

[319] A. M. Steane, "Error correcting codes in quantum theory," *Phys. Rev. Lett.*, vol. 77, no. 5, pp. 793–797, Jul 1996.

[320] 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, no. 4, pp. 1369–1387, Jul 1998.

[321] S. Lloyd, "Capacity of the noisy quantum channel," *Phys. Rev. A*, vol. 55, pp. 1613–1622, Mar 1997. [Online]. Available: http://link.aps.org/doi/10.1103/PhysRevA.55.1613

[322] P. W. Shor, "The quantum channel capacity and coherent information," *Lecture Notes, MSRI Workshop on Quantum Computation*, 2002.

[323] I. Devetak, "The private classical capacity and quantum capacity of a quantum channel," *IEEE Transactions on Information Theory*, vol. 51, no. 1, pp. 44–55, Jan 2005.

[324] 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, May 1993, pp. 1064–1070 vol.2.

[325] Z. Babar, S. X. Ng, and L. Hanzo, "EXIT-chart-aided near-capacity quantum turbo code design," *IEEE Transactions on Vehicular Technology*, vol. 64, no. 3, pp. 866–875, March 2015.

[326] 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. [Online]. Available: http://link.aps.org/doi/10.1103/PhysRevA.57.830

[327] G. Smith and J. A. Smolin, "Degenerate quantum codes for pauli channels," *Phys. Rev. Lett.*, vol. 98, p. 030501, Jan 2007. [Online]. Available: http://link.aps.org/doi/10.1103/PhysRevLett.98.030501

[328] 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. [Online]. Available: http://link.aps.org/doi/10.1103/PhysRevA.54.3824

[329] 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. [Online]. Available: http://dx.doi.org/10.1007/s11128-013-0704-8

[330] 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. [Online]. Available: http://link.aps.org/doi/10.1103/PhysRevA.68.042318

[331] D. Gottesman, *The Heisenberg Representation of Quantum Computers.* [online] http://arxiv.org/pdf/quant-ph/9807006v1.pdf, December 2001.

[332] M. El-Hajjar and L. Hanzo, "EXIT charts for system design and analysis," *IEEE Communications Surveys & Tutorials*, vol. 16, no. 1, pp. 127–153, 2014.

[333] 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.

[334] 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.

[335] 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.

[336] A. Grant, "Convergence of non-binary iterative decoding," in *IEEE Global Telecommunications Conference*, vol. 2, 2001, pp. 1058–1062 vol.2.

[337] M. M. Wilde, *Quantum Information Theory*. Cambridge University Press, May 2013. [Online]. Available: http://arxiv.org/abs/1106.1445

[338] D. P. Divincenzo, D. Leung, and B. Terhal, "Quantum data hiding," *IEEE Transactions on Information Theory*, vol. 48, no. 3, pp. 580–598, Mar 2002.

[339] E. Rains, "Nonbinary quantum codes," *IEEE Transactions on Information Theory*, vol. 45, no. 6, pp. 1827–1832, 1999.

[340] M. Abbara and J.-P. Tillich, "Quantum turbo codes with unbounded minimum distance and excellent error-reducing performance," in *Proceedings of IEEE Information Theory Workshop (ITW)*, 2011, pp. 252–256.

[341] Z. Babar, H. V. Nguyen, P. Botsinis, D. Alanis, D. Chandra, S. X. Ng, and L. Hanzo, "Serially concatenated unity-rate codes improve quantum codes without coding-rate reduction," *IEEE Communications Letters*, vol. 20, no. 10, pp. 1916–1919, 2016.

[342] S. Benedetto, D. Divsalar, G. Montorsi, and F. Pollara, "Serial concatenation of interleaved codes: Performance analysis, design, and iterative decoding," *IEEE Transactions on Information Theory*, vol. 44, no. 3, pp. 909–926, 1998.

[343] I. Land, "Reliability information in channel decoding," Ph.D. dissertation, Christian-Albrechts-University of Kiel, 2005.

[344] L. Hanzo and Nasruminallah, "Short block codes for guaranteed convergence in soft-bit assisted iterative joint source and channel decoding," *Electronics Letters*, vol. 44, no. 22, pp. 1315–1316, 2008.

[345] ——, "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, no. 8, pp. 4306–4315, 2009.

[346] J. Hagenauer, "Rate-compatible punctured convolutional codes (RCPC codes) and their applications," *IEEE Transactions on Communications*, vol. 36, no. 4, pp. 389–400, 1988.

[347] Z. Babar, S. X. Ng, and L. Hanzo, "EXIT-chart-aided near-capacity quantum turbo code design," *IEEE Transactions on Vehicular Technology*, vol. 64, no. 3, pp. 866–875, 2015.

[348] P. K. Sarvepalli, A. Klappenecker, and M. Rötteler, "Asymmetric quantum codes: constructions, bounds and performance," *Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, vol. 465, no. 2105, pp. 1645–1672, 2009.

[349] L. Wang, K. Feng, S. Ling, and C. Xing, "Asymmetric quantum codes: Characterization and constructions," *IEEE Transactions on Information Theory*, vol. 56, no. 6, pp. 2938–2945, 2010.

[350] S. Ten Brink, "Convergence behavior of iteratively decoded parallel concatenated codes," *IEEE Transactions on Communications*, vol. 49, no. 10, pp. 1727–1737, 2001.

[351] J. Kliewer, S. X. Ng, and L. Hanzo, "Efficient computation of EXIT functions for nonbinary iterative decoding," *IEEE Transactions on Communications*, vol. 54, no. 12, pp. 2133–2136, 2006.

[352] 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.

[353] A. Banerjee, F. Vatta, B. Scanavino, and D. J. Costello, "Nonsystematic turbo codes," *IEEE Transactions on Communications*, vol. 53, no. 11, pp. 1841–1849, 2005.

[354] T. A. Summers and S. G. Wilson, "SNR mismatch and on-line estimation in turbo decoding," *IEEE Transactions on Communications*, vol. 46, pp. 421–423, 1998.

[355] A. Worm, P. Hoeher, and N. Wehn, "Turbo-decoding without SNR estimation," *IEEE Communications Letters*, vol. 4, no. 6, pp. 193–195, 2000.

[356] M. A. Khalighi, "Effect of mismatched SNR on the performance of log-MAP turbo detector," *IEEE Transactions on Vehicular Technology*, vol. 52, no. 5, pp. 1386–1397, 2003.

[357] Y. Xie, J. Li, R. Malaney, and J. Yuan, "Channel identification and its impact on quantum LDPC code performance," in *Proceedings of the Australian Communications Theory Workshop (AusCTW)*, 2012, pp. 140–144.

[358] ——, "Improved quantum LDPC decoding strategies for the misidentified quantum depolarization channel," in *Proceedings of the 24th European Signal Processing Conference (EUSIPCO)*, 2016, pp. 493–497.

[359] Y. Fujiwara, "Instantaneous quantum channel estimation during quantum information processing," *arXiv preprint arXiv:1405.6267*, 2014.

[360] T. C. Bschorr, D. G. Fischer, and M. Freyberger, "Channel estimation with noisy entanglement," *Physics Letters A*, vol. 292, no. 1, pp. 15–22, 2001.

[361] D. Collins and J. Stephens, "Depolarizing-channel parameter estimation using noisy initial states," *Physical Review A*, vol. 92, no. 3, 2015.

[362] E. Dumitrescu and T. S. Humble, "Direct characterization of quantum dynamics with noisy ancilla," *Physical Review A*, vol. 92, no. 5, 2015.

[363] V. Toto-Zarasoa, A. Roumy, and C. Guillemot, "Maximum likelihood BSC parameter estimation for the Slepian-Wolf problem," *IEEE Communications Letters*, vol. 15, no. 2, pp. 232–234, 2011.

[364] G. Lechner and C. Pacher, "Estimating channel parameters from the syndrome of a linear code," *IEEE Communications Letters*, vol. 17, no. 11, pp. 2148–2151, 2013.

[365] J. Combes, C. Ferrie, C. Cesare, M. Tiersch, G. Milburn, H. J. Briegel, and C. M. Caves, "In-situ characterization of quantum devices with error correction," *arXiv preprint arXiv:1405.5656*, 2014.

[366] G. Cariolaro and G. Pierobon, "Performance of quantum data transmission systems in the presence of thermal noise," *IEEE Transactions on Communications*, vol. 58, no. 2, pp. 623–630, 2010.

[367] A. S. Cacciapuoti, M. Caleffi, R. Van Meter, and L. Hanzo, "When entanglement meets classical communications: Quantum teleportation for the Quantum Internet," *IEEE Transactions on Communications (Invited Paper)*, vol. 68, no. 6, pp. 3808–3833, 2020.

[368] S. Shao, P. Hailes, T.-Y. Wang, J.-Y. Wu, R. G. Maunder, B. M. Al-Hashimi, and L. Hanzo, "Survey of turbo, LDPC, and polar decoder ASIC implementations," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 3, pp. 2309–2333, 2019.

[369] Z. B. K. Egilmez, L. Xiang, R. G. Maunder, and L. Hanzo, "The development, operation and performance of the 5G polar codes," *IEEE Communications Surveys & Tutorials*, vol. 22, no. 1, pp. 96–122, 2019.

[370] V. Bioglio, C. Condo, and I. Land, "Design of polar codes in 5G new radio," *IEEE Communications Surveys & Tutorials*, vol. 23, no. 1, pp. 29–40, 2020.

[371] Y. Fang, Y. Bu, P. Chen, F. C. Lau, and S. Al Otaibi, "Irregular-mapped protograph LDPC-coded modulation: A bandwidth-efficient solution for 6G-enabled mobile networks," *IEEE Transactions on Intelligent Transportation Systems*, vol. Early Access, pp. 1–14, 2021.

[372] E. M. Rains, "Nonbinary quantum codes," *IEEE Transactions on Information Theory*, vol. 45, no. 6, pp. 1827–1832, 1999.

[373] F. R. F. Pereira, G. G. La Guardia, and F. M. de Assis, "Classical and quantum convolutional codes derived from algebraic geometry codes," *IEEE Transactions on Communications*, vol. 67, no. 1, pp. 73–82, 2018.

[374] Z. Babar, S. X. Ng, and L. Hanzo, "EXIT-chart-aided near-capacity quantum turbo code design," *IEEE Transactions on Vehicular Technology*, vol. 64, no. 3, pp. 866–875, 2014.

[375] S. Ten Brink, "Convergence of iterative decoding," *Electronics Letters*, vol. 35, no. 10, pp. 806–808, 1999.

[376] M. Tüchler, "EXIT charts of irregular codes," in *Proceedings of the Conference on Information Sciences and Systems*, 2002, pp. 748–753.

[377] ——, "Design of serially concatenated systems depending on the block length," *IEEE Transactions on Communications*, vol. 52, no. 2, pp. 209–218, 2004.

[378] M. El-Hajjar and L. Hanzo, "EXIT charts for system design and analysis," *IEEE Communications Surveys & Tutorials*, vol. 16, no. 1, pp. 127–153, 2013.

[379] D. Chandra, Z. Babar, H. V. Nguyen, D. Alanis, P. Botsinis, S. X. Ng, and L. Hanzo, "Quantum coding bounds and a closed-form approximation of the minimum distance versus quantum coding rate," *IEEE Access*, vol. 5, pp. 11 557–11 581, 2017.

[380] ——, "Quantum topological error correction codes: The classical-to-quantum isomorphism perspective," *IEEE Access*, vol. 6, pp. 13 729–13 757, 2018.

[381] D. Chandra, S. X. Ng, and L. Hanzo, "EXIT-chart aided design of irregular multiple-rate quantum turbo block codes," *IEEE Access*, vol. 11, pp. 96 177–96 195, 2023.

[382] J. Etxezarreta Martinez, P. M. Crespo, and J. Garcia-Frías, "On the performance of interleavers for quantum turbo codes," *Entropy*, vol. 21, no. 7, pp. 1–9, 2019.

[383] J. E. Martinez, P. Fuentes, P. M. Crespo, and J. Garcia-Frías, "Pauli channel online estimation protocol for quantum turbo codes," in *Proceedings of IEEE International Conference on Quantum Computing and Engineering (QCE)*. IEEE, 2020, pp. 102–108.

[384] J. E. Martinez, P. M. Crespo, and J. Garcia-Frías, "Depolarizing channel mismatch and estimation protocols for quantum turbo codes," *Entropy*, vol. 21, no. 12, pp. 1133–1147, 2019.

[385] Y.-J. Wang, Z.-Y. Xiao, Y. Zhang, X.-Y. Xiong, and S. Shi, "Construction of multiple-rate quantum LDPC codes sharing one scalable stabilizer circuit," *IEEE Transactions on Communications*, vol. 71, no. 2, pp. 1071–1082, 2023.

[386] S. Yang and R. Calderbank, "Quantum spatially-coupled codes," *arXiv preprint arXiv:2305.00137*, 2023.

[387] X.-J. Li, D. Pan, G.-L. Long, and L. Hanzo, "Single-photon-memory measurement-device-independent quantum secure direct communication — Part I: Its fundamentals and evolution," *IEEE Communications Letters*, vol. 27, no. 4, pp. 1055–1059, 2023.

[388] ——, "Single-photon-memory measurement-device-independent quantum secure direct communication — Part II: A practical protocol and its secrecy capacity," *IEEE Communications Letters*, vol. 27, no. 4, pp. 1060–1064, 2023.

[389] D. Chandra, A. S. Cacciapuoti, M. Caleffi, and L. Hanzo, "Direct quantum communications in the presence of realistic noisy entanglement," *IEEE Transactions on Communications*, vol. 70, no. 1, pp. 469–484, 2021.

[390] D. Cruz, F. A. Monteiro, and B. C. Coutinho, "Quantum error correction via noise guessing decoding," *arXiv preprint arXiv:2208.02744*, 2022.

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

[392] 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, no. 2, pp. 58–60, Feb 2001.

[393] N. Bonello, S. Chen, and L. Hanzo, "Low-density parity-check codes and their rateless relatives," *IEEE Communications Surveys Tutorials*, vol. 13, no. 1, pp. 3–26, First 2011.

[394] ——, "Design of low-density parity-check codes," *IEEE Vehicular Technology Magazine*, vol. 6, no. 4, pp. 16–23, Dec 2011.

[395] 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)*.

[396] H. Wymeersch, F. Penna, and V. Savic, "Uniformly reweighted belief propagation: A factor graph approach," in *IEEE International Symposium on Information Theory Proceedings*, July 2011, pp. 2000–2004.

[397] ——, "Uniformly reweighted belief propagation for estimation and detection in wireless networks," *IEEE Transactions on Wireless Communications*, vol. 11, no. 4, pp. 1587–1595, April 2012.

[398] 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, no. 7, pp. 2711–2736, Nov 2001.

[399] B. Vasic and O. Milenkovic, "Combinatorial constructions of low-density parity-check codes for iterative decoding," *IEEE Transactions on Information Theory*, vol. 50, no. 6, pp. 1156 – 1176, june 2004.

[400] 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, no. 6, pp. 1257 – 1269, june 2004.

[401] A. Kitaev, "Fault-tolerant quantum computation by anyons," *Annals of Physics*, vol. 303, no. 1, pp. 2 – 30, 2003. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0003491602000180

[402] H. Bombin and M. A. Martin-Delgado, "Homological error correction: Classical and quantum codes," *Journal of Mathematical Physics*, vol. 48, no. 5, pp. –, 2007. [Online]. Available: http://scitation.aip.org/content/aip/journal/jmp/48/5/10.1063/1.2731356

[403] H. Bombin and M. Martin-Delgado, "Topological quantum distillation," *Phys. Rev. Lett.*, vol. 97, p. 180501, Oct 2006. [Online]. Available: http://link.aps.org/doi/10.1103/PhysRevLett.97.180501

[404] S. ten Brink, "Code doping for triggering iterative decoding convergence," in *IEEE International Symposium on Information Theory*, 2001, pp. 235–.

[405] M. Hagiwara and H. Imai, "Quantum quasi-cyclic LDPC codes," in *arXiv:quant-ph/0701020v4*, Aug 2010.

[406] I. Andriyanova, D. Maurice, and J.-P. Tillich, "Spatially coupled quantum ldpc codes," in *IEEE Information Theory Workshop*, Sept 2012, pp. 327–331.

[407] 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*, July 2013, pp. 907–911.

[408] T. Camara, H. Ollivier, and J.-P. Tillich, "Constructions and performance of classes of quantum LDPC codes," *arXiv:quant-ph/0502086v2*, 2005.

[409] 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.

[410] ——, "High performance entanglement-assisted quantum LDPC codes need little entanglement," *Information Theory, IEEE Transactions on*, vol. 57, no. 3, pp. 1761 –1769, march 2011.

[411] I. B. Djordjevic, "Photonic entanglement-assisted quantum low-density parity-check encoders and decoders," *Opt. Lett.*, vol. 35, no. 9, pp. 1464 – 1466, May 2010. [Online]. Available: http://ol.osa.org/abstract.cfm?URI=ol-35-9-1464

[412] 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. [Online]. Available: http://link.aps.org/doi/10.1103/PhysRevA.82.042338

[413] 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, no. 4, pp. 1860–1878, April 2015.

[414] A. Steane, "Multiple-particle interference and quantum error correction," *Royal Society of London Proceedings Series A*, vol. 452, pp. 2551–2577, Nov. 1995.

[415] T. Richardson, M. Shokrollahi, and R. Urbanke, "Design of capacity-approaching irregular low-density parity-check codes," *IEEE Transactions on Information Theory*, vol. 47, no. 2, pp. 619–637, Feb 2001.

[416] Y. Fujiwara, "Quantum error correction via less noisy qubits," *Phys. Rev. Lett.*, vol. 110, p. 170501, Apr 2013. [Online]. Available: http://link.aps.org/doi/10.1103/PhysRevLett.110.170501

[417] D. J. MacKay, *Information Theory, Inference, and Learning Algorithms*. Cambridge University Press, 2003.

[418] E. Berlekamp, R. McEliece, and H. Van Tilborg, "On the inherent intractability of certain coding problems (corresp.)," *IEEE Transactions on Information Theory*, vol. 24, no. 3, pp. 384–386, May 1978.

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

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

[421] ——, "A family of irregular LDPC codes with low encoding complexity," *IEEE Communications Letters*, vol. 7, no. 2, pp. 79–81, Feb 2003.

[422] 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, no. 8, pp. 3108–3113, August 2013.

[423] R. Bose, "On the construction of balanced incomplete block designs," *Ann. Eugen*, vol. 9, pp. 353 – 399, 1939.

[424] W. Ryan and S. Lin, *Channel Codes: Classical and Modern*. Cambridge University Press, 2009. [Online]. Available: http://books.google.co.uk/books?id=n9BNngEACAAJ

[425] 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, no. 7, pp. 2313–2335, July 2005.

[426] T. Roosta, M. Wainwright, and S. Sastry, "Convergence analysis of reweighted sum-product algorithms," *IEEE Transactions on Signal Processing*, vol. 56, no. 9, pp. 4293–4305, Sept 2008.

[427] 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*, Aug 2012, pp. 984–988.

[428] ——, "Low-latency reweighted belief propagation decoding for LDPC codes," *IEEE Communications Letters*, vol. 16, no. 10, pp. 1660–1663, October 2012.

[429] 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*, April 2013, pp. 4311–4316.

[430] H. Wymeersch, F. Penna, V. Savic, and J. Zhao, "Comparison of reweighted message passing algorithms for LDPC decoding," in *IEEE International Conference on Communications*, June 2013, pp. 3264–3269.

[431] 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. [Online]. Available: http://link.aps.org/doi/10.1103/PhysRevA.78.012341

[432] M.-H. Hsieh and M. M. Wilde, "Entanglement-assisted communication of classical and quantum information," *IEEE Transactions on Information Theory*, vol. 56, no. 9, pp. 4682 –4704, sept. 2010.

[433] ——, "Trading classical communication, quantum communication, and entanglement in quantum shannon theory," *IEEE Transactions on Information Theory*, vol. 56, no. 9, pp. 4705–4730, Sept 2010.

[434] Nasruminallah and L. Hanzo, "Short block codes for guaranteed convergence in soft-bit assisted iterative joint source and channel decoding," *Electronics Letters*, vol. 44, no. 22, pp. 1315–1316, October 2008.

[435] ——, "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, no. 8, pp. 4306–4315, Oct 2009.

Chapter 5

Near-Capacity Codes for Entanglement-Aided Classical Communication

5.1 Introduction

Throughout the previous three chapters we have laid the foundations for the more deep-routed discussions of this chapter inserted here for providing a more fast-paced portrayal of near-capacity code design principles in the classical domain first. In Section 5.5 we will highlight the classical capacity improvements that may be attained with the aid of entanglement and discuss the maximum code-rate capable of achieving an infinitesimally low BER at a given depolarizing probability. Then in Section 5.5 we will delve into sophisticated EXIT-chart-based near-capacity designs relying on so-called irregular convolutional codes and concatenated unity-rate codes iteratively exchanging extrinsic information. These powerful design tools will be heavily relied upon in the sophisticated frontier-research of Paft III of the book, but they are first introduced from a classical-domain perspective.

Quantum-domain communication constitutes an attractive solution for absolute secure transmission [199]. More explicitly, any 'measurement' or 'observation' of the transmitted qubits by the eavesdropper perturbs the associated quantum superposition, hence intimating the parties concerned [199]. In this context, entanglement-assisted transmission of classical information over quantum channels is of particular significance. This idea was conceived by Bennett [232] 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 [233, 234].

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

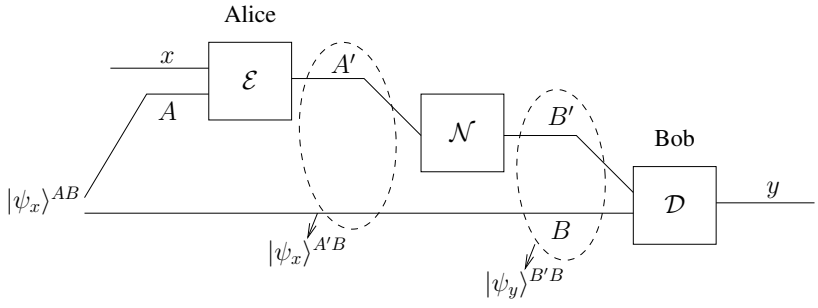


Figure 5.1: Classical-quantum-classical transmission model employing 2-qubit SD. ©Babar *et al.* [235, 236]

is transmitted over a quantum channel with the aid of the SuperDense (SD) coding protocol, is depicted in Figure 5.1. 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, 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 5.1 into the corresponding quantum state using the 2SD coding protocol of [232]. 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 5.1 Bob's receiver performs symbol-by-symbol Bell-basis measurement[1] [199, 4] 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 [237], whereby the 2SD scheme of [232] 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.* [238] 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 [235, 236]:

- 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.
- 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.

Again, the rest of the chapter is laid out as follows. We review the SD coding protocol in Section 5.2, while the EACC of N-qubit SD schemes is investigated in Section 5.3. The bit-based system model and our near-capacity design are detailed in Sections 5.4 and 5.5, respectively, while the corresponding simulation results

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

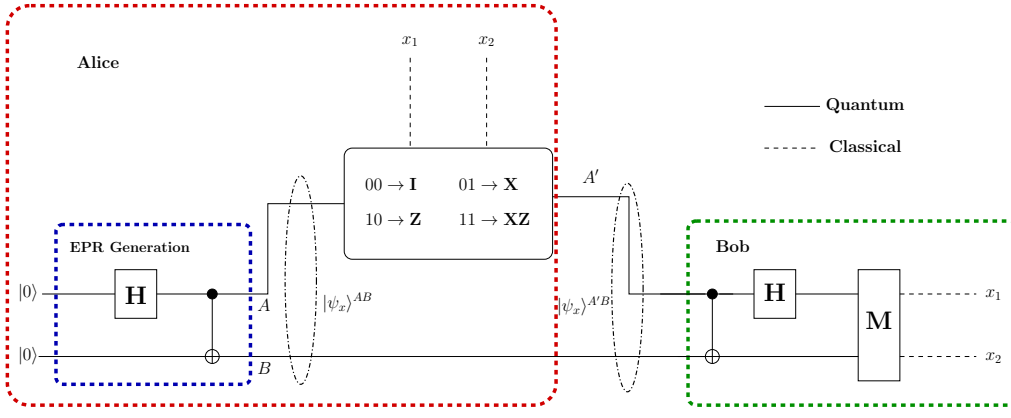


Figure 5.2: The quantum circuit of 2-qubit superdense coding. Alice generates the Einstein-Podolsky-Rosen (EPR) pair $|\psi_x\rangle^{AB}$ using a **H** gate and CNOT gate, respectively. Qubit A is used for encoding the 2-bit message, while the qubit B is pre-shared with Bob. Bob performs Bell-basis measurement on the received state $|\psi_x\rangle^{A'B}$, yielding the original classical message.

are discussed in Section 5.6. Next we have detailed our symbol-based scheme in Section 5.7 and the associated results are discussed in Section 5.8. Finally, our conclusions are offered in Section 5.9.

5.2 Review of the Superdense Coding Protocol

5.2.1 2-Qubit Superdense Coding

The 2SD protocol [232] invokes the peculiar law of quantum entanglement for transmitting two classical bits using a single qubit. Figure 5.2 shows the quantum circuit of 2SD [232]. 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 [232], which is given by [232]:

$$|\psi_x\rangle^{AB} = \frac{|00\rangle + |11\rangle}{\sqrt{2}}. \tag{5.1}$$

This is achieved by applying the Hadamard gate (**H**) of Section 2.3.1.2 to two qubits initialized to the state $|0\rangle$, as depicted in the 'EPR Generation' block of Figure 5.2. More explicitly, the Einstein-Podolsky-Rosen pair generation proceeds as follows:

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

$$|00\rangle \rightarrow \frac{1}{\sqrt{2}}\left(|0\rangle + |1\rangle\right)|0\rangle\,. \tag{5.2}$$

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

Table 5.1: Classical-to-quantum mapping for 2-qubit superdense coding. *(The normalization factor $\frac{1}{\sqrt{2}}$ is ignored for simplicity.)*

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

$$\frac{1}{\sqrt{2}}\left(|00\rangle + |10\rangle\right) \rightarrow \frac{1}{\sqrt{2}}\left(|00\rangle + |11\rangle\right) \equiv |\psi_x\rangle^{AB}. \tag{5.3}$$

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 5.1 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 Section 2.3.1.1 to A,

$$\frac{1}{\sqrt{2}}\left(|00\rangle + |11\rangle\right) \rightarrow \frac{1}{\sqrt{2}}\left(|10\rangle + |01\rangle\right). \tag{5.4}$$

- For $(x_1\ x_2) = (0\ 1)$, apply the **Z** gate of Section 2.3.1.1 to A,

$$\frac{1}{\sqrt{2}}\left(|00\rangle + |11\rangle\right) \rightarrow \frac{1}{\sqrt{2}}\left(|00\rangle - |11\rangle\right). \tag{5.5}$$

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

$$\frac{1}{\sqrt{2}}\left(|00\rangle + |11\rangle\right) \rightarrow \frac{1}{\sqrt{2}}\left(|10\rangle - |01\rangle\right). \tag{5.6}$$

This classical to quantum mapping is summarized in Table 5.1.

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 5.1 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:

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

$$
\begin{aligned}
|\psi_{00}\rangle &= \frac{1}{\sqrt{2}}\left(|00\rangle + |11\rangle\right) \rightarrow \frac{1}{\sqrt{2}}(|00\rangle + |10\rangle),\\
|\psi_{01}\rangle &= \frac{1}{\sqrt{2}}(|10\rangle + |01\rangle) \rightarrow \frac{1}{\sqrt{2}}(|11\rangle + |01\rangle),\\
|\psi_{10}\rangle &= \frac{1}{\sqrt{2}}(|00\rangle - |11\rangle) \rightarrow \frac{1}{\sqrt{2}}(|00\rangle - |10\rangle),\\
|\psi_{11}\rangle &= \frac{1}{\sqrt{2}}(|10\rangle - |01\rangle) \rightarrow \frac{1}{\sqrt{2}}(|11\rangle - |01\rangle).
\end{aligned} \tag{5.7}
$$

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

$$
\begin{aligned}
&\frac{1}{\sqrt{2}}(|00\rangle + |10\rangle) \rightarrow \frac{1}{\sqrt{2}}\frac{1}{\sqrt{2}}((|0\rangle + |1\rangle)|0\rangle + (|0\rangle - |1\rangle)|0\rangle) \equiv |00\rangle,\\
&\frac{1}{\sqrt{2}}(|11\rangle + |01\rangle) \rightarrow \frac{1}{\sqrt{2}}\frac{1}{\sqrt{2}}((|0\rangle - |1\rangle)|1\rangle) + (|0\rangle + |1\rangle)|1\rangle) \equiv |01\rangle,\\
&\frac{1}{\sqrt{2}}(|00\rangle - |10\rangle) \rightarrow \frac{1}{\sqrt{2}}\frac{1}{\sqrt{2}}((|0\rangle + |1\rangle)|0\rangle - (|0\rangle - |1\rangle)|0\rangle) \equiv |10\rangle,\\
&\frac{1}{\sqrt{2}}(|11\rangle - |01\rangle) \rightarrow \frac{1}{\sqrt{2}}\frac{1}{\sqrt{2}}((|0\rangle - |1\rangle)|1\rangle - (|0\rangle + |1\rangle)|1\rangle) \equiv -|11\rangle.
\end{aligned} \tag{5.8}
$$

Step 3: Measure the first qubit:

$$
\begin{aligned}
|00\rangle &\rightarrow 00,\\
|01\rangle &\rightarrow 01,\\
|10\rangle &\rightarrow 10,\\
-|11\rangle &\rightarrow 11.
\end{aligned} \tag{5.9}
$$

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 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 [232]. Alternatively, if both the transmitter and receiver are mobile, sharing may take place if and when they are close to each other [239].

5.2.2 N-Qubit Superdense Coding

N-qubit SuperDense coding (NSD) [237] 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 [240], 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

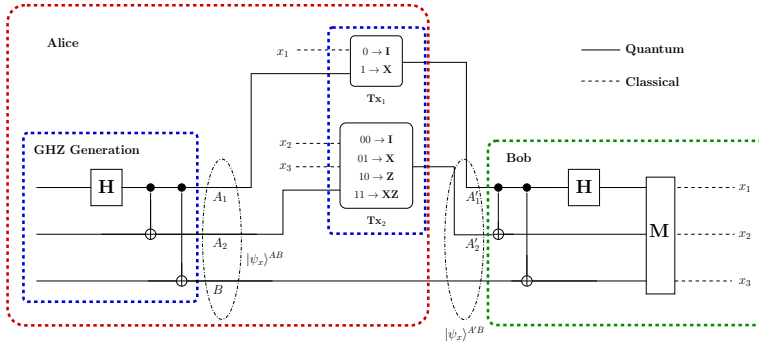


Figure 5.3: The quantum circuit for 3-qubit superdense coding. Alice generates the Greenberger-Horne-Zeilinger state $|\psi_x\rangle^{AB}$ using a **H** gate and CNOT gates respectively. Qubits A_1 and A_2 are used for encoding the 3-bit message, while the qubit B is pre-shared with Bob. Bob measures the received state $|\psi_x\rangle^{A'B}$ in the orthonormal basis, yielding the original classical message.

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 5.3, 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, is given by:

$$|\psi_x\rangle^{AB} = \frac{|000\rangle + |111\rangle}{\sqrt{2}}. \tag{5.10}$$

Here A constitutes a 2-qubit subsystem having qubits A_1 and A_2. The entangled triplet $|\psi_x\rangle^{AB}$ 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 5.3, 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 5.3. 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 $|\psi_x\rangle^{A'B}$, 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 $|\psi_x\rangle^{A'B}$ are orthonormal. Table 5.2 enlists one such mapping.

5.3 Entanglement-Assisted Classical Capacity

Under the assumptions discussed in Section 5.2.1, 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 [233, 241]. The corresponding EACC of 2SD has already been derived in [233, 241] 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

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

Table 5.2: Classical-to-quantum mapping for 3-qubit superdense coding. *The normalization factor $\frac{1}{\sqrt{2}}$ is ignored for simplicity.*

mutual information $I(x, y)$ between the transmitted symbol x and the received symbol y, i.e. we have [242]:

$$\mathrm{C} = \max_{\mathrm{P}(x)} I(x, y) \ = \max_{\mathrm{P}(x)} [\mathrm{H}(y) - \mathrm{H}(y|x)], \tag{5.11}$$

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:

$$\mathrm{C} = \log_2 M - \mathrm{H}(y|x), \tag{5.12}$$

which is further defined as follows [243, 244]:

$$\mathrm{C} = \log_2 M + \mathrm{E}\left[\sum_{m=0}^{M-1} \mathrm{P}(y|x = x^{(m)}) \log_2 \mathrm{P}(y|x = x^{(m)})\right], \tag{5.13}$$

using Eq. (10) and (11) of [243]. 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. (5.13), the capacity of NSD coding relying on a single noiseless pre-shared entangled qubit may be readily expressed as:

$$\mathrm{C}_{Nsd} = \frac{N + \sum_{m=0}^{M-1} \mathrm{P}(y|x = x^{(m)}) \log_2 \mathrm{P}(y|x = x^{(m)})}{N-1} \quad \text{cbits/use}, \tag{5.14}$$

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 5.1 to a 4-ary classical channel. Consequently, the channel transition probabilities of the induced classical channel may be encapsulated as:

$$\mathrm{P}(y|x = x^{(m)}) = \begin{cases} 1-p, & \text{if } \mathtt{E} = 0 \\ p/3, & \text{if } \mathtt{E} \in \{1, 2, 3\}, \end{cases} \tag{5.15}$$

[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. (5.13) can be therefore ignored.

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:

$$y_i = x_i \oplus e_i \quad \text{or} \quad e_i = y_i \oplus x_i. \tag{5.16}$$

Substituting Eq. (5.15) in Eq. (5.14) yields the entanglement-assisted classical capacity of 2SD over a quantum depolarizing channel, i.e we have:

$$\mathrm{C}_{2sd} = 2 + (1-p)\log_2(1-p) + p\log_2(p/3), \tag{5.17}$$

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 5.3. 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 5.3. It must be noted that different quantum errors may result in the same e, for example the third and sixth rows of Table 5.3 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:

$$\mathrm{P}(y|x = x^{(m)}) = \begin{cases} (1-p)^2 + p^2/9, & \text{if } \mathtt{E} \in \{0\} \\ (1-p)(p/3) + p^2/9, & \text{if } \mathtt{E} \in \{2, 3, 6, 7\} \\ 2(1-p)(p/3), & \text{if } \mathtt{E} \in \{4\} \\ 2p^2/9, & \text{if } \mathtt{E} \in \{1, 5\}, \end{cases} \tag{5.18}$$

where we have $m \in \{0, 1, \ldots, 7\}$. Eq. (5.18) may be substituted in Eq. (5.14) to compute the EACC of the 3SD scheme, when communicating over a quantum depolarizing channel. More explicitly, we have:

$$\mathrm{C}_{3sd} = \frac{3}{2} + \frac{1}{2}\sum_{m=0}^{7} \mathrm{P}(y|x = x^{(m)}) \log_2 \mathrm{P}(y|x = x^{(m)}). \tag{5.19}$$

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

5.4 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 [245, 132, 246]. Figure 5.4 shows the general schematic of the proposed system, which employs a classical IRCC [247, 248] for achieving the near-capacity performance. Furthermore, a classical symbol-based recursive URC having a generator polynomial of $G(D) = \frac{1}{1+D}$ [132] is used as a precoder for reaching the $(1, 1)$ point of perfect decoding convergence in the EXIT chart [249]. 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 5.4 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 5.4. Classical to quantum domain conversion

[3]Bit-to-symbol convertor is assumed to be inside the URC Encoder block of Figure 5.4.

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 5.3: List of all the possible quantum errors when the first and second qubit, A_1 and A_2 respectively, of the 3-qubit Greenberger-Horne-Zeilinger state $\frac{1}{\sqrt{2}}(|000\rangle + |111\rangle)$ are transmitted over a quantum depolarizing channel. *The normalization factor $\frac{1}{\sqrt{2}}$ is ignored for simplicity.*

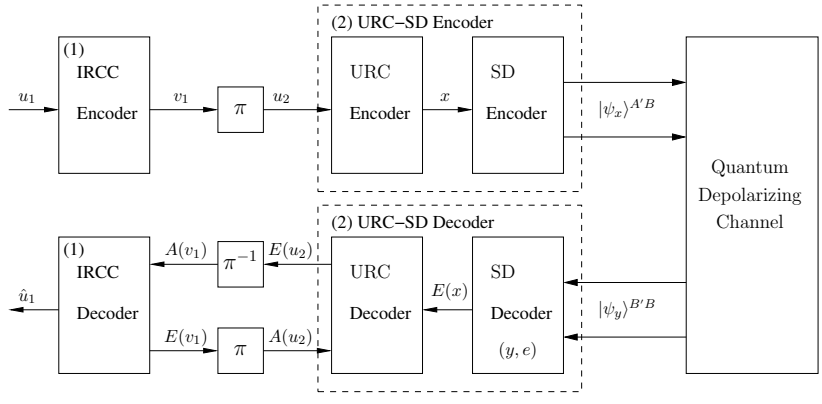


Figure 5.4: Schematic of the proposed IRCC-URC-SD classical-quantum communication system. ©Babar *et al.* [235]

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 Section 5.1. 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:

$$E(x) \approx \mathrm{P}(y|x), \tag{5.20}$$

where $\mathrm{P}(y|x)$ is given by Eq. (5.15) and (5.18) 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.

5.5 Near-Capacity Design

5.5.1 EXIT Charts

EXIT charts [245, 132, 246] 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 5.4, the EXIT chart visualizes the exchange of the following four MI terms:

[4] As illustrated earlier in Figure 5.1, the processed qubit(s) A' is transmitted over the noisy quantum channel, while B is shared between Alice and Bob over a noiseless channel.

(a) average *a priori* MI between u_2 and $A(u_2)$: $I_{A(u_2)}$,

(b) average *a priori* MI between v_1 and $A(v_1)$: $I_{A(v_1)}$,

(c) average *extrinsic* MI between u_2 and $E(u_2)$: $I_{E(u_2)}$, and

(d) 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 [243, 244]:

$$I_{E(u_2)} = \log_2 M + \mathrm{E}\left[\sum_{m=0}^{M-1} E(u_2^{(m)}) \log_2(E(u_2^{(m)}))\right], \tag{5.21}$$

and

$$I_{E(v_1)} = \log_2 M + \mathrm{E}\left[\sum_{m=0}^{M-1} E(v_1^{(m)}) \log_2(E(v_1^{(m)}))\right]. \tag{5.22}$$

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. (5.21) and (5.22) we have $M = 2$ and $m \in \{0, 1\}$. The resultant inner EXIT function T_{u_2} is given by:

$$I_{E(u_2)} = T_{u_2}[I_{A(u_2)}, p], \tag{5.23}$$

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

$$I_{E(v_1)} = T_{v_1}[I_{A(v_1)}]. \tag{5.24}$$

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. (5.23) and (5.24) 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 stair-case-shaped decoding trajectory, as the iterations proceed. Examples of EXIT charts will be given in Section 5.6.

5.5.2 Near-Capacity IRCC-URC-SD Design

We have exploited the area property of EXIT charts [250] for designing a near-capacity classical error correction code for our classical-quantum communication system of Figure 5.4. According to this property, the area under the normalized EXIT curve of the inner decoder is approximately equal to the attainable channel capacity [250], provided that the channel's input symbols are equiprobable. Since our system model of Figure 5.4 transmits classical information over a quantum depolarizing channel, the attainable channel capacity of the system is the entanglement-assisted classical capacity given in Eq. (5.14). However, as mentioned in Section 5.4, symbol-to-bit conversion takes place at the output of the URC decoder. This incurs a capacity loss [251]. 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. (5.15) to the bit-based probabilities $\mathrm{P}(y_i|x_i)$ for $i \in \{1, 2\}$, assuming that

[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 [132]:

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

the constituent bits are independent. More specifically, we get:

$$\begin{aligned} \mathrm{P}(y_i = x_i|x_i) &= 1 - \frac{2}{3}p, \\ \mathrm{P}(y_i \neq x_i|x_i) &= \frac{2}{3}p. \end{aligned} \tag{5.25}$$

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:

$$\mathrm{C}^i_{\mathrm{BSC}} = 1 + (1 - \frac{2}{3}p)\log_2(1 - \frac{2}{3}p) + \frac{2}{3}p\log_2(\frac{2}{3}p), \tag{5.26}$$

for $i \in \{1, 2\}$. Since 2 classical bits are transmitted per channel use in the 2SD scheme, the symbol-based capacity of Eq. (5.17) is reduced to the sum of the capacity of these two BSCs, which is equivalent to:

$$\begin{aligned} \mathrm{C}^{\mathrm{bit}}_{2sd} &= \sum_{i=1}^{2} \mathrm{C}^i_{\mathrm{BSC}} \\ &= 2 \cdot \left[1 + (1 - \frac{2}{3}p)\log_2(1 - \frac{2}{3}p) + \frac{2}{3}p\log_2(\frac{2}{3}p)\right], \end{aligned} \tag{5.27}$$

where $\mathrm{C}^i_{\mathrm{BSC}}$ is the capacity of the ith BSC given in Eq. (5.26). 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:

$$\mathrm{C}^{\mathrm{bit}}_{Nsd} = \frac{1}{N-1} \cdot \sum_{i=1}^{N} \mathrm{C}^i_{\mathrm{BSC}}, \tag{5.28}$$

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. (5.18) for each of the three constituent bits as follows:

$$\begin{aligned} \mathrm{P}(y_1 = x_1|x_1) &= \sum_{e_1=0} \mathrm{P}(y = x + e|x = x^{(m)}) = 1 - \frac{4}{3}p + \frac{8}{9}p^2, \\ \mathrm{P}(y_1 \neq x_1|x_1) &= \sum_{e_1=1} \mathrm{P}(y = x + e|x = x^{(m)}) = \frac{4}{3}p - \frac{8}{9}p^2, \\ \mathrm{P}(y_2 = x_2|x_2) &= \sum_{e_2=0} \mathrm{P}(y = x + e|x = x^{(m)}) = 1 - \frac{4}{3}p + \frac{8}{9}p^2, \\ \mathrm{P}(y_2 \neq x_2|x_2) &= \sum_{e_2=1} \mathrm{P}(y = x + e|x = x^{(m)}) = \frac{4}{3}p - \frac{8}{9}p^2, \\ \mathrm{P}(y_3 = x_3|x_3) &= \sum_{e_3=0} \mathrm{P}(y = x + e|x = x^{(m)}) = 1 - \frac{2}{3}p, \\ \mathrm{P}(y_3 \neq x_3|x_3) &= \sum_{e_3=1} \mathrm{P}(y = x + e|x = x^{(m)}) = \frac{2}{3}p. \end{aligned} \tag{5.29}$$

Using Eq. (5.13) as well as (5.29) and exploiting the fact that 3 classical bits are transmitted per 2 channel uses in 3SD, the symbol-based capacity of Eq. (5.19) is reduced to:

$$\begin{aligned} \mathrm{C}^{\mathrm{bit}}_{3sd} &= \frac{1}{2} \cdot \sum_{i=1}^{3} \mathrm{C}^i_{\mathrm{BSC}} \\ &= \frac{1}{2} \cdot [3 - 2 \times \mathrm{H}_2(\frac{4}{3}p - \frac{8}{9}p^2) - \mathrm{H}_2(\frac{2}{3}p)], \end{aligned} \tag{5.30}$$

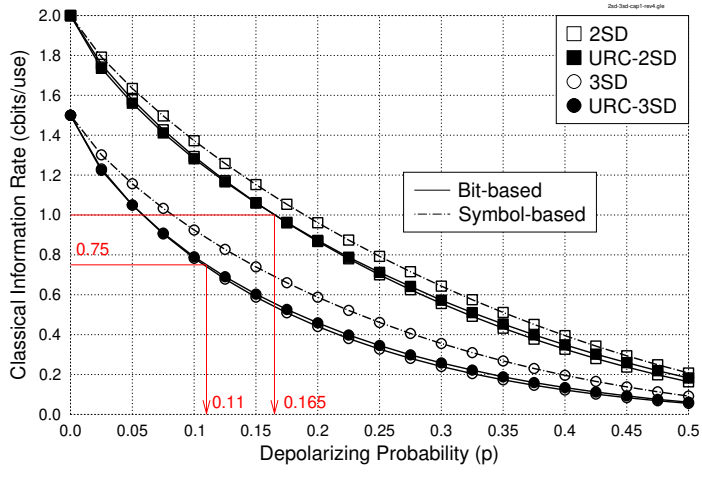


Figure 5.5: Classical information rate (cbits/use) versus quantum depolarizing probability for 2-qubit and 3-qubit superdense codes with and without URC. Symbol-to-bit conversion incurs a capacity loss for 2SD as well as 3SD. ©Babar *et al.* [235]

where C^i_{BSC} is the capacity of the ith BSC, while $H_2(z)$ is the binary entropy function, which is given by,

$$H_2(z) = -z\log_2(z) - (1-z)\log_2(1-z). \tag{5.31}$$

The capacity loss for both the 2SD and 3SD schemes is quantified in Figure 5.5, 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 5.5. 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 [250]. Therefore, our near-capacity 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, 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 [248], 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 [247, 248]:

$$\sum_{l=1}^{L} \varrho_l = 1 \ , \ R_o = \sum_{l=1}^{L} \varrho_l r_l \ , \ \varrho_l \in [0,1], \forall l \ , \tag{5.32}$$

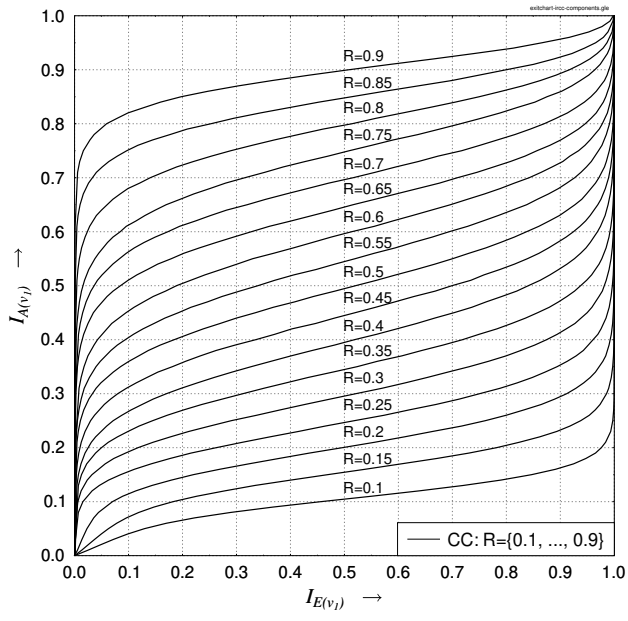


Figure 5.6: Normalized outer EXIT curves (inverted) of the 17 IRCC subcodes. ©Babar *et al.* [235]

which can be conveniently represented in the following matrix form:

$$\begin{bmatrix} 1 & 1 & \dots & 1 \\ r_1 & r_2 & \dots & r_L \end{bmatrix} \begin{bmatrix} \varrho_1 & \varrho_2 \dots & \varrho_L \end{bmatrix}^T = \begin{bmatrix} 1 \\ R_o \end{bmatrix}$$

$$\mathbf{C}\ \boldsymbol{\varrho} = \mathbf{d}\ . \tag{5.33}$$

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 [248]. 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 $r_l \in \{0.1, 0.15, 0.2, \dots, 0.85, 0.9\}$. Figure 5.6 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 5.6, 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:

$$I_{E(v_1)} = T_{v_1}\left[I_{A(v_1)}\right] = \sum_{l=1}^{L} \varrho_l\ T_{v_1,l}\left[I_{A(v_1)}\right]\ , \tag{5.34}$$

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 [247, 248] for optimizing the weighting coefficients of the IRCC subcodes by ensuring that a

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 5.4: Simulation parameters of the IRCC-URC-SD scheme of Figure 5.4.

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.

5.6 Results and Discussions I

Based on the near-capacity design of Section 5.5, 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 5.6.1 and 5.6.2, respectively.

5.6.1 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 5.7 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 [247, 248], the IRCC weight vector of Eq. (5.33) was optimized to get a narrow open tunnel as evident in Figure 5.7. The corresponding simulation parameters are summarized in Table 5.4, where only seven subcodes are activated. The tunnel of Figure 5.7 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 5.5, 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 $[10 \times \log_{10}(\frac{0.165}{0.15})] = 0.4$ 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,

[6]The difference in dB between two channel depolarizing probabilities p_1 and p_2 is calculated as follows [164, 157]: $\left(10 \times \log_{10} \frac{p_1}{p_2}\right)$.

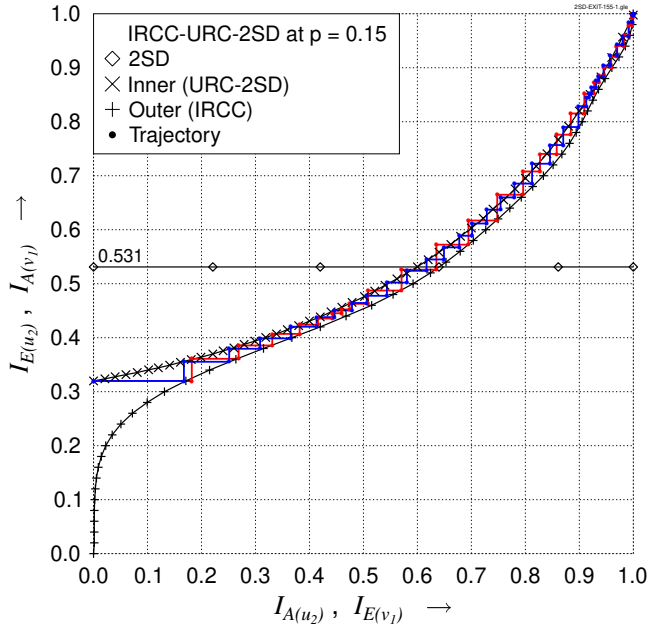


Figure 5.7: Normalized EXIT curves of the IRCC-URC-2SD system of Figure 5.4 at a depolarizing probability of 0.15 using the simulation parameters of Table 5.4. ©Babar *et al.* [235]

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 5.8 for the simulation parameters of Table 5.4. As it can be observed in Figure 5.8, 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 $\mathtt{I} = 16$ 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 5.9, 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 5.9 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 5.4 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 5.10 shows the resultant EXIT curves for $p = 0.15$ and $p = 0.125$. It can be

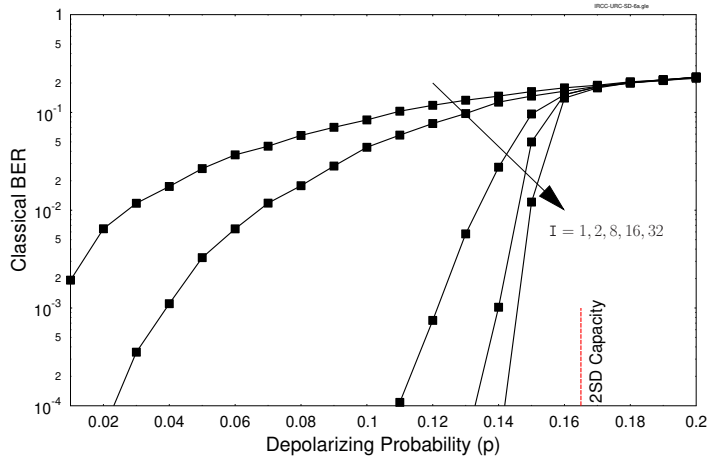


Figure 5.8: Achievable BER performance of the IRCC-URC-2SD scheme of Figure 5.4 upon increasing the number of iterations, i.e. $\mathtt{I} = \{1, 2, 8, 16, 32\}$. The simulation parameters are summarized in Table 5.4. The dashed-line at $p^* = 0.165$ marks the noise limit for a classical information rate of 1 cbit/use, which is obtained from the bit-based EACC curve of 2SD given in Figure 5.5.

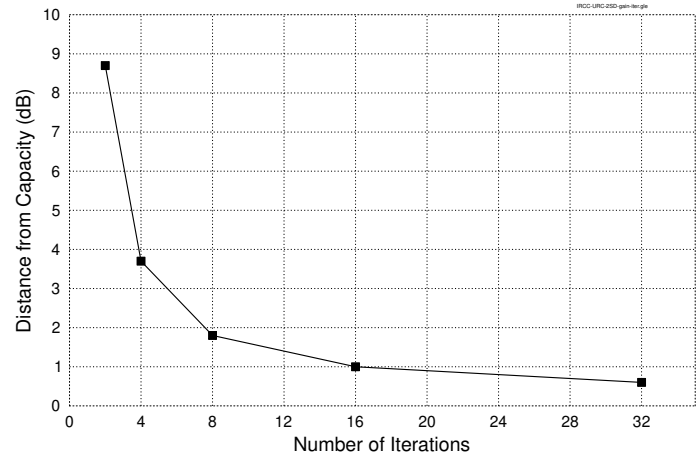


Figure 5.9: Performance of the IRCC-URC-2SD scheme of Figure 5.8, at a BER of 10^{-4}, which is quantified in terms of the distance from the bit-based EACC of 2SD, i.e. $p^* = 0.165$, as the number of iterations is increased.

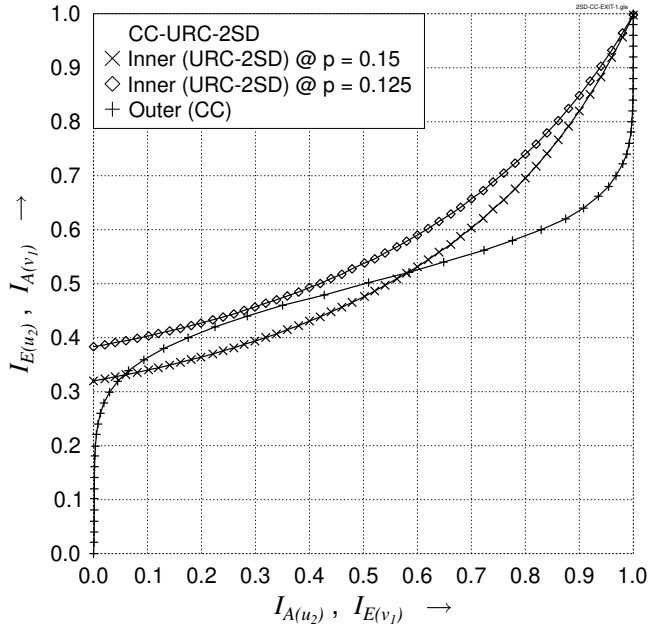


Figure 5.10: Normalized EXIT curves of the CC-URC-2SD system using the simulation parameters of Table 5.4, but only the 9th subcode of IRCC is active. ©Babar *et al.* [235]

observed in Figure 5.10 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 5.7. 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 5.7. 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.

5.6.2 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 5.11 shows the EXIT curves for our 3-qubit SD at a depolarizing probability of 0.1. The optimized IRCC weights are given in Table 5.5, 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 5.5 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 5.12 for the simulation parameters

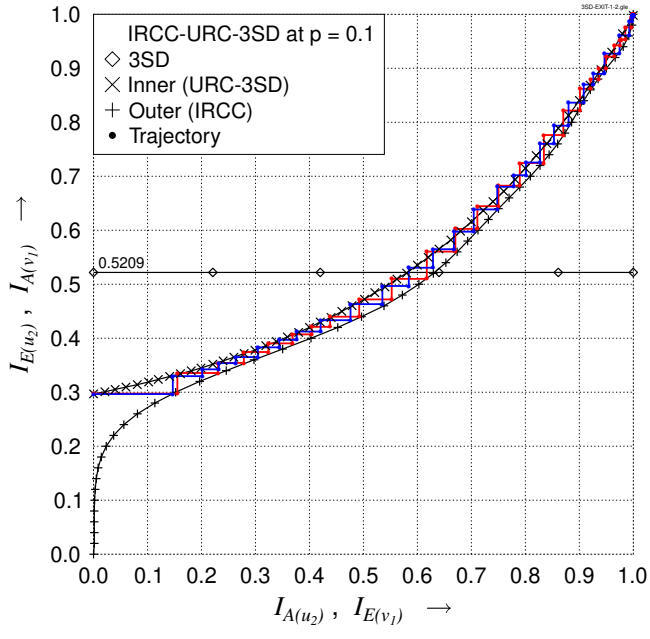


Figure 5.11: Normalized EXIT curves of the IRCC-URC-3SD system at a depolarizing probability of 0.1 using the simulation parameters of Table 5.5. ©Babar *et al.* [235]

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 5.5: Simulation parameters of the IRCC-URC-SD scheme of Figure 5.4.

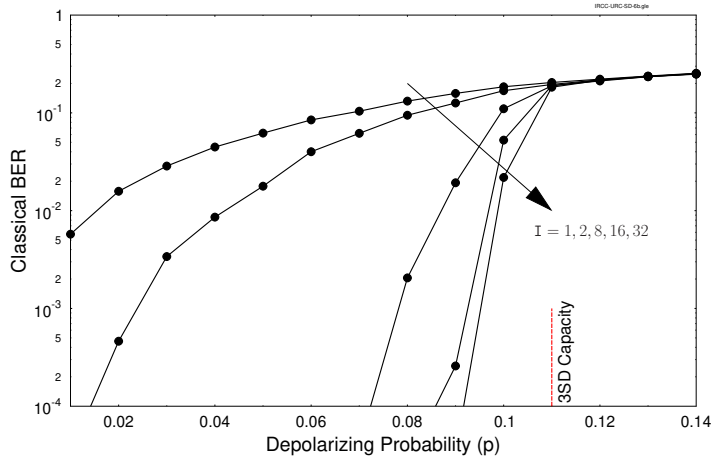


Figure 5.12: Achievable BER performance of the IRCC-URC-3SD scheme of Figure 5.4 with increasing number of iterations, i.e. $\mathtt{I} = \{1, 2, 8, 16, 32\}$. The simulation parameters are summarized in Table 5.5. The dashed-line at $p^* = 0.11$ marks the noise limit for a classical information rate of 0.75 cbit/use, which is obtained from the bit-based EACC curve of 3SD given in Figure 5.5.

of Table 5.5. Analogous to our IRCC-URC-2SD scheme in Figure 5.8, the performance in Figure 5.12 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 5.11. Furthermore, for $p \leq 0.1$, the performance tends to approach the noise limit of $p^* = 0.11$. This is also demonstrated in Figure 5.13, 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 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 5.4 against the classical Turbo Code (TC) in Figure 5.14 for both 2SD as well as 3SD designs. This was achieved by replacing the IRCC-URC unit of Figure 5.4 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 5.14. 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 5.9 and Figure 5.13 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 5.14. 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 5.6, 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. (5.15) and (5.18) 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.

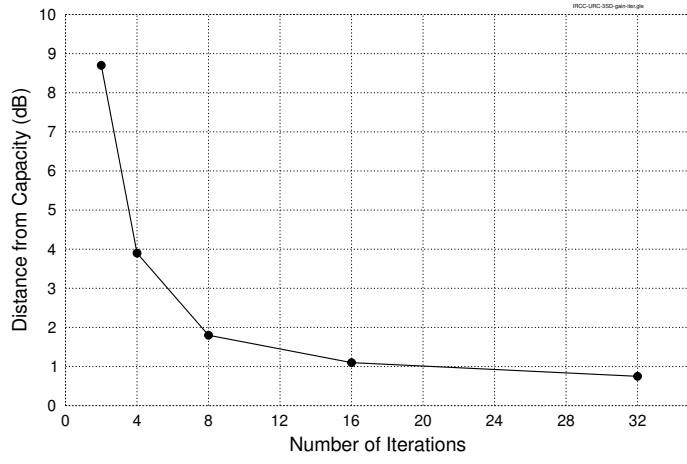


Figure 5.13: Performance of the IRCC-URC-3SD scheme of Figure 5.12 at a BER of 10^{-4}, which is quantified in terms of the distance from the bit-based EACC of 3SD, i.e. $p^* = 0.11$, as the number of iterations is increased.

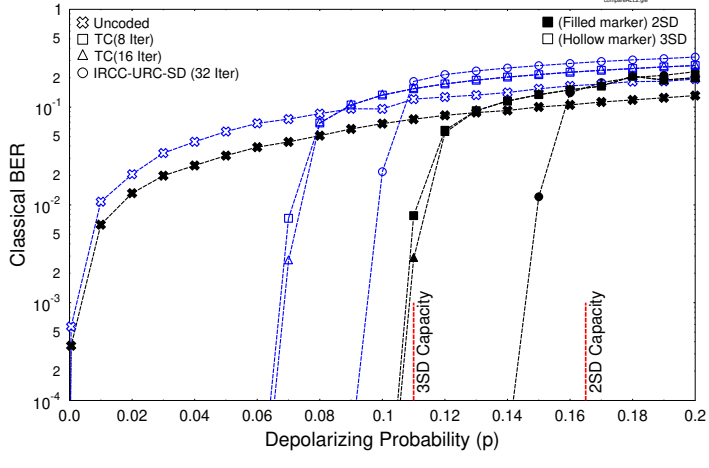


Figure 5.14: Comparison of the achievable BER performance of our IRCC-URC-SD scheme with TC-SD having a memory-3 1/2-rate TC as the outer component. 2SD and 3SD schemes are plotted with filled and hollow markers, respectively, and their simulation parameters are summarized in Table 5.4 and Table 5.5, respectively. Uncoded BER curves for both 2SD as well as 3SD are also plotted for comparison. Results are summarized in Table 5.6. ©Babar *et al.* [235]

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

Table 5.6: 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 5.14.

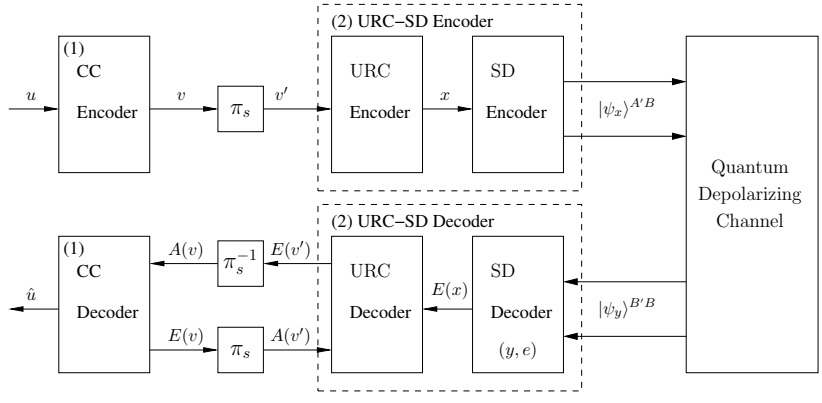


Figure 5.15: Schematic of the proposed symbol-based CC-URC-SD classical-quantum communication system. ©Babar *et al.* [236]

5.7 Symbol-Based Code Structure

Since the design of Figure 5.4 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 5.5, 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 [251] for classical discrete-memoryless channels and a modified binary LDPC code was proposed to circumvent this 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 5.15 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 $G(D) = \frac{1}{1+D}$ [132]. Similar to the bit-based design of Figure 5.4, classical to quantum domain conversion then takes place at the SD encoder of Figure 5.15 and the encoded qubits $|\psi_x\rangle^{A'B}$ are serially transmitted over the quantum depolarizing channel. The receiver of Figure 5.15 is also same as that of Figure 5.4 with the bit interleaver replaced by a symbol interleaver.

Since our proposed model of Figure 5.15 relies on symbol-based iterative decoding, we invoke non-binary EXIT charts of [243, 132, 246] for the sake of achieving a near-capacity performance.

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 5.7: Simulation parameters of the CC-URC-2SD scheme of Figure 5.15.

5.8 Results and Discussions II

Using the non-binary EXIT-charts, we have optimized our iterative code structure of Figure 5.15 to design a system with a coding rate of 1 cbit/use. According to the symbol-based capacity curve of Figure 5.5, 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 5.15 with the bit-based scheme of Figure 5.4, 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 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 5.16 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 5.16 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 5.16 is recorded in Figure 5.17 using the simulation parameters of Figure 5.7. As it can be observed, the turbo-cliff formulation starts around $p = 0.15$, 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 5.16, 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 5.18, 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 5.18 that there is only a negligible improvement in performance, when the number of iterations is

[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.

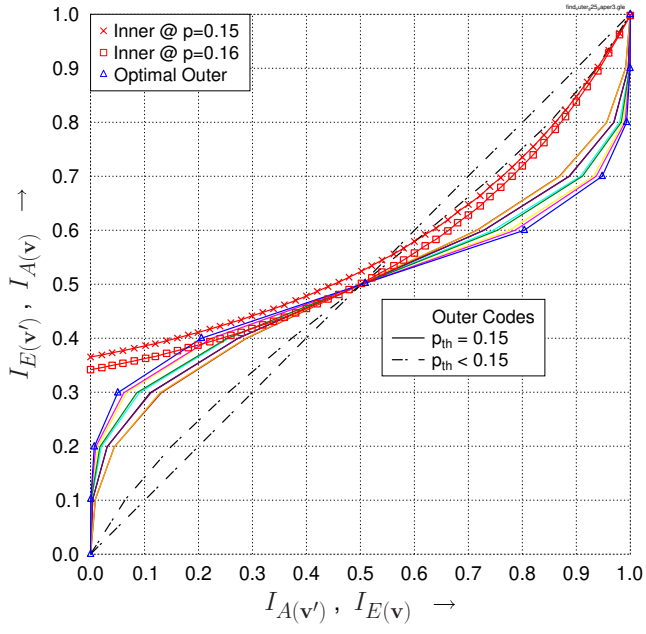


Figure 5.16: Normalized EXIT curves of the CC-URC-2SD system. Various 1/2-rate memory-4 convolutional codes were used as outer components. The optimal outer code has generator polynomials $(g_1, g_2) = (31, 36)_8$. ©Babar *et al.* [236]

increased from 15 to 20.

We further compare our symbol-based CC-URC-2SD to the bit-based IRCC-URC-2SD of Figure 5.4 in Figure 5.19, 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 5.19. 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 5.8, 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 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

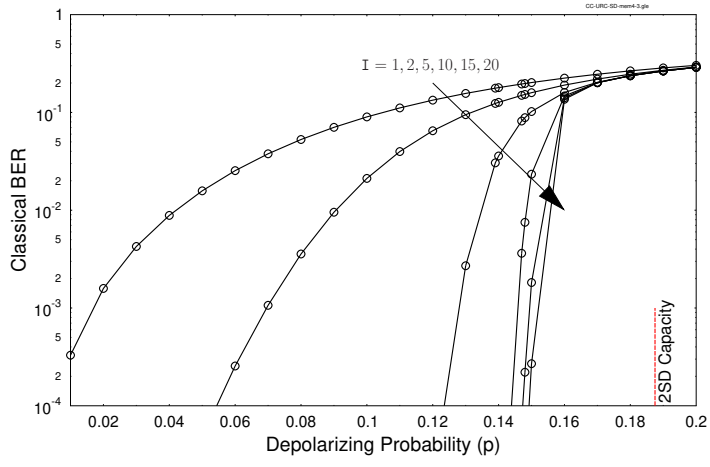


Figure 5.17: Achievable BER performance of the CC-URC-2SD scheme of Figure 5.15 with increasing number of iterations, i.e. $\mathtt{I} = \{1, 2, 5, 10, 15, 20\}$. Simulation parameters are summarized in Table 5.7. The dashed-line at $p^* = 0.1875$ marks the noise limit for a classical information rate of 1 cbit/use, which is obtained from the symbol-based EACC curve of 2SD given in Figure 5.5.

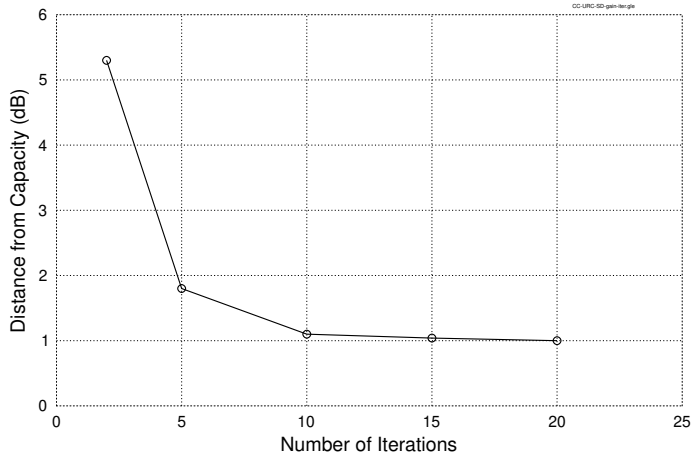


Figure 5.18: Performance of the CC-URC-2SD scheme of Figure 5.17, at a BER of 10^{-4}, which is quantified in terms of the distance from the symbol-based EACC of 2SD, i.e. $p^* = 0.1875$, as the number of iterations is increased.

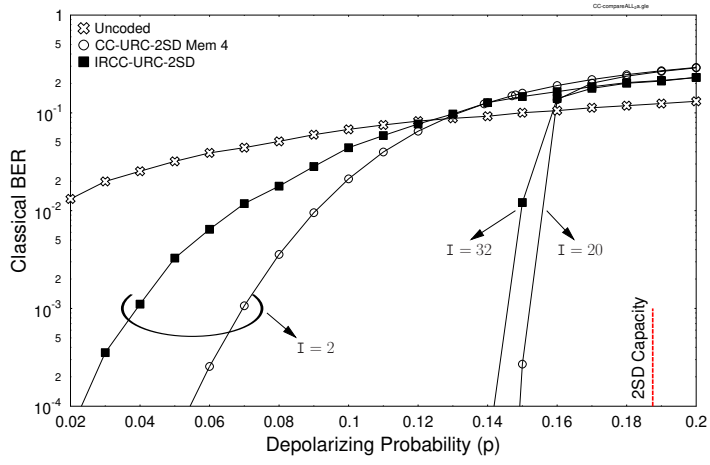


Figure 5.19: Comparison of the achievable BER performance of the bit-based IRCC-URC-2SD of Figure 5.4 and the symbol-based CC-URC-2SD design of Figure 5.15 using the simulation parameters of Table 5.4 and Table 5.7, respectively.

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 5.8: 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 5.21 to the bit-based IRCC-URC-2SD design of Figure 5.19 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.

EXIT curves for all the possible codes, as we previously did in Figure 5.16. It was found that the CC(2, 1, 2) having the generators $(g_1, g_2) = (7, 5)_8$ and the 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 5.20 together with the optimal memory-4 CC of Figure 5.16. All codes have the same decoding convergence threshold. The corresponding BER performance is compared in Figure 5.21 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 5.21. 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 5.8 by quantifying their performance at a BER of 10^{-4} in terms of the distance (dB) from the noise limit of $p^* = 0.1875$. All the three symbol-based configurations outperform the bit-based IRCC-URC-SD scheme.

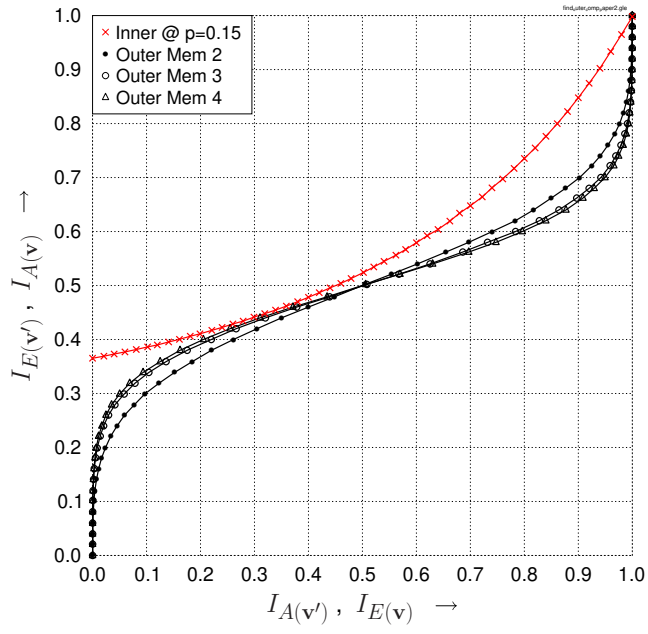


Figure 5.20: Normalized EXIT curves of the CC-URC-2SD system optimized for varying constraint lengths. Optimal outer components are plotted here: CC$(2,1,2)$ with $(g_1, g_2) = (7,5)_8$, CC$(2,1,3)$ with $(g_1, g_2) = (17,15)_8$ and CC$(2,1,4)$ with $(g_1, g_2) = (31,36)_8$. ©Babar *et al.* [236]

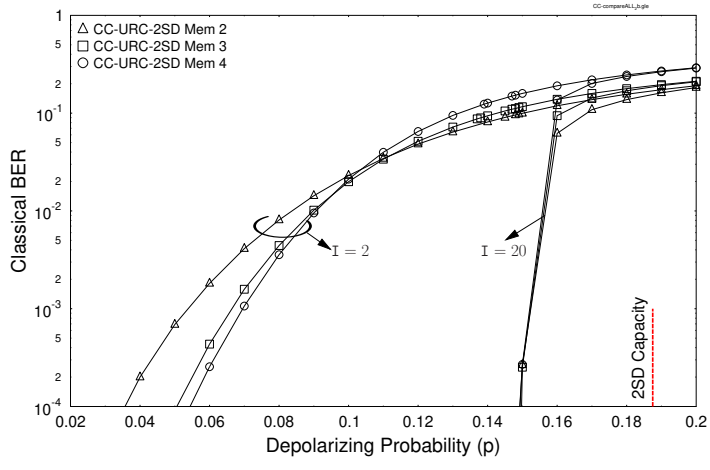


Figure 5.21: Comparison of the achievable BER performance of the symbol-based CC-URC-2SD design of Figure 5.15 optimized for varying constraint lengths. Optimal CCs are: CC$(2,1,2)$ with $(g_1, g_2) = (7,5)_8$, CC$(2,1,3)$ with $(g_1, g_2) = (17,15)_8$ and CC$(2,1,4)$ with $(g_1, g_2) = (31,36)_8$. Other simulation parameters are same as Table 5.7.

5.9 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 5.2, 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 5.2.1 and a rate of $\frac{N}{N-1}$ cbits/use for the general NSD scheme of Section 5.2.2. We then derived the EACC for the general NSD transmission in Section 5.3, which was also customized for the 2SD and 3SD schemes. In Section 5.4, we presented our proposed bit-based IRCC-URC-SD system of Figure 5.4, 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 (IRCC) decoders. Furthermore, we presented our EXIT-chart aided near-capacity design criterion in Section 5.5 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 5.6.1 and 5.6.2, respectively, which was benchmarked against the bit-based EACC given in Figure 5.5. It was demonstrated that our BER performance curves of Figure 5.8 and Figure 5.12 conform to the EXIT chart predictions of Figure 5.7 and Figure 5.11, respectively. Furthermore, the proposed system of Figure 5.4 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 5.10 and Figure 5.14, respectively. It was shown in Figure 5.14 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 5.4 incurs a capacity loss due to the symbol-to-bit conversion, as quantified in Figure 5.5. To overcome this capacity loss, we conceived a symbol-based code design in Section 5.7, which employs a single-component CC and a symbol interleaver in contrast to the IRCC and bit interleaver of Figure 5.4. We optimized our symbol-based CC-URC-2SD design with the aid of non-binary EXIT charts in Figure 5.16. Our simulation results of Section 5.8 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 5.19. 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 5.21 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 5.21 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 5.22, 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 Chapter **??**.

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 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.

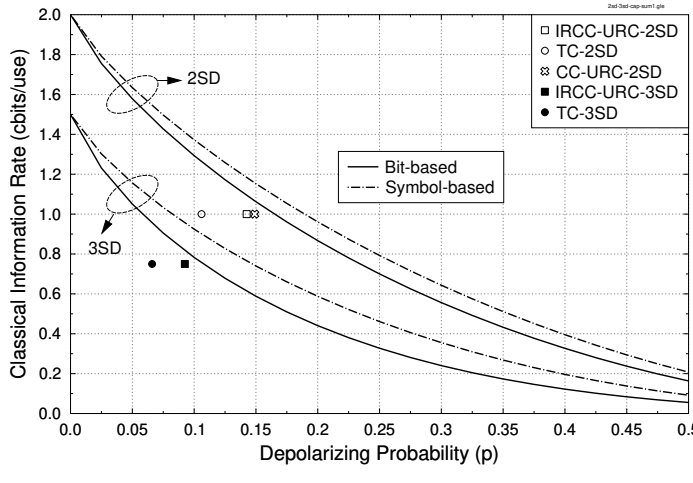


Figure 5.22: Classical information rate (cbits/use) versus the quantum depolarizing probability for bit-based and symbol-based 2SD as well as 3SD schemes. The performances of our designed bit-based IRCC-URC-SD ($\mathtt{I} = 32$) and symbol-based CC-URC-SD ($\mathtt{I} = 20$), along with the TC-SD ($\mathtt{I} = 16$) benchmark, are compared at a BER of 10^{-4}.

Part II

Near-Term Quantum Codes

Chapter 6

Quantum Coding Bounds and a Closed-Form Approximation of the Minimum Distance versus Quantum Coding Rate

Having paved the way for the classical to quantum coding evolution throughout the chapters of Part I, in this chapter of Part II we commence by outlining some quantum coding bounds derived from the classical coding bounds as a function of code-rate and codeword length. In Chapter 7 we will then discuss the design and performance of diverse Quantum Topological Error Correction Codes (QTECCs). These discussions will be followed in Chapter **??** by demonstrating, how to apply them for the protection of quantum gates. The last chapter of Part II of, namely Chapter **??**, will finally highlight the so-called 'universal decoding' principles, which are applicable for the decoding of any arbitrary linear codes. The associated performance results are based on Bose-Chaudhuri-Hocquenghem (BCH) and polar codes.

6.1 Introduction

To return to the subject of this chapter, the trade-off between the coding rate and the minimum distance as well as the codeword length is widely acknowledged in designing classical error correction codes. Since most of the QSCs are derived from their classical counterparts, the same problem persists. Even though extensive efforts have been invested in designing the QSCs, the question of how to determine the realistically achievable minimum distance d of a QSC given the quantum coding rate r_Q and codeword length n remains unresolved. For example, for a given codeword length of $n = 128$ and quantum coding rate of $r_Q = 1/2$, the achievable minimum distance is loosely bounded by $11 < d < 22$, while for $n = 1024$ and $r_Q = 1/2$, the achievable minimum distance is bounded by $78 < d < 157$. Naturally, having such a wide range of estimated minimum distance is undesirable. For binary classical codes, this problem has been circumvented by the closed-form

approximation proposed by Akhtman *et al.* [95]. Therefore, in this chapter, we present an appealingly simple and invertible closed-form approximation for determining the realistically achievable minimum distance of a QSC, given the codeword length n and the quantum coding rate r_Q. Our formulation is suitable for both the idealized asymptotical case and for the practical finite-length codeword. Furthermore, our proposed closed-form approximation is also applicable to the family of EA-QSCs.

This chapter is organized as follows. In Section 6.2, we survey the existing quantum coding bounds and derive the bounds by exploiting the classical-to-quantum isomorphism. This is followed by our closed-form approximation proposed for the idealized asymptotical limit in Section 6.3. Since the asymptotical limit has limited relevance when it comes to practical implementations, we also proposed an approximation for practical finite-length codewords. To unify the quantum coding bound approximation for both entanglement-assisted and unassisted QSCs, we derive a closed-form approximation for arbitrarily entangled QSCs in Section 6.5. Finally, we conclude this chapter in Section 6.6.

6.2 On Classical to Quantum Coding Bounds

In this section, we present the quantum version of the most well-known classical coding bounds, namely the Singleton bound [252] and Hamming bound [253], which serve as the upper bounds, as well as the Gilbert-Varshamov (GV) bound [254], which acts as the lower bound. Although there are several ways of deriving the coding bounds in the quantum domain, we are interested in exploring the duality of coding bounds in the classical and quantum domain. Therefore, we present the derivation of quantum coding bounds using the classical to quantum isomorphism approach and demonstrate that the final results agree with the coding bounds that are derived from a purely quantum domain perspective.

6.2.1 Singleton Bound

The Singleton bound of classical binary code constructions $\mathcal{C}(n, k, d)$ is defined as

$$n - k \geq d - 1, \tag{6.1}$$

where the notation n denotes the codeword length, k for the length of the uncoded information segment, and d is the minimum distance amongst the legitimate codewords in the codebook $\mathcal{C}$. The Singleton bound acts as an upper bound in classical code constructions. The bound implies that the number of rows in a PCM associated with the length of the syndrome vector, which is equal to $(n-k)$, has to be greater than $(d-1)$. For the QSC $\mathcal{C}[n, k, d]$, the rows of the PCM correspond to the number stabilizer operators. Since the stabilizer formalism has to correct both the bit-flip errors and the phase-flip errors, the classical Singleton bound of Eq. (6.1) can be readily transformed into the quantum Singleton bound as follows:

$$n - k \geq 2(d - 1), \tag{6.2}$$

where n now may also be referred to as the number of physical qubits and k is the number of logical qubits. In order to show explicitly the trade-off between the minimum distance and the quantum coding rate, Eq. (6.2) can be modified to

$$\frac{k}{n} \leq 1 - 2\left(\frac{d-1}{n}\right). \tag{6.3}$$

In the quantum domain, the Singleton bound is also known as the Knill-Laflamme bound [89]. The QSCs achieving the quantum Singleton bound by satisfying the equality are classified as the quantum Maximum Separable Distance (MDS) codes. One of the well-known QSCs having a minimum distance of $d = 3$ that reaches the quantum Singleton bound is the 'perfect' 5-qubit code $\mathcal{C}[n, k, d] = \mathcal{C}[5, 1, 3]$.

6.2.2 Hamming Bound

In classical binary coding, a codebook $\mathcal{C}(n, k, d)$ maps the information words containing k bits into a codeword of length n bits. The maximal number of errors, which is denoted by t that can be corrected by codebook $\mathcal{C}$ is given by

$$t = \lfloor \frac{d-1}{2} \rfloor. \tag{6.4}$$

Therefore the maximum size of a binary codebook $|\mathcal{C}| = 2^k$ is bounded by the sphere-packing bound which is defined as:

$$2^k \leq \frac{2^n}{\sum_{j=0}^{t=\lfloor \frac{d-1}{2} \rfloor} \binom{n}{j}}. \tag{6.5}$$

Since the QSCs have to correct three different types of errors – namely the bit-flip errors ($\mathbf{X}$), phase-flip errors ($\mathbf{Z}$), as well as both bit-flip and phase-flip errors ($\mathbf{Y}$) – the size of the codebook for a QSC $\mathcal{C}[n, k, d]$ is now bounded by

$$2^k \leq \frac{2^n}{\sum_{j=0}^{t=\lfloor \frac{d-1}{2} \rfloor} \binom{n}{j} 3^j}. \tag{6.6}$$

By modifying Eq. (6.6), we can express explicitly the bound of the quantum coding rate as a function of the minimum distance d and codeword length n, as shown below:

$$\frac{k}{n} \leq 1 - \frac{1}{n} \log_2 \left(\sum_{j=0}^{t=\lfloor \frac{d-1}{2} \rfloor} \binom{n}{j} 3^j \right). \tag{6.7}$$

If n tends to ∞, we obtain

$$\frac{k}{n} \leq 1 - \left(\frac{d}{2n} \right) \log_2 3 - H\left(\frac{d}{2n} \right), \tag{6.8}$$

where $H(x)$ is the binary entropy of x formulated as $H(x) = -x \log_2 x - (1-x) \log_2 (1-x)$. Equation (6.7) and (6.8) are also known as the quantum Hamming bound [62], which also constitutes the upper bound for QSC constructions.

6.2.3 Gilbert-Varshamov Bound

The analogy exploited to derive the quantum Hamming bound may also be used for transforming the classical Gilbert-Varshamov (GV) bound, namely the lower bound for classical code constructions, into its quantum counterpart. In the classical domain, the GV bound for classical codes $\mathcal{C}(n, k, d)$ is formulated as

$$2^k \geq \frac{2^n}{\sum_{j=0}^{d-1} \binom{n}{j}}. \tag{6.9}$$

Considering that the QSCs have to tackle three different types of errors, the size of the codebook $\mathcal{C}[n, k, d]$ is bounded by

$$2^k \geq \frac{2^n}{\sum_{j=0}^{d-1} \binom{n}{j} 3^j}. \tag{6.10}$$

Hence, we can readily derive the quantum GV bound, the lower bound of the quantum coding rate as a function of the minimum distance d and codeword length n as follows:

$$\frac{k}{n} \geq 1 - \frac{1}{n} \log_2 \left(\sum_{j=0}^{d-1} \binom{n}{j} 3^j \right). \tag{6.11}$$

Again, if n aprroaches ∞, we obtain

$$\frac{k}{n} \geq 1 - \left(\frac{d}{n}\right) \log_2 3 - H\left(\frac{d}{n}\right), \tag{6.12}$$

where $H(x)$ is the binary entropy of x. The quantum GV bounds in Eq. (6.11) and (6.12) are valid for non-CSS QSCs. However, a special case should be considered for dual-containing quantum CSS codes. It will be shown in Section 6.4 that for some dual-containing CSS codes the code constructions violate the quantum GV bound. Hence, a special bound has to be derived to accomodate the dual-containing CSS codes. In the classical domain, a binary code $\mathcal{C}(n, k, d)$ maps a k-bit information word into an n-bit encoded codeword. The number of syndrome measurement operators is determined by the number of rows in the PCM $\mathbf{H}$ of $\mathcal{C}(n, k, d)$, which is equal to $2^{(n-k)}$. With a simple modification of Eq. (6.9), the number of syndrome measurement operators in $\mathcal{C}(n, k, d)$ is bounded by

$$2^{(n-k)} \leq \left(\sum_{j=0}^{d-1} \binom{n}{j}\right). \tag{6.13}$$

Recall that the dual-containing quantum CSS codes rely on dual-containing classical binary codes, which satisfy the symplectic criterion of Eq. (**??**) and also comply with the constraint of $\mathbf{H}_z = \mathbf{H}_x$. Explicitly, half of the stabilizer operators of $\mathcal{C}[n, k, d]$ are mapped onto $\mathbf{H}_z$, while the other half are mapped onto $\mathbf{H}_x$. Therefore, the number of stabilizer operators of a dual-containing quantum CSS code is bounded by

$$2^{\frac{(n-k)}{2}} \leq \left(\sum_{j=0}^{d-1} \binom{n}{j}\right). \tag{6.14}$$

Based on Eq. (6.14), we may formulate the lower bound on the quantum coding rate of a dual-containing quantum CSS code as follows:

$$\frac{k}{n} \geq 1 - \frac{2}{n} \log_2 \left(\sum_{j=0}^{d-1} \binom{n}{j}\right). \tag{6.15}$$

As n approaches ∞, we obtain the quantum GV bound for CSS codes, as suggested in [90], which is formulated as

$$\frac{k}{n} \geq 1 - 2H\left(\frac{d}{n}\right), \tag{6.16}$$

where $H(x)$ is the binary entropy of x. Based on the discussions above, we compare the asymptotic classical and quantum coding bounds in Table. 6.1 as well as in Fig. 6.2. Since the QSCs are designed to mitigate both bit-flip errors as well as phase-flip errors, the bounds of QSCs are significantly lower than those of their classical counterparts. Nevertheless, the general concept still holds, the Singleton bound serves as a loose upper bound, whilst the Hamming bound is a tighter upper bound.

6.3 Quantum Coding Bounds in the Asymptotical Limit

Although the classical to binary isomorphism assists us in the development of QSCs from the well-known classical code designs, the issue of determining the actual achievable minimum distance, given the coding rate and the codeword length remains unresolved. In the classical domain as we described previously, finding the unique solution to the realistically achievable minimum distance of binary classical codes is still an open problem, even though the upper bound and lower bound of the quantum coding rate versus the achievable minimum distance can be found in the literature [252, 253, 254, 255, 256]. The bounds for the classical code constructions are listed in Table. 6.2, while the corresponding asymptotic bounds are also plotted in Fig. 6.1. Observe in the figure that we normalized d by n for the sake of being able to compare different-length codes under fair experimental conditions. To elaborate briefly, it is plausible that a longer code is capable of achieving a higher minimum distance and a higher error correction capability, but this normalized minimum distance allows us to gauge, whether the codelength extension does or does not yield an attractive increase in terms of d. This is particularly important, because longer codes generally impose a higher decoding complexity. In

Table 6.1: Comparison of various classical and quantum coding bounds.

Coding Bound	Asymptotic	
	Classical	Quantum
Singleton	$\frac{k}{n} \leq 1 - \left(\frac{d}{n}\right)$	$\frac{k}{n} \leq 1 - 2\left(\frac{d}{n}\right)$
Hamming	$\frac{k}{n} \leq 1 - H\left(\frac{d}{2n}\right)$	$\frac{k}{n} \leq 1 - \left(\frac{d}{2n}\right)\log_2 3 - H\left(\frac{d}{2n}\right)$
GV	$\frac{k}{n} \geq 1 - H\left(\frac{d}{n}\right)$	$\frac{k}{n} \geq 1 - \left(\frac{d}{n}\right)\log_2 3 - H\left(\frac{d}{n}\right)$
Coding Bound	Finite-length	
	Classical	Quantum
Singleton	$\frac{k}{n} \leq 1 - \left(\frac{d-1}{n}\right)$	$\frac{k}{n} \leq 1 - 2\left(\frac{d-1}{n}\right)$
Hamming	$\frac{k}{n} \leq 1 - \frac{1}{n}\log_2\left(\sum_{j=0}^{t=\lfloor\frac{d-1}{2}\rfloor}\binom{n}{j}\right)$	$\frac{k}{n} \leq 1 - \frac{1}{n}\log_2\left(\sum_{j=0}^{t=\lfloor\frac{d-1}{2}\rfloor}\binom{n}{j}3^j\right)$
GV	$\frac{k}{n} \geq 1 - \frac{1}{n}\log_2\left(\sum_{j=0}^{d-1}\binom{n}{j}\right)$	$\frac{k}{n} \geq 1 - \frac{1}{n}\log_2\left(\sum_{j=0}^{d-1}\binom{n}{j}3^j\right)$

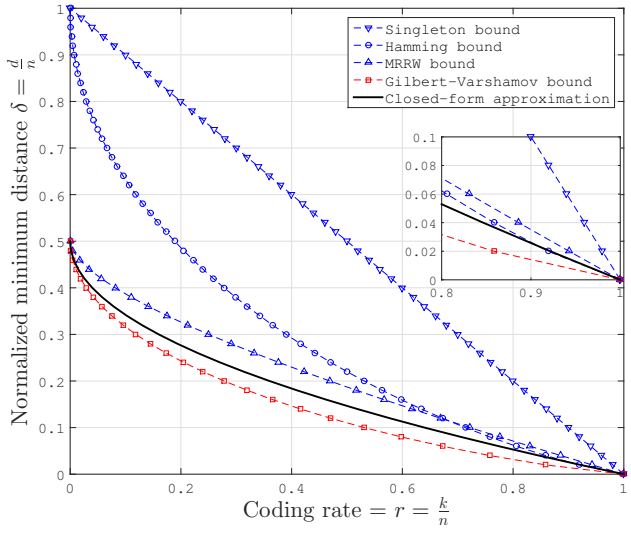


Figure 6.1: The trade-off between classical coding rate r and normalized minimum distance δ as described by classical binary coding bounds. A simple quadratic function $r(\delta) = (2\delta - 1)^2$, which satisfies all of the bounds, acts as a closed-form approximation for classical binary error correction codes as suggested in [95]. ©Chandra *et al.* [204]

the classical domain, the tightest known lower bound was derived by Gilbert [254]. The Hamming bound [253] serves as a tight upper bound for high coding rates, while the McEliece-Rodemich-Rumsey-Welch (MRRW) bound [255] serves as the tightest upper bound for moderate and low coding rates. As seen in Fig. 6.1, the gap between the tight upper bounds and the lower bound is quite narrow. It was observed in [95] that a simple

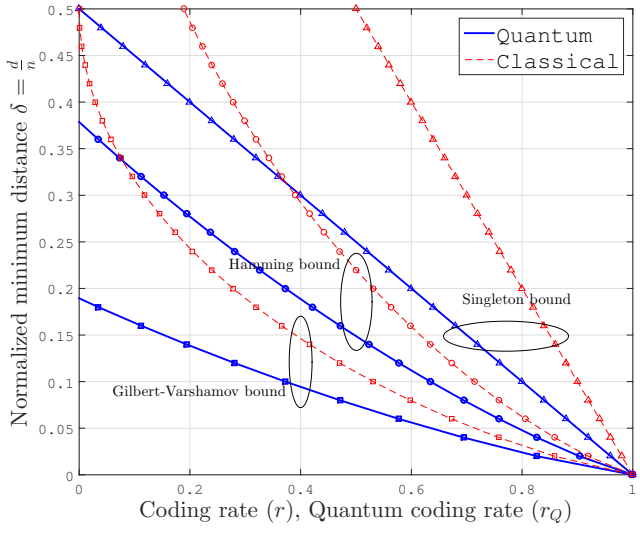


Figure 6.2: The evolution from asymptotic classical binary coding bounds to the asymptotic quantum coding bounds. ©Chandra *et al.* [204]

quadratic expression $r(\delta) = (2\delta - 1)^2$, where δ denotes the normalized minimum distance d/n, satisfies all the known asymptotic bounds.

Table 6.2: The coding bounds for classical code constructions, with a minor modification from [95].

Classical Bound	Finite	Asymptotic	Notes
Singleton [252]	$\frac{k}{n} \leq 1 - \left(\frac{d-1}{n}\right)$	$\frac{k}{n} \leq 1 - \left(\frac{d}{n}\right)$	a loose upper bound
Hamming [253]	$\frac{k}{n} \leq 1 - \frac{1}{n}\log_2\left(\sum_{j=0}^{t=\lfloor\frac{d-1}{2}\rfloor}\binom{n}{j}\right)$	$\frac{k}{n} \leq 1 - H\left(\frac{d}{2n}\right)$	tight upper bound for very high code rate
MRRW [255]		$\frac{k}{n} \leq H\left(\frac{1}{2} - \sqrt{\frac{d}{n}\left(1 - \frac{d}{n}\right)}\right)$	tightest known asymptotic upper bound for medium and low rate codes
Plotkin [256]	$\frac{k}{n} \leq \frac{1}{n}\left(1 - \log_2\left(2 - \frac{n}{d}\right)\right)$		tight upper bound for finite-length at $\delta > \frac{1}{2}$
GV [254]	$\frac{k}{n} \geq 1 - \frac{1}{n}\log_2\left(\sum_{j=0}^{d-1}\binom{n}{j}\right)$	$\frac{k}{n} \geq 1 - H\left(\frac{d}{n}\right)$	tightest known lower bound

The well-known bounds for QSC constructions are listed in Table. 6.3 and they are also portrayed in Fig. 6.3. The quantum Singleton bound serves as the loose upper bound, the quantum Hamming bound as a tighter upper bound, and quantum GV bound as the tightest lower bound. However, a wide discrepancy can be observed between the upper bound and the lower bound. For the sake of narrowing this gap, the quantum Rain bound was derived using quantum weight enumerators [257]. To elaborate a little further, the quantum Rain bound states that any quantum code of length n can correct at most $\lfloor\frac{n-1}{6}\rfloor$ errors. The resultant bound is only a function of codeword length n. Hence, under the asymptotic limit, the quantum Rain bound is a

straigth line at $\delta = 1/3$, which does not exhibit any further trade-off between the quantum coding rate and the minimum distance. In order to enhance the accuracy of the quantum Rain bound, Sarvepalli and Klappenecker derived a quantum version of the Griesmer bound [190]. By utilizing the quantum Griesmer bound and also the quantum Rain bound, a stronger bound was created for CSS type constructions. In this treatise, we will refer to this particular bound as the quantum Griesmer-Rain bound. For the sake of tightening the upper bound, Ashikhmin and Litsyn generalized the classical linear programming approach to the quantum domain using the MacWilliams identities [258]. The resultant quantum linear programming bound was proven to be tighter than the quantum Hamming bound in the low coding rate domain. As the quantum coding rate approaches zero, the achievable normalized minimum distance returned by the quantum Griesmer-Rain bound becomes $\delta = 0.33$ and that of the quantum linear programming bound becomes $\delta = 0.32$.

Recall from Section 3.3 that the QSCs may exhibit either a CSS or non-CSS structure. For CSS codes, the minimum distance is upper-bounded by the quantum Hamming bound for moderate to high quantum coding rates and by the quantum Griesmer-Rain bound for low coding rates, while it is also lower-bounded by the quantum GV bound for CSS codes. On the other hand, for non-CSS QSCs, the minimum distance is upper-bounded by the quantum Hamming bound for moderate to high coding rates and by the quantum linear-programming bound for low coding rates. It is also lower-bounded by the quantum GV bound for general quantum stabilizer codes. Even though substantial efforts have been invested in tightening the gap between the upper and lower bounds, a significant amount of discrepancy persists. Hence, creating a simple approximation may be beneficial for giving us further insights into the realistic construction of QSCs.

Table 6.3: The well-known quantum coding bounds found in the literature.

Quantum Bound	Finite-Length	Asymptotic	Notes
Singleton [89]	$\frac{k}{n} \leq 1 - 2\left(\frac{d-1}{n}\right)$	$\frac{k}{n} \leq 1 - 2\left(\frac{d}{n}\right)$	very loose upper bound
Hamming [62]	$\frac{k}{n} \leq 1 - \frac{1}{n}\log_2\left(\sum_{j=0}^{t=\lfloor\frac{d-1}{2}\rfloor}\binom{n}{j}3^j\right)$	$\frac{k}{n} \leq 1 - \left(\frac{d}{2n}\right)\log_2 3 - H\left(\frac{d}{2n}\right)$	tight upper bound for moderate and high coding rate
Griesmer-Rain [257, 190]	$\frac{k}{n} \leq 1 - \left(\frac{3d-4}{n}\right)$	$\frac{k}{n} \leq 1 - 3\left(\frac{d}{n}\right)$	tighter upper bound for low coding rates CSS codes
Linear Programming [258]		$\frac{k}{n} \leq H(\tau) + \tau\log_2 3 - 1$ $\tau = \frac{3}{4} - \frac{1}{2}\delta - \frac{1}{2}\sqrt{3\delta(1-\delta)}$	strengthen the upper bound
GV [62]	$\frac{k}{n} \geq 1 - \frac{1}{n}\log_2\left(\sum_{j=0}^{d-1}\binom{n}{j}3^j\right)$	$\frac{k}{n} \geq 1 - \left(\frac{d}{n}\right)\log_2 3 - H\left(\frac{d}{n}\right)$	tight lower bound for general stabilizer codes
GV for CSS [90]		$\frac{k}{n} \geq 1 - 2H\left(\frac{d}{n}\right)$	tight lower bound for CSS codes
	$\frac{k}{n} \geq 1 - \frac{2}{n}\log_2\left(\sum_{j=0}^{d-1}\binom{n}{j}\right)$		lower bound for dual-containing CSS codes

Analogous to the classical closed-form approximation of [95], we also found that there exists a simple closed-form quadratic approximation, which satisfies all the well-known quantum coding bounds. Explicitly for quantum stabilizer codes, the following quadratic function was found to satisfy all the quantum coding bounds:

$$r_Q(\delta) = \frac{32}{9}\delta^2 - \frac{16}{3}\delta + 1 \text{ for } 0 \leq \delta \leq 0.2197. \tag{6.17}$$

We will further elaborate on the selection of this function in Section 6.5. It is important to note that the closed-form approximation is subject to the asymptotical bound for either CSS type or non-CSS type quantum

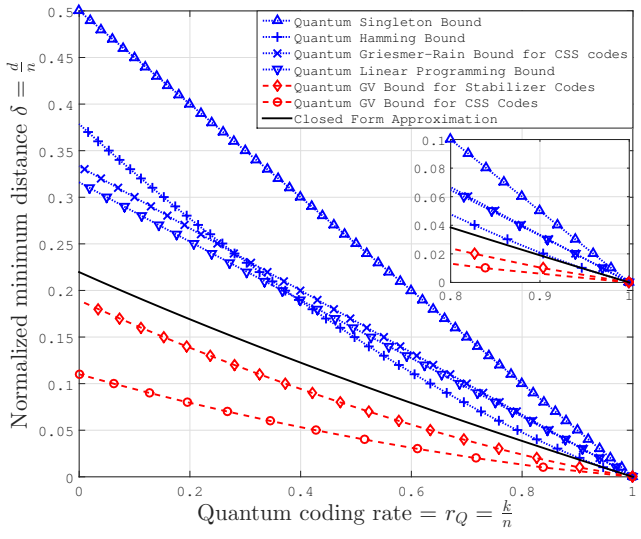


Figure 6.3: The trade-off between quantum coding rate r_Q and normalized minimum distance δ is characterized using quantum coding bounds. A simple quadratic closed-form $r_Q(\delta) = \frac{32}{9}\delta^2 - \frac{16}{3}\delta + 1$ satisfies all of the well-known quantum coding bounds, which is portrayed by black solid lines. The blue dashed lines portrays the upper bounds, while the red dashed lines denotes the lower bounds. ©Chandra *et al.* [204]

code constructions. The closed-form approximation in Eq. (6.17) offers the benefit of simplicity and it has the inverse function as given by

$$\delta(r_Q) = \frac{3\left(\sqrt{2} - \sqrt{r_Q + 1}\right)}{4\sqrt{2}} \text{ for } 0 \leq r_Q \leq 1. \tag{6.18}$$

This closed-form approximation suggests that it is possible to create a code construction whose minimum distance grows linearly with the codeword length at the asymptotical limit since for a given quantum coding rate r_Q, it will correspond to a unique constant value of δ.

6.4 Quantum Coding Bounds on Finite-Length Codes

The asymptotic limits are only relevant for $n \to \infty$. For practical applications, we require code constructions with shorter codeword length, which necessitates a different formulation for the quantum coding bounds. Finding a closed-form approximation will be beneficial for determining the realistically attainable minimum distance for the given code parameters. The well-known quantum coding bounds are listed in Table 6.3 and also portrayed in Fig. 6.3. It is clearly seen that a simple quadratic approximation can satisfy all the well-known bounds. For the finite-length quantum codes, we propose the closed-form approximation of

$$r_Q(n, \delta) = a\delta^2 + b\delta + c. \tag{6.19}$$

To arrive at the closed-form approximation in Eq. (6.19), we have to determine three definitive points corresponding to realistic quantum code constructions. As an example in this treatise, we use three QSC constructions from the literature as listed below:

- For uncoded logical qubits and unity rate codewords, we have

$$r_Q(n,\delta) = r(n,\frac{1}{n}) = 1. \tag{6.20}$$

- For a high coding rate, we will use the construction given in [61]. For $n = 2^j$, there is a quantum stabilizer code construction $[n,k,d] = [n, n-j-2, 3]$, which can be used to correct $t = 1$ error. This code construction reaches the quantum Hamming bound. For arbitrary n, it can be written as

$$r_Q(n,\delta) = r(n,\frac{3}{n}) = 1 - \frac{1}{n}\log_2(n) - \frac{2}{n}. \tag{6.21}$$

- For a very low coding rate, we are using the quantum stabilizer code constructions derived from quadratic residues [259, 260]. By using simple linear regression, we arrive at

$$r_Q(n,\delta) = r(n,\frac{2}{n}+\frac{1}{4}) = \frac{1}{n}. \tag{6.22}$$

Table 6.4: The list of QSC constructions that are used to plot practical code in Fig. 6.4.

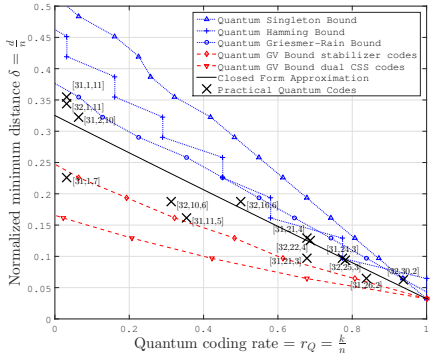
$\mathcal{C}[n,k,d]$ for $n = 31$ and $n = 32$	
QBCH [65]	[31,1,7], [31,11,5], [31,21,3]
QRM [67]	[32,10,6], [32,25,3]
QGF(4) [64]	[31,1,11], [31,2,10], [31,21,4], [31,26,2], [32,1,11], [32,16,6], [32,22,4], [32,25,3], [32,30,2]
$\mathcal{C}[n,k,d]$ for $n = 63$ and $n = 64$	
QBCH [65]	[63,27,7], [63,39,5], [63,45,4], [63,51,3], [63,57,2]
QRM [67]	[64,35,6], [64,56,3]
QGF(4) [64]	[63,51,4], [63,55,3], [63,60,2], [64,44,6], [64,48,5], [64,52,4], [64,56,3], [64,62,2]
$\mathcal{C}[n,k,d]$ for $n = 127$ and $n = 128$	
QBCH [65]	[127,1,19], [127,15,16], [127,29,15], [127,43,13], [127,57,11], [127,71,9], [127,85,7], [127,99,5], [127,113,3]
QRM [67]	[128,35,12], [128,91,6], [128,119,3]
QGF(4) [64]	[127,114,4], [127,118,3], [127,124,2], [128,105,6], [128,110,5], [128,114,4], [128,119,3], [128,126,2]

Using the three definitive points from the constructions given in Eq. (6.20), (6.21) and (6.22), we arrive at a system of three linear equations, which have a unique value of a, b and c for an arbitrary value of n. More explicitly, we have

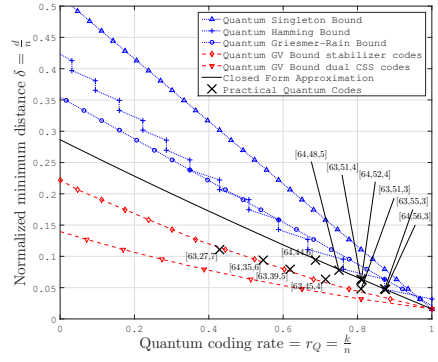
$$r_1 = a\delta_1^2 + b\delta_1 + c, \tag{6.23}$$

$$r_2 = a\delta_2^2 + b\delta_2 + c, \tag{6.24}$$

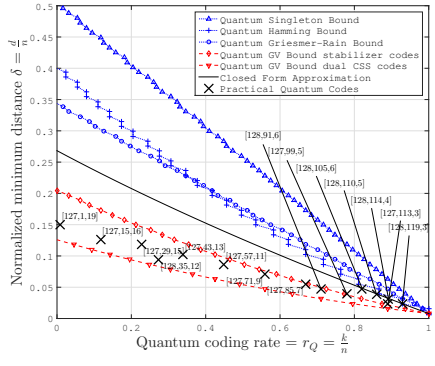
$$r_3 = a\delta_3^2 + b\delta_3 + c. \tag{6.25}$$



(a) $n = 31$ and $n = 32$.



(b) $n = 63$ and $n = 64$.



(c) $n = 127$ and $n = 128$.

Figure 6.4: Quantum coding rate r_Q versus normalized minimum distance δ for finite-length QSCs. The points for portraying the practical QSCs are taken from QBCH codes [65], QRM codes [67] and quantum codes from $GF(4)$ formulation [64]. ©Chandra *et al.* [204]

The analytical solution of Eq. (6.23), (6.24), and (6.25) is based on the following unique parameter values:

$$a = \frac{(r_3 - r_2)\,\delta_1 + (r_1 - r_3)\,\delta_2 + (r_2 - r_1)\,\delta_3}{(\delta_2 - \delta_1)\,(\delta_3 - \delta_2)\,(\delta_1 - \delta_3)}, \tag{6.26}$$

$$b = \frac{(r_2 - r_3)\,\delta_1^2 + (r_3 - r_1)\,\delta_2^2 + (r_1 - r_2)\,\delta_3^2}{(\delta_2 - \delta_1)\,(\delta_3 - \delta_2)\,(\delta_1 - \delta_3)}, \tag{6.27}$$

$$c = \frac{(r_3\delta_2 - r_2\delta_3)\,\delta_1^2 + (r_1\delta_3 - r_3\delta_1)\,\delta_2^2 + (r_2\delta_1 - r_1\delta_2)\,\delta_3^2}{(\delta_2 - \delta_1)\,(\delta_3 - \delta_2)\,(\delta_1 - \delta_3)}. \tag{6.28}$$

Despite the cluttered appearance of the analytical solution, it contains a simple closed-form approximation, because the value of r_1, r_2, r_3, δ_1, δ_2 and δ_3 may be readily calculated using Eq. (6.20), (6.21) and (6.22). Furthermore, the closed-form approximation derived for finite-length codewords has an inverse function of

$$\delta(n, r_Q) = \frac{-b - \sqrt{b^2 - 4a(c - r_Q)}}{2a}. \tag{6.29}$$

The accuracy of the proposed method is now tested for QSCs having codeword lengths of $n = \{31, 32, 63, 64, 127, 128\}$ as shown in Fig. 6.4. The list of practical QSC constructions which are used in these plots can be seen in Table. 6.4[1]. The closed-form approximation lies entirely between the upper and the lower quantum coding bounds. The practical QSCs are also plotted in the same figure to show the relative position with respect to the quantum coding bounds. The QSCs based on [64, 67] lay perfectly on the approximate curves, but it has been observed in [261] that as the codeword length increases and the quantum coding rate is reduced, the exact value of the minimum distance becomes unclear. As depicted in Fig. 6.4(b) and 6.4(c), we can hardly find definitive points associated with actual codes to plot in the low quantum coding-rate region constructed from quantum $GF(4)$. However, the QBCH code constructions lie quite close to the GV lower bound for dual-containing CSS codes. As predicted, since the constructions of QBCH codes rely on dual-containing CSS type constructions, which employ two separate PCMs for their stabilizer operators, we expect a lower coding rate compared to their non-CSS relatives.

The proposed closed-form approximation offers substantial benefits for the development of QSCs. We can readily find an approximation of the realistically achievable minimum distance for given code parameters. For instance, for half-rate quantum stabilizer codes of length 128, the minimum distance is bounded by $11 < d < 22$. By using our formulation, we obtain $d(n = 128, r_Q = 1/2) = 17$ from our finite-length approximation. Likewise, for half-rate quantum stabilizer codes of length 1024, the minimum distance is bounded by $78 < d < 157$. Using our method, we can obtain $d(n = 1024, r_Q = 1/2) = 103$ from our asymptotic bound approximation. One of the logical questions that may arise is concerned with the existence of the corresponding codes. For example, does a half-rate QSCs relying on $n = 128$ physical qubits and a minimum distance of $d = 16$ exist? The answer to this question is not definitive. Let us refer to the code table given in [261], which is mainly based on the QSC constructions of [64]. Due to space limitations, we are unable to capture the entire table and the associated PCM formulation. However, it is shown in [261] that a half-rate QSC relying on $n = 128$ physical qubits indeed exists, although the minimum distance is only loosely specified by the bounds of $11 < d < 20$. The bound is similar to the quantum GV bound and to the quantum Hamming bound of the minimum distance given by $11 < d < 22$. By contrast, upon using our approximation, we have a minimum distance of $d = 17$, which is again only an approximation and it does not imply the existence of a quantum code having a similar minimum distance. Nonetheless, we believe that our approximation is beneficial for approximating the attainable QBER performance of QSCs based on hard-decision syndrome decoding for short to moderate codeword length as follows (without considering degeneracy):

$$\mathrm{QBER}(n, d, p) = 1 - \sum_{i=0}^{t=\lfloor \frac{d-1}{2} \rfloor} \binom{n}{i} p^i (1-p)^{n-i}, \tag{6.30}$$

where the realistically achievable value of d is obtained from our approximation. In our view, the combination of our closed-form approximation and the QBER of Eq. (6.30) constitutes a useful benchmarker for the future development of QSCs, since it quantifies the realistically achievable QBER performance based on hard-decision syndrome-based decoding.

The evolution of our closed-form approximation as the codeword length increases for $n = \{31, 32, 63, 64, 127, 128\}$ can be seen in Fig. 6.5. By using our example, it can be clearly observed that as the codeword length increases, the approximation derived for finite-length codes slowly approaches the closed-form approximation of the asymptotic bound. However, inaccuracies emerge as the codeword length increases. This phenomenon emerges because we do not have definitive QSC constructions to rely on in the low coding rate region. In our approximation example, we are using the QSCs from quadratic residues based construction for the low coding rate region and the number of QSC constructions is limited to a handful codeword lengths. By contrast, in the low coding rate region of classical codes, we have the simple repetition codes relying on the construction $\mathcal{C}(n, 1)$ having a normalized minimum distance of $\delta(n, r) = \delta(n, \frac{1}{n}) = 1$.

Albeit the finite and infinite-length-based approximation curves start to deviate for a very long codeword $n \gg 100$, the minimum distance still grows with the codeword length, as portrayed in Fig. 6.6. Both the finite-

[1] A comprehensive list of practical quantum stabilizer codes can be found online at [261]. In this treatise, we only consider quantum stabilizer codes with a definitive minimum distance in the list.

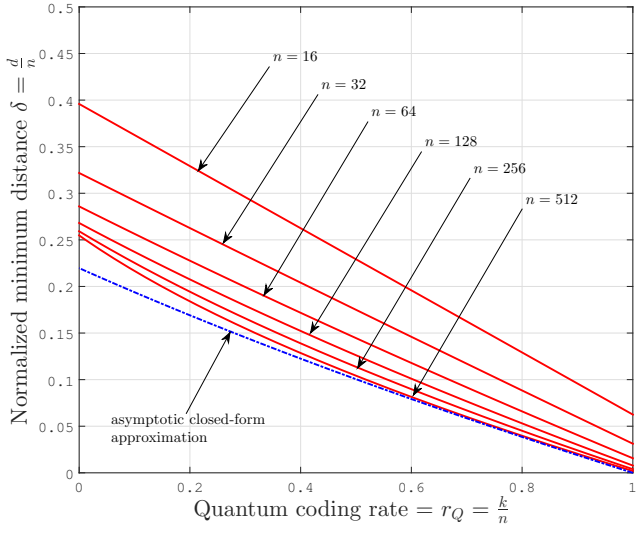


Figure 6.5: The evolution of our closed-form approximation for finite-length codewords parameterized by the codeword length n. ©Chandra *et al.* [204]

length approximation and asymptotic approximation follow the same trend. For $n \gg 100$, we can simply utilize the asymptotic formulation given in Eq. (6.17) for calculating the quantum coding rate for a certain desired minimum distance, or the inverse of the asymptotic formulation in Eq. (6.18) to determine the realistically achievable minimum distance, given the quantum coding rate. We may conclude from this figure that it is indeed possible to have a QSC construction associated with a growing minimum distance, as the codeword length increases.

6.5 The Bounds on Entanglement-Assisted Quantum Stabilizer Codes

One of the distinctive characteristics of quantum systems, which does not bear any resemblance with the classical domain is the ability of creating entanglement. This unique property can be exploited for increasing the achievable minimum distance of quantum codes, hence increasing the error correction capability of QSCs. The EA-QSC constructions are denoted by $\mathcal{C}(n, k, d; c)$, where c denotes the number of preshared entangled qubits. It is important to note that even though the EA-QSCs expand the Pauli group operators from $\mathcal{P}_n$ into $\mathcal{P}_{(n+c)}$, we only consider the error operators in $\mathcal{P}_n$. This is because the paradigm of EA-QSCs assumes that the preshared entangled qubits are not subjected to transmission error. Hence, for EA-QSCs, the quantum Hamming bound of Eq. (6.6) can be modified to

$$2^k \leq \frac{2^{n+c}}{\sum_{j=0}^{t=\lfloor \frac{d_{ea}-1}{2} \rfloor} \binom{n}{j} 3^j}, \tag{6.31}$$

where the notation d_{ea} denotes the minimum distance of EA-QSCs. Equation (6.31), can be rewritten to show explicitly the trade-off between the quantum coding rate r_Q and the minimum distance d_{ea} of EA-QSCs as

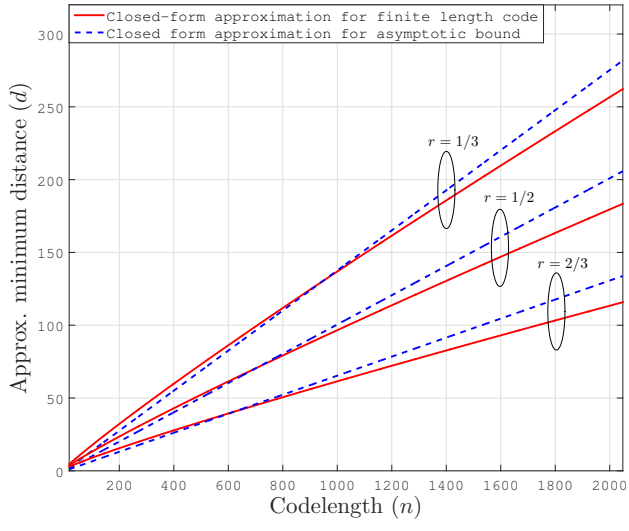


Figure 6.6: The growth of achievable minimum distance for short block QSCs versus the codeword length. ©Chandra *et al.* [204]

follows:

$$\frac{k}{n} \leq 1 - \frac{1}{n}\log_2\left(\sum_{j=0}^{t=\lfloor\frac{d_{ea}-1}{2}\rfloor}\binom{n}{j}3^j\right) + \left(\frac{c}{n}\right). \tag{6.32}$$

When n tends to ∞, we have

$$\frac{k}{n} \leq 1 - \left(\frac{d_{ea}}{2n}\right)\log_2 3 - H\left(\frac{d_{ea}}{2n}\right) + \left(\frac{c}{n}\right). \tag{6.33}$$

As encapsulated in Eq. (6.33), an additional conflicting parameter is involved in determining the quantum coding bounds, namely the entanglement consumption rate. The entanglement consumption rate E is the ratio between the number of preshared maximally entangled qubits c to the number of physical qubits n as encapsulated below:

$$E = \frac{c}{n}. \tag{6.34}$$

A maximally entangled[2] QSCs requires $c = n - k$ preshared qubit pairs. Hence, for a maximally entangled QSCs, the quantum Hamming bound of Eq. (6.33) can be reformulated as follows by substituting $c = n - k$ into Eq. (6.33), yielding:

$$\frac{k}{n} \leq 1 - \frac{1}{2}\left(\left(\frac{d_{ea}}{2n}\right)\log_2 3 - H\left(\frac{d_{ea}}{2n}\right)\right). \tag{6.35}$$

Let us now consider the more general cases, where we may have a range of different entanglement ratios $0 \leq \theta \leq 1$. The entanglement ratio is defined as the ratio of the number of preshared qubits c to that of the maximally-entangled preshared qubits $(n - k)$, yielding:

$$\theta = \frac{c}{n-k}. \tag{6.36}$$

[2]For maximally-entangled QSCs, all of the auxiliary qubits required to generate the encoded state are already preshared using maximally entangled qubit paies. Hence, the maximal number of entangled qubits that can be shared beforehand is equal to the total number of auxiliary qubits, which is equal to $(n - k)$

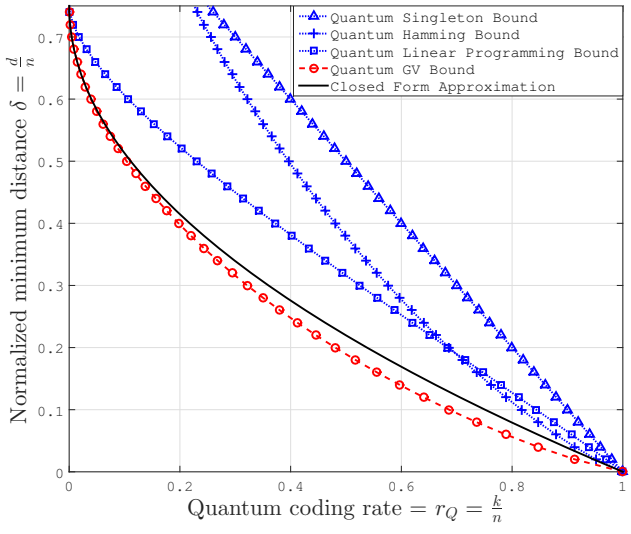


Figure 6.7: The asymptotic quantum coding bounds on EA-QSCs for maximally-entangled constructions. A simple quadratic function $r(\delta) = \frac{16}{9}\delta^2 - \frac{8}{3}\delta + 1$ satisfies all of the quantum coding bounds. ©Chandra *et al.* [204]

The quantum Hamming bound for EA-QSCs with arbitrary entanglement ratios of θ is given by

$$\frac{k}{n} \leq 1 - \frac{1}{1+\theta}\left(\left(\frac{d_{ea}}{2n}\right)\log_2 3 - H\left(\frac{d_{ea}}{2n}\right)\right). \tag{6.37}$$

Table 6.5: The entanglement-assisted quantum coding bounds found in the literature.

Quantum Bound	Finite-Length	Asymptotic	Notes
Singleton [69]	$\frac{k}{n} \leq 1 - 2\left(\frac{d_{ea}-1}{n}\right) + \left(\frac{c}{n}\right)$	$\frac{k}{n} \leq 1 - 2\left(\frac{d_{ea}}{n}\right) + \left(\frac{c}{n}\right)$	very loose upper bound
Hamming [68]	$\frac{k}{n} \leq 1 - \frac{1}{n}\log_2\left(\sum_{j=0}^{t=\lfloor\frac{d_{ea}-1}{2}\rfloor}\binom{n}{j}3^j\right) + \left(\frac{c}{n}\right)$	$\frac{k}{n} \leq 1 - \left(\frac{d_{ea}}{2n}\right)\log_2 3 - H\left(\frac{d_{ea}}{2n}\right) + \left(\frac{c}{n}\right)$	tight upper bound
Linear Progr. [262]		$\frac{k}{n} \leq H(\tau) + \tau\log_2 3 - 1 + \left(\frac{c}{n}\right)$ $\tau = \frac{3}{4} - \frac{1}{2}\delta - \frac{1}{2}\sqrt{3\delta(1-\delta)}$	strengthen the upper bound
Plotkin [263, 264]	$\frac{d_{ea}}{n} \leq \frac{3\left(4^k\right)}{8\left(4^k-1\right)}\left(1 + \left(\frac{k}{n}\right) + \left(\frac{c}{n}\right)\right)$	$\frac{d_{ea}}{n} \leq \frac{3}{8}\left(1 + \left(\frac{k}{n}\right) + \left(\frac{c}{n}\right)\right)$	upper bound for minimum distance
GV [264]	$\frac{k}{n} \geq 1 - \frac{1}{n}\log_2\left(\sum_{j=0}^{d_{ea}-1}\binom{n}{j}3^j\right) + \left(\frac{c}{n}\right)$	$\frac{k}{n} \geq 1 - \left(\frac{d_{ea}}{n}\right)\log_2 3 - H\left(\frac{d_{ea}}{n}\right) + \left(\frac{c}{n}\right)$	tight lower bound

The rest of the quantum coding bounds can be readily derived using similar procedures. The resultant

Table 6.6: The asymptotic quantum coding bounds for EA-QSCs given the arbitrary entanglement ratios of θ. ©Chandra *et al.* [204]

Quantum Bound	**Entanglement Ratio** $= \theta$	**Maximally Entangled** $(\theta = 1)$
Singleton [69]	$\frac{k}{n} \leq 1 - \left(\frac{2}{1+\theta}\right)\left(\frac{d_{ea}}{n}\right)$	$\frac{k}{n} \leq 1 - \left(\frac{d_{ea}}{n}\right)$
Hamming [68]	$\frac{k}{n} \leq 1 - \frac{1}{1+\theta}\left(\left(\frac{d_{ea}}{2n}\right)\log_2 3 - H\left(\frac{d_{ea}}{2n}\right)\right)$	$\frac{k}{n} \leq 1 - \frac{1}{2}\left(\left(\frac{d_{ea}}{2n}\right)\log_2 3 - H\left(\frac{d_{ea}}{2n}\right)\right)$
Linear Programming [262]	$\frac{k}{n} \leq \frac{1}{1+\theta}\left(H(\tau) + \tau\log_2 3 - 1 + \theta\right)$	$\frac{k}{n} \leq \frac{1}{2}\left(H(\tau) + \tau\log_2 3\right)$
Plotkin [263, 264]	$\frac{d_{ea}}{n} \leq \frac{3}{8}\left(1 + \theta + \frac{k}{n}(1-\theta)\right)$	$\frac{d_{ea}}{n} \leq \frac{3}{4}$
GV [264]	$\frac{k}{n} \geq 1 - \frac{1}{1+\theta}\left(\left(\frac{d_{ea}}{n}\right)\log_2 3 - H\left(\frac{d_{ea}}{n}\right)\right)$	$\frac{k}{n} \geq 1 - \frac{1}{2}\left(\left(\frac{d_{ea}}{n}\right)\log_2 3 - H\left(\frac{d_{ea}}{n}\right)\right)$

entanglement-assisted quantum coding bounds are portrayed in Fig. 6.7 and 6.8. By substituting the entanglement ratio of $\theta = 0$, we arrive again at the quantum coding bounds derived for unassisted QSCs. By contrast, upon substituting the entanglement ratio of $\theta = 1$ into Eq. (6.37), we arrive at the quantum coding bounds of maximally-entangled QSCs. Figure 6.7 portrays the bounds on maximally-entangled QSCs. It is observed in Fig. 6.7 that at the point $(\delta = 0.75)$, the quantum GV bound (lower bound) intersects the quantum linear programming bound (upper bound). Indeed, it is confirmed by the quantum Plotkin bound of the maximally-entangled QSC constructions shown in Table 6.6 that for asymptotical maximally-entangled QSCs the highest normalized minimum distance that can be achieved is $\delta = 0.75$. Hence, based on this observation, we propose a simple quadratic function as the closed-form approximation of entanglement-assisted quantum stabilizer codes that will satisfy all of the well-known bounds. A quadratic function associated with a symmetry line at $(\delta = 0.75)$ and crossing the point of $(\delta, r) = (0, 1)$ is given by

$$r_Q(\delta) = \frac{16}{9}\delta^2 - \frac{8}{3}\delta + 1 \text{ for } 0 \leq \delta \leq 0.75. \tag{6.38}$$

The simple quadratic approximaton given in Eq. (6.38), can also be inverted, yielding

$$\delta(r_Q) = \frac{3}{4}(1 - \sqrt{r_Q}) \text{ for } 0 \leq r_Q \leq 1. \tag{6.39}$$

From the simple quadratic function in Eq. (6.38), we can also derive a simple closed-form approximation for a given arbitrary entanglement ratio of $0 \leq \theta \leq 1$, as shown below:

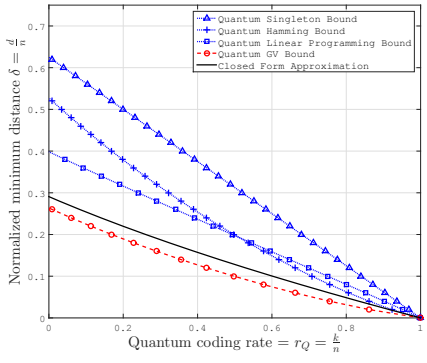
$$r_Q(\delta) = \frac{1}{1+\theta}\left(\frac{32}{9}\delta^2 - \frac{16}{3}\delta + 1 + \theta\right), \tag{6.40}$$

for $0 \leq \delta \leq \frac{3}{4}\left(1 - \sqrt{\frac{1-\theta}{2}}\right)$ and $0 \leq \theta \leq 1$. The expression given in Eq. (6.40) may be inverted to arrive at the following equation:

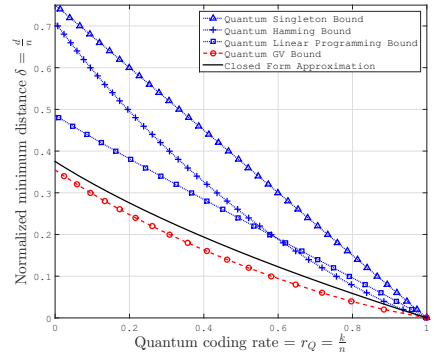
$$\delta(r_Q) = \frac{3(\sqrt{2} - \sqrt{r_Q(1+\theta) + (1-\theta)}}{4\sqrt{2}}, \tag{6.41}$$

for $0 \leq r_Q \leq 1$ and $0 \leq \theta \leq 1$.

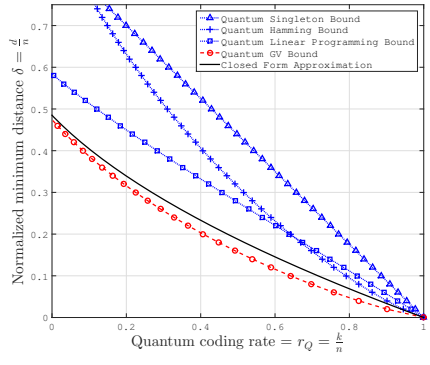
The simple closed-form approximation given in Eq. (6.40) and (6.41) satisfies all entanglement-assisted quantum coding bounds for arbitrary entanglement ratios, as confirmed by Fig. 6.8. We should point out at this stage that as we substitute the value of $\theta = 0$ into Eq. (6.40) and (6.41), we arrive at the closed-form approximation presented in the Eq. (6.17) and (6.18) for unassisted asymptotic quantum coding bounds. Hence, we completed our closed-form approximations conceived for all of the different constructions of quantum stabilizer codes.



(a) $\theta = 0.25$.



(b) $\theta = 0.50$.



(c) $\theta = 0.75$.

Figure 6.8: The asymptotic quantum coding bounds on EA-QSCs for entanglement ratios $\theta = \{0.25, 0.50, 0.75\}$. ©Chandra *et al.* [204]

6.6 Summary and Conclusions

We have conducted a survey of quantum coding bounds, which describe the trade-off between the quantum coding rate and the error correction capability of a wide range of QSC constructions. Furthermore, we provided insights into their relationships with their classical counterparts. For the family of unassisted QSCs, we have provided both lower and upper bounds for both CSS and non-CSS code constructions. For the EA-QSCs, we have presented the quantum coding bounds for maximally-entangled constructions and also for arbitrary entanglement ratios.

We have also proposed a closed-form approximation as a beneficial tool for analyzing the performance of QSCs. The resultant closed-form approximation may indeed be used as a simple benchmark for developing QSCs, because the resultant minimum distance δ and quantum coding rate r_Q values from our approximations are unambiguous. For instance, for a half-rate quantum stabilizer code having a given codeword length of $n = 128$, the minimum distance is bounded by $11 < d < 22$. Upon using our approximation, we arrive at $d(n = 128, r_Q = 1/2) = 16$ from our finite-length approximation. Likewise, for a half-rate quantum stabilizer code having the codeword length of 1024, the minimum distance is bounded by $78 < d < 157$. By using our proposal, we have an approximate minimum distance of $d(n = 1024, r_Q = 1/2) = 103$ from our asymptotic bound approximation. Ultimately, the proposed method can be utilized as an efficient tool for the

characterization of quantum stabilizer codes.

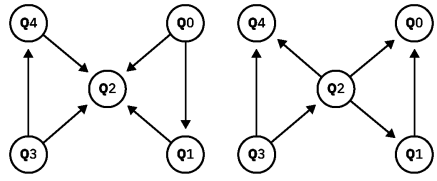
Chapter 7

Quantum Topological Error Correction Codes: The Classical-to-Quantum Isomorphism Perspective

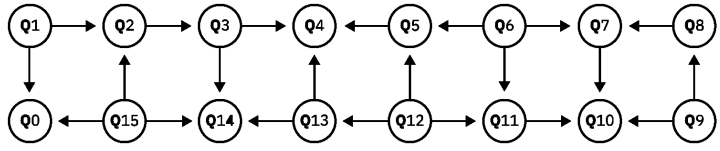
7.1 Introduction

In Chapter 3 and Chapter 6, we have demonstrated that classical error correction codes can be transformed into their quantum domain counterparts and we can readily characterize their associated QBER performances by exploiting the classical-to-quantum isomorphism as a function of their code parameters. However, the real physical implementations of quantum computers impose additional challenges that prevent us from directly transplanting QSCs. Let us observe the layout of the physical qubits in some of IBM's quantum computers in Fig. 7.1. In these figures, the circles represent the qubits and the lines with the arrows represent the interactions among the qubits. Observe that any quantum computation task performed by the quantum computers is designed based on local qubit interactions. However, most of the QSCs derived from potent classical error correction codes may require interactions between distant qubits of a codeword, but arranging for this remains unrealistic at the current state-of-the-art. This particular constraint motivated the invention of new families of short QSCs, which were specifically designed for circumventing the above-mentioned limitation of local interactions. These codes belong to the family of QTECCs. The stabilizer operators of QTECCs can be explicitly defined by the underlying lattice structure accommodating the qubits. Following the same line of investigation as in Chapter 6, we will characterize the QBER performance of diverse QTECCs versus the code parameters given a certain lattice structure by exploiting the classical-to-quantum isomorphism.

The rest of this chapter is organized as follows. In Section 7.2, we commence with design examples of classical TECCs to pave the way for delving into the quantum domain. In Section 7.3, we detail our QSC design examples for QTECCs. We will continue by characterizing the QBER performance of QTECCs subjected to



(a) 5 qubits (ibmqx2). (b) 5 qubits (ibmqx4).



(c) 16 qubits (ibmqx5).

Figure 7.1: The qubit arrangement of IBM's superconducting quantum computers. The circles represent the qubits, while the arrows represent the possible qubit interactions within the computers [99].

the quantum depolarizing channel, their distance from the hashing bound, and fidelity in Section 7.4. Finally, we conclude our discussions in Section 7.5.

7.2 Classical Topological Error Correction Codes: Design Examples

Classical error correction codes can be developed relying on diverse approaches [265]. We can find in the literature various family of codes based on algebraic formalisms, such as BCH codes and RS codes, codes based on conventional trellis structures, such as convolutional codes and turbo codes, codes based on bipartite graphs such as LDPC codes, and codes based on channel polarization, such as polar codes. Another approach that can be adopted to formulate a classical error correction code is by exploiting the topological or lattice structure. By mapping the bits of a codeword to a lattice structure we can create an error correction scheme [266]. For instance, let us assume that a codeword of classical bits is arranged on the square lattice given in Fig. 7.2. The black circles on the edges of the lattice define the encoded information bits, namely the codeword. The red squares laying on the vertices of the lattice define the PCM of the code, which also directly defines the syndrome values of the received codeword. The number of black circles is associated with the codeword length of n bits and the number of red squares is associated with the length of the syndrome vector or the number of rows in the PCM, which is equal to $(n-k)$ bits. For this particular square lattice seen in Fig. 7.2, the codeword length n is equal to 13 bits and the length $(n-k)$ of the redundant part is 6. Hence, the number of information bits k is equal to 7. Therefore, this code has $2^7 = 128$ legitimate codewords out of the theoretically possible $2^{13} = 8192$ received words. In conceptually simple terms one could construct a look-up table for implementing the corresponding decoder, which has to find the specific legitimate codeword that has the lowest Hamming distance from the corrupted word.

For the sake of comparison, we could consider a classical BCH code having n=15, k=7 and d=5, which is capable of correcting two classical bit errors in a 15-bit codeword. In general, BCH codes do not have to satisfy the above-mentioned topological constraint of only having localized interactions and this design flexibility typically results in a higher error correction capability. This classical BCH code has 2^7=128 legitimate codewords and 2^{15}=32 768 possible words. With the aid of its 8 parity bits we would be able to distinguish

$2^{(15-7)} = 2^8 = 256$ distinct error patterns (including the error-free scenario) and correct two arbitrary bit errors.

The coding rate r is defined by the ratio between the information bits k and the codeword length n, yielding:

$$r = \frac{k}{n} \tag{7.1}$$

Hence, the coding rate of the square lattice code of Fig. 7.2 is $r = 7/13$.

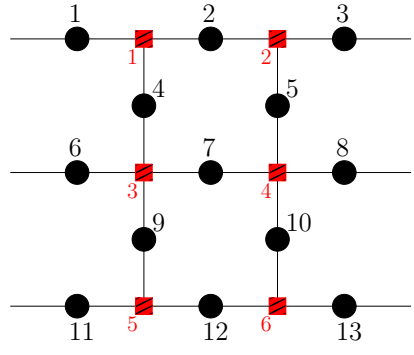


Figure 7.2: Example of a classical bit arrangement on a square lattice structure. The black circles laying on the edges of the lattice denote the bits of the codeword, while the vertices of the lattice denoted by red squares define the PCM and also the syndrome values. ©Chandra *et al.* [174]

Now, let us delve deeper into how the error correction works. Let us revisit the square lattice of Fig. 7.2. The k information bits are encoded to n-bit codewords, where $n > k$. Noise or decoherence imposed by the channel corrupts the legitimate codeword. The syndrome computation is invoked to generate the $(n-k)$-bit syndrome vector, which tells us both the predicted number and the position of the errors. In Fig. 7.2, each of the red squares indicates a syndrome bit of s_i. Hence, the syndrome vector $\mathbf{s}$ is a 6-bit vector, which is given by

$$\mathbf{s} = [s_1 \ s_2 \ s_3 \ s_4 \ s_5 \ s_6]. \tag{7.2}$$

In the case of an error-free received codeword, the resultant syndrome vector is $\mathbf{s} = [0\ 0\ 0\ 0\ 0\ 0]$. By contrast, if an error is imposed on the codeword, it triggers a syndrome bit value of 1 at the adjacent syndrome bit positions. For example, if an error occurs at the bit index 4 of Fig. 7.2, it triggers the syndrome values of $s_1 = 1$ and $s_3 = 1$. The rest of the syndrome values remain equal to 0. Therefore, an error corrupting the bit index 4 generates a syndrome vector of $\mathbf{s} = [1\ 0\ 1\ 0\ 0\ 0]$. Hence, the decoder flips the value of bit index 4. Similarly, if an error occurs at bit number 3, it only triggers the syndrome value of $s_2 = 1$. Hence, it generates the syndrome vector of $\mathbf{s} = [0\ 1\ 0\ 0\ 0\ 0]$ and the error recovery procedure proceeds accordingly.

Now let us consider the ocurrence of two bit errors in the codeword. For instance, let us assume that errors occur at bit indices of 6 and 7 of Fig. 7.2. Note that both these errors affect s_3, therefore they cancel each other's effect on s_3, hence generating a syndrome bit value of $s_3 = 0$. However, we still do not receive an all-zero syndrome vector, because the bit index 7 results in the syndrome bit value of $s_4 = 1$ in Fig. 7.2. Therefore, the resultant syndrome vector due to a bit error in both bit 6 and 7 is $\mathbf{s} = [0\ 0\ 0\ 1\ 0\ 0]$. Since the syndrome vector of $\mathbf{s} = [0\ 0\ 0\ 1\ 0\ 0]$ is also associated with the error incident upon bit index 8, the error recovery procedure decides to flip bit 8 instead, because for the error probability less than 1/2, a single error occurance is more likely to happen than a double-error. As a result, we end up having three errors. This example is an illustration that the occurence of two bit errors in the codeword is beyond the error correction capability of the code given in Fig. 7.2. We conclude that the code based on the square lattice illustrated in Fig. 7.2 is capable of correcting only a single bit error. The error correction capability of t bits for a given code

construction is defined by the minimum distance d of the code as formulated by

$$t = \left\lfloor \frac{d-1}{2} \right\rfloor. \tag{7.3}$$

Hence, a code that is only capable of correcting a single error has a minimum distance of $d = 3$, as exemplified by the square lattice code given in Fig. 7.2. Moreover, the minimum distance of a square lattice code is defined by dimension of the lattice. Therefore, to increase the error correction capability of the code, we can simply increase the dimension of the lattice, which directly translates into increasing the minimum distance. The square lattice considered in our example can be generalized to a rectangular lattice structure having a dimension of $(l \times h)$, where l is the length of the lattice and h is the height of the lattice. In the case of a rectangular structure, the minimum distance is defined by

$$d = \min(l, h). \tag{7.4}$$

The codeword length is also uniquely defined by the dimension of the lattice. More explicitly, for a rectangular lattice of dimension $(l \times h)$, the codeword length is equal to the number of the lattice edges, which is given by

$$n\text{-edges} = n_{\text{square}} = 2lh - l - h + 1. \tag{7.5}$$

The number of rows in the PCM of a square lattice is defined by the number of faces or plaquettes of the rectangular lattice, which is formulated as follows:

$$n\text{-vertices} = n_{\text{square}} - k_{\text{square}} = h(l-1). \tag{7.6}$$

Hence, from Eq. (7.5) and (7.6), the number of information bits k encoded by the rectangular lattice codes is

$$\begin{aligned} k_{\text{square}} &= n_{\text{square}} - (n_{\text{square}} - k_{\text{square}}) \\ &= lh - l + 1. \end{aligned} \tag{7.7}$$

The most efficient code can be constructed by a square lattice, where $d = l = h$. Therefore, the expression given in Eq. (7.5) and (7.7) can be simplified to

$$n_{\text{square}} = 2d^2 - 2d + 1 \tag{7.8}$$

$$k_{\text{square}} = d^2 - d + 1. \tag{7.9}$$

Hence, the coding rate of square lattice based codes can be formulated as follows:

$$r_{\text{square}} = \frac{k_{\text{square}}}{n_{\text{square}}} = \frac{d^2 - d + 1}{2d^2 - 2d + 1}. \tag{7.10}$$

The PCM can be readily constructed in a similar fashion. Each red square of Fig. 7.2 represents the row of the PCM, where the adjacent black circles denote the index of the column containing a value of 1. For example, the first red square is adjacent to the black circles numbered 1, 2, and 4. Therefore, in the first row of the PCM, there are only three elements containing a value of 1 and those are marked by the index 1, 2, and 4. The remaining rows of the PCM are generated using the same principle. Explicitly, each row of the PCM of the square lattice code of Fig. 7.2 is portrayed in Table 7.1. Finally, the PCM $\mathbf{H}$ of the square lattice code of Fig. 7.2 is given by

$$\mathbf{H} = \begin{bmatrix} \mathbf{h}_1 & \mathbf{h}_2 & \mathbf{h}_3 & \mathbf{h}_4 & \mathbf{h}_5 & \mathbf{h}_6 \end{bmatrix}^T. \tag{7.11}$$

The code construction based on the general lattice structure is not limited to a rectangular lattice. Let us consider, for instance, the triangular lattice of Fig. 7.3. The black circles laying on the vertex of the lattice

Table 7.1: Constructing the PCM of the square lattice code of Fig. 7.2 with minimum distance of $d = 3$. Each row is associated with the syndrome operators denoted by red squares in Fig. 7.2

	1	2	3	4	5	6	7	8	9	10	11	12	13
$\mathbf{h}_1$	1	1	0	1	0	0	0	0	0	0	0	0	0
$\mathbf{h}_2$	0	1	1	0	1	0	0	0	0	0	0	0	0
$\mathbf{h}_3$	0	0	0	1	0	1	1	0	1	0	0	0	0
$\mathbf{h}_4$	0	0	0	0	1	0	1	1	0	1	1	0	0
$\mathbf{h}_5$	0	0	0	0	0	0	0	0	1	0	1	1	0
$\mathbf{h}_6$	0	0	0	0	0	0	0	1	0	1	0	0	1

Table 7.2: Constructing the PCM of the triangular lattice code with minimum distance of $d = 3$. Each row is associated with the syndrome operators denoted by blue circles in Fig. 7.3

	1	2	3	4	5	6	7
$\mathbf{h}_1$	1	1	1	1	0	0	0
$\mathbf{h}_2$	0	0	1	1	1	1	0
$\mathbf{h}_3$	0	1	0	1	0	1	1

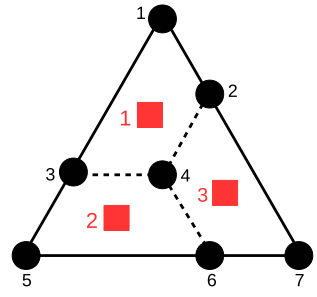


Figure 7.3: Example of a classical bit arrangement constructed over a triangular lattice structure. The black circles laying on the vertices of the lattice represent the codeword bits, while the faces or the plaquettes of the lattice denoted by red squares define the parity-check matrix and the syndrome bits of the error correction code. This configuration bears a resemblance to the $\mathcal{C}(7, 4)$ classical Hamming code. ©Chandra *et al.* [174]

define the codeword and the red squares on the faces of the lattice define the syndrome vector. The error correction principle of the triangular lattice code is similar to that of its square counterpart. Hence, the PCM of the triangular lattice code is readily derived using the following equation:

$$\mathbf{H} = \begin{bmatrix} \mathbf{h}_1 \ \mathbf{h}_2 \ \mathbf{h}_3 \end{bmatrix}^T, \tag{7.12}$$

where $\mathbf{h}_1$, $\mathbf{h}_2$, and $\mathbf{h}_3$ correspond to the syndrome bits given in Table 7.2. It is important to point out that the resultant triangular lattice code bears a strong resemblance to the classical $\mathcal{C}(7, 4, 3)$ Hamming code. Specifically, both codes have a codeword length of $n = 7$ and the number of information bits is $k = 4$. Hence,

the length of the syndrome vector is 3 bits. Consequently, the codes have 2^4 legitimate codewords out of the 2^7 possible received words. Based on the sphere packing bound, the codes are capable of distinguishing $2^3 = 8$ distinct error patterns including the error-free scenario. Therefore, both constructions are capable of correcting exactly a single error with an identical coding rate of $r = 4/7$.

Similar to its rectangular counterpart, increasing the error correction capability of a triangular lattice code is achieved by expanding the underlying lattice configuration. However, increasing the number of vertices of the triangular lattice structure is not as straightforward as that of its rectangular counterpart because it can be carried out in several different ways. In this example, we use the construction proposed in [105] and Fig. 7.4 illustrates how to increase the number of encoded bits of the triangular lattice code of Fig. 7.3 by using hexagonal tiles.

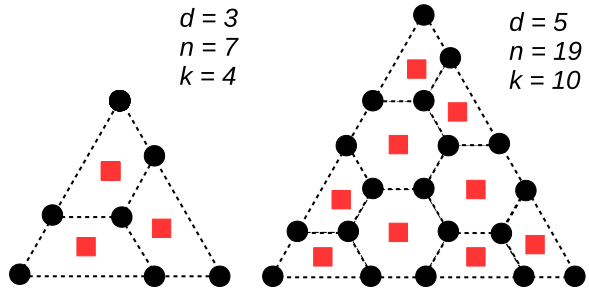


Figure 7.4: Extending the length of the triangular lattice code, which directly increases the numbers of error corrected. ©Chandra *et al.* [174]

Following the pattern of Fig. 7.4, the codeword length, which is also given by the number of vertices of the given lattices, is explicitly formulated as follows:

$$n\text{-vertices} = n_{\text{triangular}} = \frac{1}{4}(3d^2 + 1). \tag{7.13}$$

Hence for d=3 we have 7 and for d=5 we have 19 verticies. The number of faces in the triangular lattice, which corresponds to the number of rows of the PCM and also to the syndrome vector length, can be expressed as

$$n\text{-faces} = n_{\text{triangular}} - k_{\text{triangular}} = \frac{1}{8}(3d^2 - 3). \tag{7.14}$$

Hence, the number of information bits can be expressed as

$$\begin{aligned} k_{\text{triangular}} &= n_{\text{triangular}} - (n-k)_{\text{triangular}} \\ &= \frac{1}{8}(3d^2 + 5). \end{aligned} \tag{7.15}$$

Finally, the coding rate of the triangular lattice codes of Fig. 7.4 is formulated as follows:

$$r_{\text{triangular}} = \frac{k_{\text{triangular}}}{n_{\text{triangular}}} = \frac{3d^2 + 5}{2(3d^2 + 1)}. \tag{7.16}$$

Then, the normalized minimum distance, which directly correponds to the error correction capability per-bit of a code may be defined as:

$$\delta = \frac{d}{n} \tag{7.17}$$

Table 7.3: Code parameters of classical Hamming code having a single error correction capability, which is used in Fig. 7.6 and 7.7. The coding rate r and normalized mnimum distance δ is calculated using Eq. (7.1) and (7.17), respectively.

n	k	d	n	k	d
3	1	3	127	120	3
7	4	3	255	247	3
15	11	3	511	502	3
31	26	3	1023	1013	3
63	57	3	...	...	...

For square lattice and triangular lattice codes, the normalized minimum distances are given by

$$\begin{aligned} \delta_{\text{square}} &= \frac{d}{2d^2 - 2d + 1} \\ \delta_{\text{triangular}} &= \frac{4d}{3d^2 + 1}. \end{aligned} \tag{7.18}$$

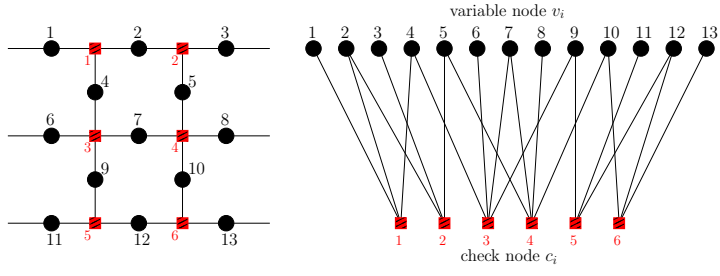


Figure 7.5: Example of how to represent the square lattice code. (a) The representation in lattice structure. (b) The representation in Tanner or bipartite graph. ©Chandra *et al.* [174]

In the rest of this treatise, we will consider the family of error correction codes based on lattice structures as a prominent representative of classical topological error correction codes (TECC). The lattice structures given in Fig. 7.2 and 7.3 can be transformed to Tanner graphs [267]. The dual representation of TECCs in the rectangular lattice domain and in the Tanner graph domain is given in Fig. 7.5 as exemplified by the square lattice code. We can observe that TECCs based on square lattices have a maximum row weight of $\rho_{\max} = 4$ and a maximum column weight of $\gamma_{\max} = 2$. By contrast, the codes based on triangular lattices have $\rho_{\max} = 6$ and $\gamma_{\max} = 3$. For a very long codeword, these properties lead to sparse PCMs. Hence, classical TECCs can be viewed as a specific family of LDPC codes. The asymptotical limit of the coding rate for LDPC codes based on TECCs can be directly derived from Eq. (7.10) and (7.16). As the codeword length tends to infinity ($n \to \infty$), the minimum distance d is also expected to tend to infinity. Hence, for the asymptotical limit we have

$$r^{\infty}_{\text{square}} = \lim_{d \to \infty} \frac{d^2 - d + 1}{2d^2 - 2d + 1} = \frac{1}{2}, \tag{7.19}$$

$$r^{\infty}_{\text{triangular}} = \lim_{d \to \infty} \frac{3d^2 + 5}{2(3d^2 + 1)} = \frac{1}{2}. \tag{7.20}$$

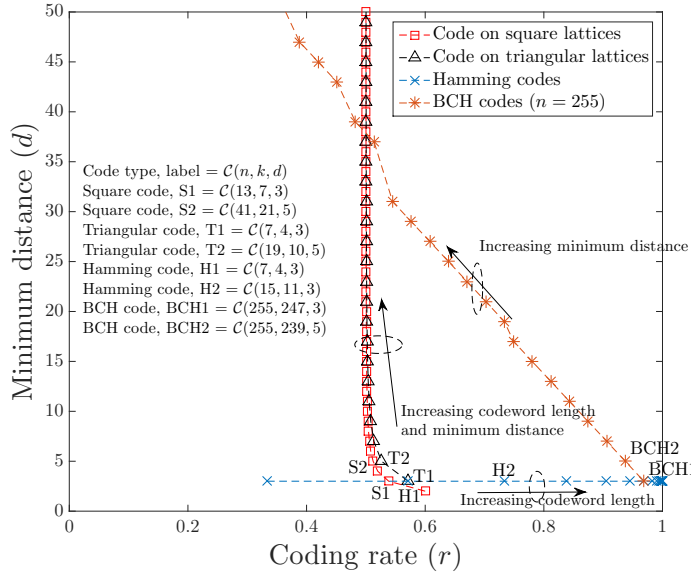


Figure 7.6: The coding rate versus minimum distance of TECCs. For asymptotical limit, the TECCs may be categorized into LPDC codes and the coding rates converge to $r = \frac{1}{2}$. We also include the BCH codes and Hamming codes for the sake of comparison. The coding rate for the square lattice based codes and the triangular lattice based codes are defined in Eq. (7.10) and (7.16), respectively. The code parameters for classical Hamming and BCH codes are described in Table 7.3 and 7.4, respectively. We put labels only for several codes as examples on how to convert the given code parameters into the figure. ©Chandra *et al.* [174]

Let us observe Fig. 7.6, where we plot the minimum distance (d) versus coding rate (r) of TECCs based on Eq. (7.10) and (7.16). We also include the classical codes based on the sphere packing concept, namely the Hamming codes and the BCH codes, whose parameters are portrayed in Table 7.3 and 7.4, respectively. Additionally, we also include some labels for several codes in the figure, in order to show how to convert the code parameters into constellation points in the figure. More explicitly, let us consider the specific triangular codes T1 and T2, where T1 represents the triangular code having a minimum distance of 3, which we have already used in the example in Fig. 7.3. As it has been elaborated on earlier, the resultant code T1 is $\mathcal{C}(7, 4, 3)$. Hence, the coding rate is $r = 4/7 \approx 0.57$. Again, the triangular code T1 has identical code parameters to the Hamming code $\mathcal{C}(7, 4, 3)$, which is labelled H1. Hence, the same point in Fig. 7.6 represents both T1 and H1. Next, the code parameters of the triangular code T2 having a minimum distance of $d = 5$ are obtained using Eq. (7.13) and (7.15) for determining the codeword length n and the information length k, respectively. Explicitly, by substituting $d = 5$ into Eq. (7.13) and (7.15), we have $n = 19$ and $k = 10$. Finally, we arrive at the coding rate of $r = k/n = 10/19 \approx 0.53$ for the triangular code T2. The rest of the code parameters for square codes, triangular codes, Hamming codes and BCH codes are portrayed in the same way in Fig. 7.6.

In general, increasing the minimum distance of the codes while maintaining the codeword length can be achieved at the expense of reducing the coding rate. This phenomenon is perfectly reflected by the behaviour of classical BCH codes in Fig. 7.6. Explicitly, in Fig. 7.6 we portray BCH codes having a constant codeword length of $n = 255$, which are described in Table 7.4. As seen, upon increasing the minimum distance of BCH codes, the coding rate is gradually reduced. Next, increasing the coding rate while maintaining the minimum distance of the code can indeed be achieved by increasing the codeword length. In this case, the Hamming codes, whose code parameters are described in Table 7.3, perfectly reflect this phenomenon. Observe in Fig. 7.6,

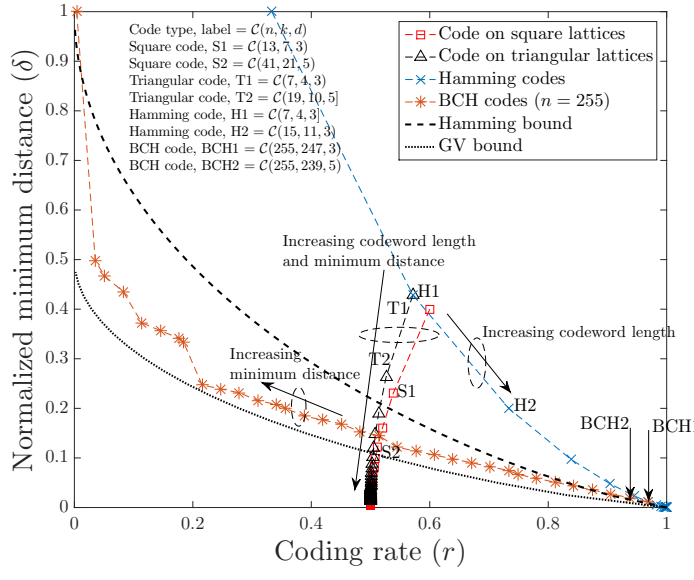


Figure 7.7: The coding rate versus normalized minimum distance of TECCs. For asymptotical limit, the TECCs may be categorized into LPDC codes and the coding rates converge to $r = \frac{1}{2}$, while the normalized minimum distances (δ) vanish to zero. In addition, we also include the classical Hamming and BCH codes, which constructed based on sphere packing bound, for the sake of comparison. The code parameters for classical Hamming and BCH codes are portrayed in Table 7.3 and 7.4, respectively. We put labels only for several codes as examples on how to convert the given code parameters into the figure. ©Chandra *et al.* [174]

that for the Hamming codes exhibiting a constant minimum distance of $d = 3$, we can see the gradual increase of coding rate upon increasing the codeword length. However, the behaviour of the BCH and Hamming codes is not reflected by the TECCs. Let us elaborate on the TECCs behaviour in Fig. 7.6. The increase of the minimum distance of TECCs upon increasing the codeword length looks very impressive since they do not seem to require much sacrifice in terms of coding rate reduction. In fact, the coding rate is saturated at approximately $r = 1/2$ for long codewords. This is indeed a rather different behaviour compared to that of the classical BCH codes. However, it is of pivotal importance to mention again that the increasing error correction capability per codeword does not necessarily imply the improvement of error correction capability per bit. Therefore, we have to normalize the performance to the codeword length for a fair comparison.

Let us now observe Fig. 7.7, where we plot the normalized minimum distance (δ) versus the coding rate (r) of TECCs based on Eq. (7.18). We include both the BCH codes as well as the Hamming codes for the sake of comparison. We also plot the classical Hamming bound [253] and GV bound [254] in this figure to portray the upper bound and lower bound of the normalized minimum distance, which correspond directly to the normalized error correction capability, given the coding rate. The classical Hamming bound is formulated as follows [253]:

$$\frac{k}{n} \leq 1 - H\left(\frac{d}{2n}\right), \tag{7.21}$$

where $H(x)$ is the binary entropy of x defined by $H(x) = -x \log_2 x - (1-x) \log_2 (1-x)$, while the classical GV bound is expressed as [254]

$$\frac{k}{n} \geq 1 - H\left(\frac{d}{n}\right). \tag{7.22}$$

Table 7.4: Code parameters of classical BCH codes having codeword length of $n = 255$, which is used in Fig. 7.6 and 7.7. The coding rate r and normalized mnimum distance δ is calculated using Eq. (7.1) and (7.17), respectively.

n	k	d	n	k	d	n	k	d
255	1	255	255	87	53	255	171	23
255	9	127	255	91	51	255	179	21
255	13	119	255	99	47	255	187	19
255	21	111	255	107	45	255	191	17
255	29	95	255	115	43	255	199	15
255	37	91	255	123	39	255	207	13
255	45	87	255	131	37	255	215	11
255	47	85	255	139	31	255	223	9
255	55	63	255	147	29	255	231	7
255	63	61	255	155	27	255	239	5
255	71	59	255	163	25	255	247	3
255	79	55						

The classical Hamming bound and GV bound defined in Eq. (7.21) and (7.22) are valid for asymptotical limit where $n \to \infty$.

The classical Hamming codes constitute the so-called *perfect codes* for a finite-length, since they always achieve the Hamming bound for finite-length codes[1]. Therefore, the Hamming codes also mark the upper bound of normalized minimum distance, given the coding rate of finite-length codewords. Secondly, the classical BCH codes having a codeword length of $n = 255$ lay perfectly - as expected - between the Hamming and GV bound in the asymptotical limit, as shown in Fig. 7.7. However, we observe an unusual behaviour for the family of TECCs, since the normalized minimum distance drops to zero upon increasing the codeword length, while the coding rate saturates at $r = 1/2$. We hypothesize that since these codes were not designed using the sphere packing concept - which the Hamming and BCH codes are based on - the Hamming distance radius of the associated decoding sphere in the TECCs codespace is most likely to be non-identical for the different codewords. Also, the minimum distance of TECCs is only on the order of $\mathcal{O}(\sqrt{n})$, which implies that the codeword length of TECCs is proportional to the factor of $\mathcal{O}(d^2)$. By contrast, for classical BCH and Hamming codes the growth of the minimum distance is approximately linear, i.e. of order $\mathcal{O}(n)$. It can be seen that even though the growth of minimum distance per codeword of the TECCs appears to be impressive in Fig. 7.6, it is not fast enough to compensate for the undesired effect of the increasing codeword length. Hence, the TECC error correction capability per bit tends to zero in the asymptotical limit. Nevertheless, we leave the definitive answer for this peculiar phenomenon open for future research, since our focus in this treatise is on finding the classical-to-quantum isomorphism of TECCs.

Since the TECCs associated with the asymptotical limit of $n \to \infty$ belong to the family of LDPC codes, an efficient LDPC decoder such as the belief propagation (BP) technique [268] can be invoked for these code constructions. However, the normalized minimum distance of the LDPC codes based on topological order tends to zero, as the codeword length increases. Nevertheless, TECC-based LDPC codes exhibit several desirable code properties, such as an attractive coding rate ($r \approx 1/2$), structured construction and unbounded minimum distance. However, another aspect worth considering for TECC-based LDPC codes is the fact that we can find numerous cycles of length 4 in triangular constructions and cycles of length 6 in square constructions, which potentially degrades the performances of the codes. A summary of the specific code parameters of TECC-based LDPC codes is given in Table 7.5.

[1]The Hamming bound for finite length codes has a different formulation from that of asymptotical limit. Therefore, we refer to [204] for further explanations.

Table 7.5: The code parameters of TECC-based LDPC codes.

Parameter	Square lattice	Triangular lattice
r	$\approx \frac{1}{2}$	$\approx \frac{1}{2}$
d	$\mathcal{O}(\sqrt{n})$	$\mathcal{O}(\sqrt{n})$
δ	$\frac{d}{2d^2-2d+1}$	$\frac{4d}{3d^2+1}$
$\rho_{\max}$	4	6
$\gamma_{\max}$	2	3
Girth	6	4

7.3 Quantum Topological Error Correction Codes: Design Examples

Let us now delve deeper into the TECC concept in the quantum domain. The quantum version of TECCs, namely the QTECCs, constitute a member of the QSC family, whose stabilizer operators are defined by the underlying lattice structure. This formalism offers several benefits for the implementation of quantum computers. Firstly, it explicitly accommodates the physical implementation of quantum memory by mapping the qubits to the lattice arrangement exemplified by Fig. 7.2 and 7.3. Secondly, the localized nature of the stabilizer measurements confines the interaction amongst qubits and also eliminates the interaction of qubits associated with specific quantum gates that are physically far from each other within the topology. Thirdly, the number of errors corrected can be increased simply by extending the size of the lattice. For now, let us assume having a square-shaped lattice structure similar to Fig. 7.2 for defining the stabilizer operators of a surface code illustrated in Fig. 7.8 [103]. Explicitly, surface codes represent the quantum equivalent of classical TECCs defined on rectangular lattice structures. The physical qubits are portrayed by the black circles laying on the edge of the lattice, the **X** stabilizer operators are defined by the red squares on the lattice vertices, while the **Z** stabilizers are defined by the blue triangles on the lattice plaquettes (faces). The stabilizer operators of QTECCs are defined as follows:

$$A_v = \prod_{i \in \text{vertex}(v)} \mathbf{X}_i \ , \ B_p = \prod_{i \in \text{plaquette}(p)} \mathbf{Z}_i, \tag{7.23}$$

where i indicates the index of stabilizer operators containing the Pauli matrix **X** as well as **Z** and the rest of the stabilizer operators are given by the Pauli identity matrix **I**. Hence, the encoded state of the physical qubits of QTECCs is constrained within a code space $\mathcal{C}$ satisfying

$$\mathcal{C} = \{|\overline{\psi}\rangle \in \mathcal{H} | A_v|\overline{\psi}\rangle = |\overline{\psi}\rangle, \ B_p|\overline{\psi}\rangle = |\overline{\psi}\rangle; \ \forall v, p\}. \tag{7.24}$$

More specifically, let us revisit Fig. 7.8 for exemplifying the construction of the stabilizer operators of a QTECC, namely of a surface code, which is one of the QTECC constructions whose stabilizer operators are defined by a rectangular lattice structure [103]. For instance, the red square on the vertex number 3 of Fig. 7.8 represents the **X** stabilizer operator of $A_3 = \mathbf{X}_4\mathbf{X}_6\mathbf{X}_7\mathbf{X}_9$, as seen in the row S_3 of Table 7.6. Similarly, the blue triangle on the plaquette number 5 of Fig. 7.8 defines the **Z** stabilizer operator of $B_5 = \mathbf{Z}_7\mathbf{Z}_9\mathbf{Z}_{10}\mathbf{Z}_{12}$, as seen in the line B_5 of Table 7.6. By performing the same evaluation for all of the red squares and blue triangles, we arrive at the stabilizer operators for the quantum surface codes, as listed in Table 7.6.

by revisiting

Table 7.6: The stabilizer operators (S_i) of the quantum surface code having the lattice construction of Fig. 7.8. The code has a minimum distance of 3 ($d = 3$), which means that it is only capable of correcting a single qubit error.

S_i	A_v	S_i	B_p
S_1	$\mathbf{X}_1\mathbf{X}_2\mathbf{X}_4$	S_7	$\mathbf{Z}_1\mathbf{Z}_4\mathbf{Z}_6$
S_2	$\mathbf{X}_2\mathbf{X}_3\mathbf{X}_5$	S_8	$\mathbf{Z}_2\mathbf{Z}_4\mathbf{Z}_5\mathbf{Z}_7$
S_3	$\mathbf{X}_4\mathbf{X}_6\mathbf{X}_7\mathbf{X}_9$	S_9	$\mathbf{Z}_3\mathbf{Z}_5\mathbf{Z}_8$
S_4	$\mathbf{X}_5\mathbf{X}_7\mathbf{X}_8\mathbf{X}_{10}$	S_{10}	$\mathbf{Z}_6\mathbf{Z}_9\mathbf{Z}_{11}$
S_5	$\mathbf{X}_9\mathbf{X}_{11}\mathbf{X}_{12}$	S_{11}	$\mathbf{Z}_7\mathbf{Z}_9\mathbf{Z}_{10}\mathbf{Z}_{12}$
S_6	$\mathbf{X}_{10}\mathbf{X}_{12}\mathbf{X}_{13}$	S_{12}	$\mathbf{Z}_8\mathbf{Z}_{10}\mathbf{Z}_{13}$

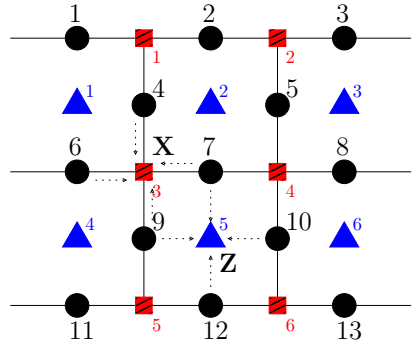


Figure 7.8: Example of qubit arrangement on a rectangular lattice structure. The black circle-based qubits on the edges of the lattice represent the physical qubits or the encoded state, the red square-based qubits lying on the vertices of the lattice act as the **X** stabilizer operators, while the blue triangle-based qubits lying on the plaquettes (faces) of the lattice constitute the **Z** stabilizer operators. ©Chandra *et al.* [174]

Let us now consider an example of how the error correction procedure works by revisiting Fig. 7.8 using QTECCs, which are similar to classical TECCs. For instance, let us assume that the quantum decoherence imposes a bit-flip (**X**) error on the physical qubit index 7. Since, the **X**-type error commutes with the **Z** stabilizer operators, which are represented by the blue triangles, the adjacent **Z** stabilizer operators return the eigenstate values of -1 upon measurement. Consequently, the **Z** stabilizer measurements yield a syndrome vector of $\mathbf{s}_z = [0\ 1\ 0\ 0\ 1\ 0]$, where only the vector elements of $i = 2, 5$ have the value of 1. For the a short block code considered in Fig. 7.8, the error recovery operators $\mathcal{R}$ to be detailed later in Fig. **??** are determined based on hard-decision maximum-likelihood (ML) decoding, which is translated into a simple look-up table (LUT) decoder. Therefore, based on the syndrome vector of $\mathbf{s}_z$, the error recovery operator $\mathcal{R}$ of Fig. **??** is given by $\mathcal{R} = \mathbf{X}_7$. Likewise, let us now assume that the qubit on index 7 also suffers from a **Z**-type error imposed by the quantum channel. The associated syndrome vector gleaned from the **X** stabilizer operators is $\mathbf{s}_x = [0\ 0\ 1\ 1\ 0\ 0]$, where only the vector elements of $i = 3, 4$ have the value of 1. Thus, based on the syndrome vector of $\mathbf{s}_x$, the decoder applies the error recovery operator of $\mathcal{R} = \mathbf{Z}_7$.

The error correction capability t of a QSC $\mathcal{C}[n, k, d]$ can be determined by its minimum distance d as follows:

$$t = \left\lfloor \frac{d-1}{2} \right\rfloor . \tag{7.25}$$

Therefore, in order to verify the error correction capability of a QSC $\mathcal{C}$, first, we have to evaluate the minimum distance d based on the stabilizer operators $S_i \in \mathcal{S}$. Let the normalizer $\mathcal{N}(\mathcal{S}) \in \mathcal{P}_n$ be the set of operators

$P_i \in \mathcal{P}_n$ so that $PS_iP^\dagger = S_j \in \mathcal{S}$ for all $S_i \in \mathcal{S}$ and i is not necessarily equal to j. It is clear that all the stabilizer operators $S_i \in \mathcal{S}$ are automatically in $\mathcal{N}(\mathcal{S})$. Now, we are interested in the set of operators in the normalizer $\mathcal{N}(\mathcal{S})$ that does not belong to the stabilizer operators $\mathcal{S}$, which is denoted by $\mathcal{N}(\mathcal{S}) - \mathcal{S}$. The minimum distance of a QSC $\mathcal{C}$ is equal to d if and only if $\mathcal{N}(\mathcal{S}) - \mathcal{S}$ contains no elements with weight less than d, where the weight of a Pauli operator $P_i \in \mathcal{P}_n$ is the number of non-identity Pauli operators. In other words, the minimum distance of a QSC $\mathcal{C}$ can be defined by the minimum weight of the operators P_i, which commute with all stabilizer operators $S_i \in \mathcal{S}$, but are not an element of $\mathcal{S}$.

In case of the rectangular lattice structure of Fig. 7.8, a good example of such an operator P_i is represented by the specific chain connecting the two boundaries of the lattice. To elaborate a little further, let us take a Pauli operator P represented by the shortened version $P = \mathbf{X}_2\mathbf{X}_7\mathbf{X}_{12}$ connecting the boundaries of the lattice given in Fig. 7.8. It can be readily checked that this specific Pauli operator P commutes with all the stabilizer generators $G_i \in \mathcal{S}$, but it cannot be represented as the product of any stabilizer generators $G_i \in \mathcal{S}$. Since P_i represents the lowest weight Pauli operator P_i commuting with all the stabilizer operators $S_i \in \mathcal{S}$, but not an element of $\mathcal{S}$, the weight of P_i determines the minimum distance of the QSC defined by the rectangular lattice of Fig. 7.8. Therefore, we conclude that based on the stabilizer generators given in Table 7.6, the minimum distance of the QSC defined by the rectangular lattice of Fig 7.8 is $d = 3$. Furthermore, the dimension of the lattices defining the stabilizer operators $G_i \in \mathcal{S}$ can be used to calculate the minimum distance d, the number of logical qubits and physical qubits, as well as the quantum coding rate r_Q.

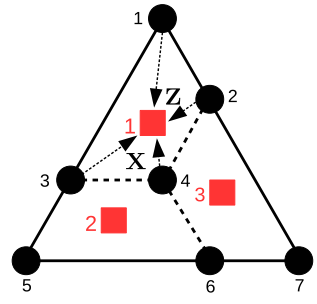


Figure 7.9: Example of a qubit arrangement for a colour code, which is a type of QTECCs whose stabilizer operators are defined by a triangular lattice structure. The black circle-based qubits on the vertices of the lattice represent the physical qubits, while the faces or the plaquettes of the lattice denoted by red squares define stabilizer operators of the colour code. The resultant code has a minimum distance of $d = 3$ and hence becomes capable of correcting a single qubit error. This specific configuration bears resemblance to Steane's 7-qubit code $\mathcal{C}[7, 1, 3]$. ©Chandra *et al.* [174]

Again, similar to the classical TECCs, the construction of QTECCs is not limited to the square lattice structure. In this spirit, let us now elaborate on another construction inspired by [105] using a triangular lattice based on the classic example of Fig. 7.3. In the proposal of [105], this specific code construction is often referred to as the (triangular) colour code, since the underlying triangular lattice may be composed by three distinct coloured tiles, each representing a non-overlapping topological region of interaction and protection. However, constructing the stabilizer operators of colour codes slightly differs from that of the surface codes. Explicitly, the colour codes use the lattice plaquettes to define both the **Z** and **X** stabilizer operators, rather than having separate stabilizers. Consequently, the resultant colour codes belong to the family of dual-containing CSS codes, which is in contrast to the rectangular-shaped surface codes constellation that belong to the class of non-dual-containing CSS codes. For colour codes, defining both the **Z** and **X** stabilizer operators using the same plaquette always guarantees to satisfy the symplectic criterion of Eq. (**??**). However, for the rectangular-shaped surface code constellations, we cannot always satisfy the symplectic criterion by using the same procedure. Therefore, the dual pair of the lattice is used for defining half of the stabilizer operators of the surface codes to satisfy

the symplectic criterion.[2]

Let us consider Fig. 7.9 for constructing the stabilizer operators of distance-3 colour codes, which are only capable of correcting a single qubit error. The plaquette denoted by red square at index 3 is used to define both the $\mathbf{Z}$ and $\mathbf{X}$ stabilizer operators. Thus, the resultant $\mathbf{X}$ stabilizer operator is $A_3 = \mathbf{X}_2\mathbf{X}_4\mathbf{X}_6\mathbf{X}_7$ and the resultant of $\mathbf{Z}$ stabilizer operator is $B_3 = \mathbf{Z}_2\mathbf{Z}_4\mathbf{Z}_6\mathbf{Z}_7$. The stabilizer operators for the colour code having the minimum distance 3 in Fig. 7.9 are listed in Table 7.7. We can observe that the colour code of Fig. 7.3 exhibits a strong resemblance to Steane's 7-qubit code.

To draw on the parallelism between classical TECCs and QTECCs, let us consider the stabilizer operators of the colour code having a minimum distance of $d = 3$, as seen in Table 7.7. Since the distance-3 colour code belongs to the family of quantum CSS codes, the PCM $\mathbf{H}$ obtained by using Eq. (3.18) and (3.21) as well as Table 7.7 may be formulated as follows:

$$\mathbf{H} = \left(\begin{array}{ccccccc|ccccccc} 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 1 \end{array}\right). \tag{7.26}$$

A CSS stabilizer code $\mathcal{C}[n, k, d]$ having $(n - k)$ stabilizer operators can be portrayed as a classical code having a PCM $\mathbf{H}$ containing $(n-k)\times 2n$ elements. Therefore, the coding rate of the classical dual of a quantum CSS code can be expressed as follows [81]:

$$\begin{aligned} r_C &= \frac{2n - (n-k)}{2n}, \\ &= \frac{n+k}{2n}, \\ &= \frac{1}{2}\left(1 + \frac{k}{n}\right), \\ &= \frac{1}{2}\left(1 + r_Q\right), \end{aligned} \tag{7.27}$$

where r_C is the coding rate of the classical dual of the stabilizer code $C[n, k, d]$ exhibiting a quantum coding rate of r_Q. The relationship between the classical and quantum coding rate in Eq. (7.27) can be rewritten as

$$r_Q = 2r_C - 1. \tag{7.28}$$

For instance, let us consider the distance-3 colour codes $\mathcal{C}[n, k, d] = \mathcal{C}[7, 1, 3]$, as exemplified in Fig. 7.9 and its classsical dual $\mathcal{C}(n, k, d) = \mathcal{C}(7, 4, 3)$, as seen in Fig. 7.3. Explicitly, we have the classical coding rate of $r_C = 4/7$ for the $\mathcal{C}(7, 4, 3)$ code. By substituting $r_C = 4/7$ into Eq. (7.28), we obtain the quantum coding rate for its quantum counterpart as $r_Q = 1/7$, which is the quantum coding rate of the distance-3 colour code $\mathcal{C}[7, 1, 3]$. The same goes for the classical square codes and their quantum counterpart, namely for the surface codes. Let us consider the distance-5 classical square code, which is labeled by S2 in Fig. 7.6 and its quantum pair, which is labeled by S2 in Fig. 7.10. We can readily determine the quantum coding rate of the surface code S2 $\mathcal{C}[41, 1, 5]$, which is $r_Q = 1/41$. Therefore, by substituting $r_Q = 1/41$ into Eq. (7.27), we arrive at the coding rate of its classical dual given by $r_C = 21/41$, which is indeed the coding rate of the classical square code S2 $\mathcal{C}(41, 21, 5)$.

[2]The dual pair of a lattice or a graph G is the graph that has a vertex for each plaquette of the graph.

Table 7.7: The stabilizer operators (S_i) of the colour code seen in Fig. 7.9. The code has a minimum distance of 3 ($d = 3$), which means that it is only capable of correcting a single qubit error.

S_i	A_p	S_i	B_p
S_1	$\mathbf{X}_1\mathbf{X}_2\mathbf{X}_3\mathbf{X}_4$	S_4	$\mathbf{Z}_1\mathbf{Z}_2\mathbf{Z}_3\mathbf{Z}_4$
S_2	$\mathbf{X}_3\mathbf{X}_4\mathbf{X}_5\mathbf{X}_6$	S_5	$\mathbf{Z}_3\mathbf{Z}_4\mathbf{Z}_5\mathbf{Z}_6$
S_3	$\mathbf{X}_2\mathbf{X}_4\mathbf{X}_6\mathbf{X}_7$	S_6	$\mathbf{Z}_2\mathbf{Z}_4\mathbf{Z}_6\mathbf{Z}_7$

Table 7.8: The code parameters for various QTECCs based on the minimum distance d of the code.

Codes type	**Dimension**	**Number of physical qubits**	**Number of stabilizers**	**Number of logical qubits**
Colour	d^*	$\frac{1}{4}\left(3d^2+1\right)$	$\frac{1}{4}\left(3d^2-3\right)$	1
Rotated-surface	$d \times d$	d^2	d^2-1	1
Surface	$d \times d$	$2d^2-2d+1$	$2d^2-2d$	1
Toric	$d \times d$	$2d^2$	$2d^2-2$	2

* for triangular colour codes the dimension is defined by the side length of the equilateral triangle

Similar to their classical counterparts, the code parameters of QTECCs, such as the number of logical qubits k, the number of physical qubits n, the minimum distance of the code d, as well as the quantum coding rate r_Q, depend on the size of the lattices. Following the same line of investigation as for the classical TECCs, in Table 7.8 we characterize the family of QTECCs.

Explicitly, we plot the minimum distance (d) versus quantum coding rate (r_Q) of QTECCs in Fig. 7.10 for colour codes [105] for the family of rotated surface codes [104], for surface codes [103] and for toric codes [100]. We also include the family of non-topological QBCH codes [65] having $n = 127$ physical qubits and the quantum Hamming codes, which constitute the quantum-domain pair of the Hamming code constructions [61]. Similarly to the classical domain, the behaviour of both the QBCH codes and the quantum Hamming codes is as expected, namely reminiscent of the behaviour inherited from their classical-domain counterparts. However,

Table 7.9: Code parameters of quantum Hamming codes having a single error correction capability, which is used in Fig. 7.10 and 7.11. The quantum coding rate r_Q and normalized minimum distance δ are calculated using Eq. (7.1) and (7.17), respectively.

n	k	d	n	k	d
8	3	3	256	246	3
16	10	3	512	501	3
32	25	3	1024	1012	3
64	56	3	2048	2035	3
128	119	3	...	...	...

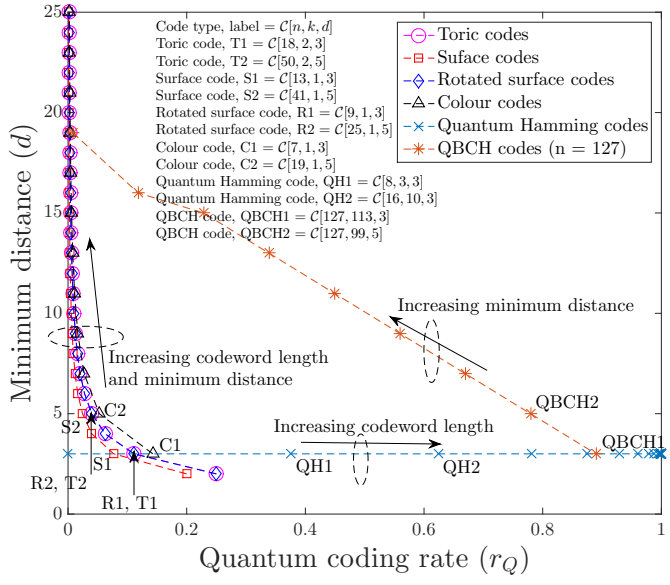


Figure 7.10: The minimum distance (d) versus the quantum coding rate (r_Q) of QTECCs based on the parameter given in Table 7.8. For QTECCs, the quantum coding rate tends to zero as we increase the minimum distance. We also include the QBCH codes having the physical qubits of $n = 127$ and quantum Hamming codes for the sake of comparing the QTECCs with the non-topological QSCs. ©Chandra *et al.* [174]

Table 7.10: Code parameters of QBCH codes having codeword length of $n = 127$, which is used in Fig. 7.10 and 7.11. The quantum coding rate r_Q and normalized minimum distance δ are calculated using Eq. (7.1) and (7.17), respectively.

n	k	d	n	k	d
127	1	19	127	71	9
127	15	16	127	85	7
127	29	15	127	99	5
127	43	13	127	113	3
127	57	11			

it is interesting to observe that the quantum coding rate of QTECCs tends to zero for long codewords, which is not unexpected, if we consider the classical to quantum isomorphism in the context of the coding rate given in Eq. (7.27) and (7.28). For the classical TECCs, the coding rate r_C approaches the value of $r_C = 1/2$ for long codewords. Hence, by substituting $r_C = 1/2$ into Eq. (7.28), we arrive at $r_Q = 0$, which is the phenomenon we observe in Fig. 7.10.

Next, we plot the normalized minimum distance (δ) versus the quantum coding rate (r_Q) in Fig. 7.11. Once again, for the sake of comparison, we also include the quantum Hamming bound [62] and the quantum GV bound derived for CSS codes [90] in addition to the QBCH codes and the quantum Hamming codes. The

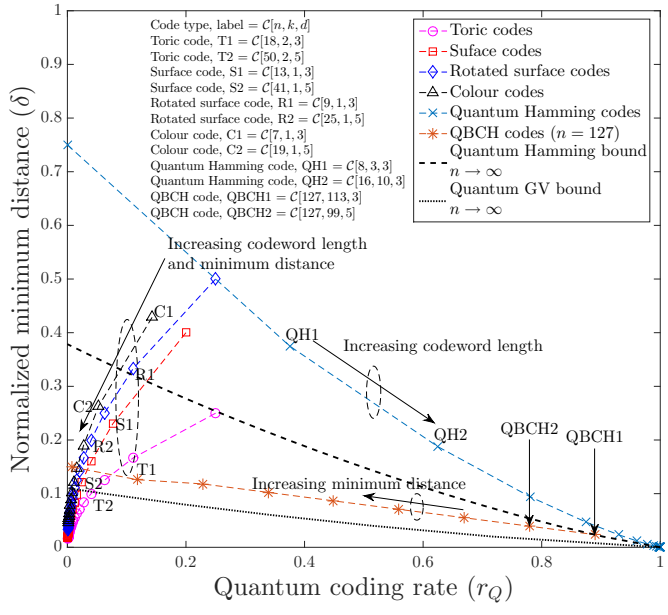


Figure 7.11: The normalized minimum distance (δ) versus the quantum coding rate (r_Q) of QTECCs based on the parameter given in Table 7.8. For QTECCs, the normalized minimum distance and quantum coding rate tend to zero as we increase the minimum distance. We also include the QBCH codes having the physical qubits of $n = 127$, quantum Hamming codes, quantum Hamming bound and also quantum GV bound for CSS codes for the sake of comparing the QTECCs with the non-topological QSCs. ©Chandra *et al.* [174]

quantum Hamming bound is defined by [62]

$$\frac{k}{n} \leq 1 - \left(\frac{d}{2n}\right) \log_2 3 - H\left(\frac{d}{2n}\right), \tag{7.29}$$

while the quantum GV bound for CSS codes is given by [90]

$$\frac{k}{n} \geq 1 - 2H\left(\frac{d}{n}\right). \tag{7.30}$$

Both the quantum Hamming bound and the quantum GV bound of Fig. 7.11 serve the same purpose as the classical Hamming bound and the GV bound seen in Fig. 7.7. Explicitly, they portray the upper bound and the lower bound of normalized minimum distance versus quantum coding rate trade-off. Once again, the puzzling behaviour of classical TECCs resurfaces for the QTECCs, as observed in Fig. 7.11. Since all the QBCH codes, quantum Hamming codes and QTECCs inherit the properties of their classical counterparts, their behaviour is reminiscent of that of their classical counterparts. As for the QTECCs, the definitive interpretation of this unusual behaviour is left for future exploration in our research. Nonetheless, for a relatively long codeword, the QTECCs are reminiscent of QLDPC codes. Observe from Fig. 7.11 that both the normalized minimum distance and the quantum coding rate of QTECCs tend to zero upon increasing the minimum distance by increasing the codeword length. Therefore, the QTECCs are deemed to be more favourable for short to medium codeword lengths.

7.4 Performance of Quantum Topological Error Correction Codes

In this section we consider the performance of QTECCs in the face of quantum depolarizing channels. Explicitly, the quantum depolarizing channel is characterized by the quantum depolarizing probability p inflicting an error pattern constituted by the Pauli operators $P \in \mathcal{P}_n$ upon the state of physical qubits. Each qubit may independently experience a bit-flip error ($\mathbf{X}$), a phase-flip error ($\mathbf{Z}$), or both bit-flip and phase-flip error ($\mathbf{Y}$) with an equal probability of $p/3$. In order to get a more precise insight into the performance trends of QTECCs, we have to distinguish how the different error patterns affect the state representing the physical qubits. Explicitly, the n-tuple Pauli error patterns exemplified in Fig. 7.12 and 7.13 may be classified as follows:

(a) **Harmful detected error pattern**. This specific type of error pattern has a similarity to the conventional bit errors experienced in the classical domain. The error pattern of $\mathcal{P}$ anti-commutes with the stabilizer operators $S_i \in \mathcal{S}$, hence it triggers non-trivial syndrome values.

(b) **Harmful undetected error pattern**. The error pattern commutes with all of the stabilizer operators, but it does not belong to the stabilizer group $\mathcal{S}$. In the classical domain, this is similar to an error pattern that returns an all-zero syndrome. The error pattern is harmful, since it does not result in a non-trivial syndrome value, yet it corrupts the legitimate state of the physical qubits.

(c) **Harmless undetected error pattern**. This particular error pattern does not have any classical counterpart. The error pattern is harmless because it belongs to the stabilizer group $\mathcal{S}$. This is also referred to as a degenerate error pattern. Consequently, the error pattern does not alter the legitimate state of the physical qubits. By considering the phenomenong referred to as degeneracy, the actual performances of QTECCs are potentially improved.

To illustrate both the harmless and harmful undetected error patterns, we refer to Fig. 7.12 and 7.13. We commence by considering the *harmless undetected error patterns*, which are illustrated in Fig. 7.12. In this example, we consider a surface code having a minimum distance of 5, which implies that it is only capable of correcting two-qubit errors. Following the stabilizer formulation of QTECCs discussed in Section 7.3, the physical qubits are arranged along the edges of the square lattice, while the $\mathbf{X}$ stabilizer operators are located in the vertices. Therefore, the $\mathbf{X}$ stabilizer operators on the vertices are used for indicating the $\mathbf{Z}$ errors, which will trigger eigenvalues of -1 if they anticommute with the $\mathbf{X}$ stabilizer operators. Let us assume that the quantum depolarizing channel inflicts three $\mathbf{Z}$ errors on the physical qubits, which are denoted by the filled black circles in Fig 7.12, while the hollow black circles represent the error-free physical qubits. All of the error patterns given in Fig 7.12 (a), (b) and (c) trigger the eigenvalues of -1 for the stabilizer operators denoted by filled red squares, while the rest of the stabilizer operators are represented by hollow red squares, which return eigenvalues of $+1$. Since the decoder relies on hard-decision ML decoding, all of the error patterns given in Fig. 7.12 (a), (b) and (c) have the same probability of occurrence. Let us assume that the decoder always decides to apply the error recovery pattern of Fig. 7.12 (a) for the specified values of stabilizer measurement. When the actual error pattern is the one given in Fig. 7.12 (a), the states of the physical qubits are fully recovered. By contrast, if the actual error pattern is the one seen in Fig. 7.12 (b), but it is corrected using the error recovery operator of Fig. 7.12 (a), we arrive at the accumulated error pattern shown in Fig. 7.12 (d). Lastly, when the actual error pattern is the one given by Fig. 7.12 (c), but we attempt to correct it using the error recovery of Fig. 7.12 (a), we obtain the error pattern seen Fig. 7.12 (e). However, if we observe closely the error pattern illustrated in Fig. 7.12 (d), it is reminiscent of a plaquette $\mathbf{Z}$ stabilizer operator denoted by the filled blue triangle. Therefore, based on the definition of stabilizer operators, the error pattern given in Fig. 7.12 (d) does not alter the legitimate state of physical qubits. Similarly, the error pattern of Fig. 7.12 (e) resembles the product of two adjacent plaquette stabilizer operators. Since the product between a pair of stabilizer operators return another valid stabilizer operator, the error pattern given in Fig. 7.12 (e) belongs to the stabilizer group $\mathcal{S}$. Once again, by definition, the error pattern given in Fig. 7.12 (e) does not corrupt the legitimate state of physical qubits. This is an example of *harmless undetectable error patterns*.

To elaborate a little further, a harmless undetected error can be directly generated by the quantum de-

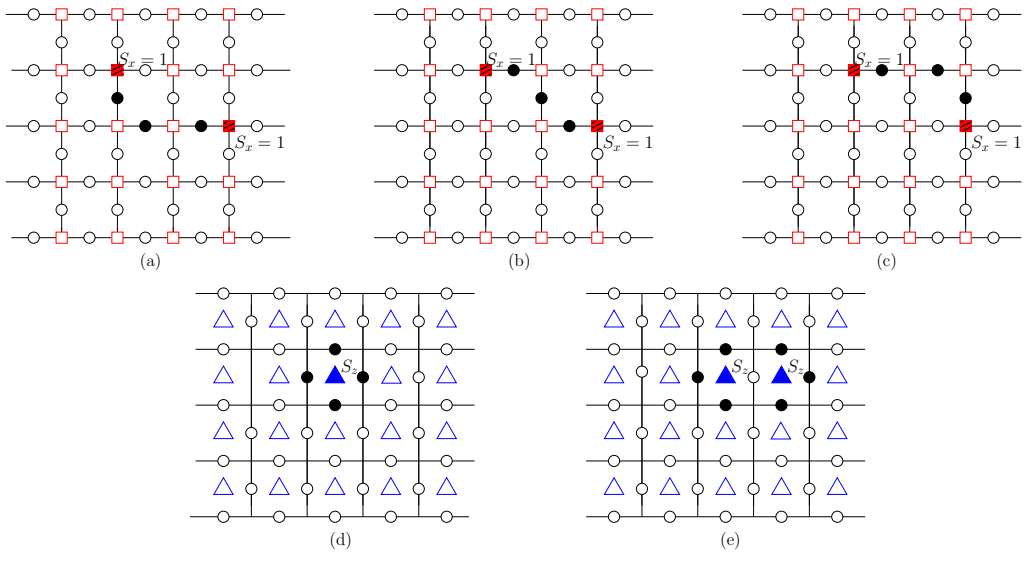


Figure 7.12: Illustration of how the error recovery operator $\mathcal{R}$ creates the degenerate error patterns and how the degeneracy nature of QECCs may improve the performance of QTECCs. All of the error patterns given in (a), (b) and (c) represent error patterns generating an identical syndrome value. Without loss of generality, let us assume that based on the generated syndrome value, the decoder always decides to perform error recovery operator $\mathcal{R}$ of (a) on the corrupted state of physical qubits. If the actual error pattern is (a), the corrupted state of physical qubits will be fully recovered. By contrast, figure (d) shows the resultant error pattern if the actual error pattern is (b), but it is corrected using the error pattern given in (a). Moreover, figure (e) represents the resultant error pattern if the actual error pattern is (c) and it is corrected using the error recovery pattern of (a). As a result, the error pattern (d) represents a stabilizer operator of a plaquette, while the error pattern (e) resembles the product of two adjacent stabilizer operators. Both error patterns of (d) and (e) constitute the *harmless undetectable error patterns*, since they belong to the stabilizer group $\mathcal{S}$. Therefore, the state of physical qubits is not altered after the recovery operator $\mathcal{R}$ of (a) is applied to all error patterns of (a), (b) and (c). In classical setup, both error patterns (d) and (e) are considered as error events. However, in the quantum domain, both error patterns (d) and (e) are considered as error-free cases. This specific error-type has no similarity in the classical domain and hence potentially improves the performance of QTECCs. ©Chandra *et al.* [174]

coherence, where the Pauli operator $P \in \mathcal{P}_n$ imposed by the quantum depolarizing channel is identical to the stabilizer operator S_i. Another possibility is that it is generated by the associated error recovery procedure when trying to recover an ambiguous error pattern, where there are more than one possible error patterns associated with a specific syndrome value, as illustrated in Fig. 7.12. The degeneracy property, which is associated with the harmless undetectable error patterns, does not have a classical analogue, because the resultant error patterns illustrated in Fig. 7.12 (d) and (e) will always be considered as an error in the classical setup. Ultimately, considering the degeneracy potentially improves the performance of QECCs.

Let us consider a range of different scenarios for illustrating the presence of harmful undetected error patterns, which is portrayed in Fig. 7.13. Similar to the previous example of Fig. 7.12, three **Z** errors are imposed on the state of logical qubits by the quantum depolarizing channel. The error patterns given in

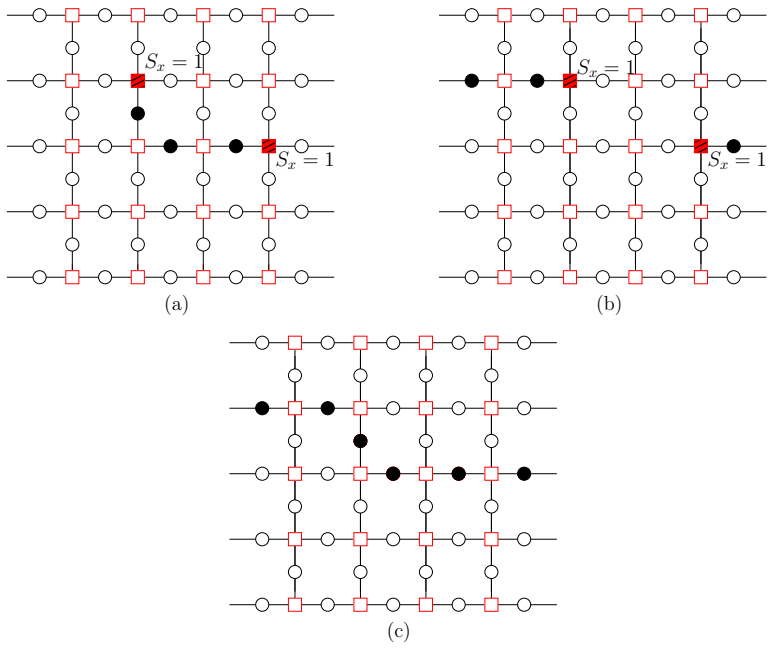


Figure 7.13: Illustration of the harmful undetectable error pattern in the quantum domain. The actual error pattern inflicts the state of physical qubits is given in (b), while the decoder always decides to perform a recovery operator given in (a). Instead of recovering the legitimate state of the physical qubits, the specified error recovery procedure generates a chain of error that commutes with all of the stabilizer operators, as shown in (c). In the quantum domain, it constitutes the *harmful undetectable error patterns*. In the classical domain, it resembles the error pattern that generates all-zero syndrome values. ©Chandra *et al.* [174]

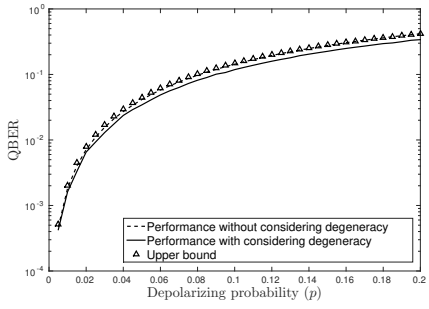
Fig. 7.13 (a) and (b) trigger the eigenvalues of -1 for the stabilizer operators denoted by filled red squares in Fig. 7.13, while the rest of the stabilizer operators represented by hollow red squares return eigenvalues of $+1$. Given the associated syndrome value, the decoder always decides to apply the error recovery operator of Fig. 7.13 (a). In the specific scenario, where the actual error pattern is the one given by Fig. 7.13 (b), the resultant error pattern is given in Fig. 7.13 (c). We can observe that the resultant error pattern of Fig. 7.13 (c) commutes with all of the stabilizer operators in Fig. 7.13. However, this specific error pattern does not belong to the stabilizer operator $\mathcal{S}$, since we cannot represent a chain of errors by the product of stabilizer operators. Consequently, this undetectable error pattern inevitably corrupts the legitimate state representing the physical qubits. This is an example of the *harmful undetectable error patterns*. This error pattern is similar to that of its counterpart in the classical domain, where the error pattern returns the all-zero syndrome.

Therefore, based on these conditions, by appropriately modifying the probabilty of correct decoding in the classical domain [269], we can readily formulate the worst-case upper bound QBER performance of QTECCs as

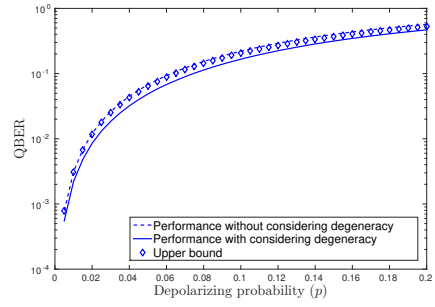
$$\mathrm{QBER}_{\mathrm{upper}}(n,d,p) = 1 - \sum_{i=0}^{t=\lfloor \frac{d-1}{2} \rfloor} \binom{n}{i} p^i (1-p)^{n-i} - \sum_{i=1,\forall S_i \in \mathcal{S}}^{|\mathcal{S}|} p^{w(S_i)} (1-p)^{n-w(S_i)}, \tag{7.31}$$

where $w(S_i)$ is the weight of the stabilizer operator S_i, which is defined by the number of non-identity Pauli operators within the stabilizer operators. The second term of Eq. (7.31) represents all the correctable error

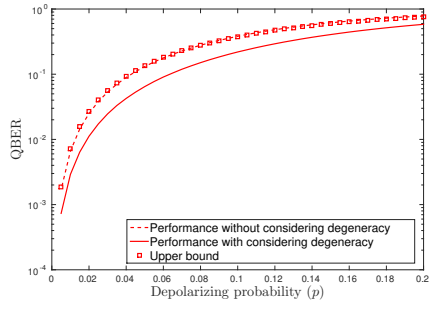
patterns of QTECCs, while the last term of Eq. (7.31) represents the degenerate error patterns that belong to the stabilizer operators. For example, let us revisit the construction of the surface codes of Fig. 7.8. There are 12 stabilizer generators for a distance-3 surface code, as seen in Table 7.6. Hence, we can potentially generate 2^{12} unique stabilizer operators, since the product of the stabilizer operators returns another valid stabilizer operator. However, in order to further simplify the expression given in Eq. (7.31), we only consider the specific error patterns resembling the specified stabilizer operators given in Table 7.6, since they exhibit a lower weight of non-identity Pauli matrices and hence have a higher probability of occurance. Therefore, for surface codes, the last term of Eq. (7.31) can be approximated as $(2d^2 - 2d)p^4(1-p)^{n-4}$. The term $(2d^2 - 2d)$ represents the number of stabilizer operators, which is given in Table 7.8, and we assume that all the weights of the stabilizer operators $w(S_i)$ are equal to 4.



(a) Colour codes.



(b) Rotated-surface codes.



(c) Surface codes.

Figure 7.14: QBER performance of the distance-3 QTECCs over the quantum depolarizing channel, which is capable of correcting a single qubit error. The code parameters are given in Table 7.11. For this scenario, the decoder using hard-input ML decoding approach for predicting the error pattern. ©Chandra *et al.* [174]

7.4.1 QBER Versus Depolarizing Probability

In order to characterize the performance of QTECCs by simulations, we exploit the fact that the QTECCs belong to the family of CSS codes, which handle the bit-flips (**X**) and phase-flips (**Z**) separately. Hence, we invoke two independent binary symmetric channels (BSC), one for the **X** channel and one for the **Z** channel, where each channel is characterized by the flip probability of $2p/3$, where p is the associated depolarizing probability of the quantum depolarizing channel [73, 77]. The decoder utilizes hard-decision ML decoding relying on a simple LUT decoder, as exemplified in Section 3.3. However, this classical-domain simulation only

Table 7.11: Code parameters for distance-3 colour code, rotated-surface code and surface code.

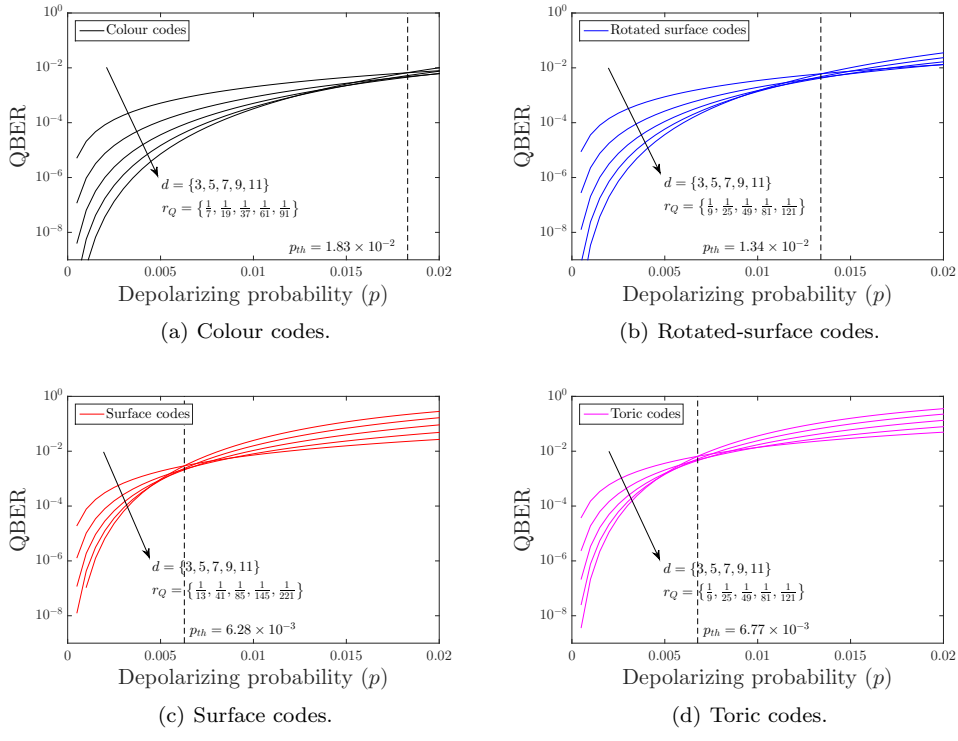
Code type	n	k	d	r_Q
Colour code	7	1	3	1/7
Rotated-surface code	9	1	3	1/9
Surface code	13	1	3	1/13

represents the performance of QTECCs without considering the degenerate error patterns. To elaborate a little further, we transmitted all-zero information bits. Therefore, we always consider the non-all-zero bits at the decoder output as an error. In order to additionally consider several cases of degenerate error patterns, which is exemplified in Fig. 7.12, we performed another evaluation step. We evaluate the non-all-zero corrected received words and checked for the degenerate error patterns. If it satisfies to the degenerate error pattern criterion that we have above, we conclude that this is an error-free case. However, we are not capable of providing a complete list of all possible degenerate error patterns. Here we only consider the error patterns resembling the stabilizer generators of S_i, as exemplified in Table 7.6 and 7.7 for surface codes and triangular codes, respectively. The QBER performance of distance-3 QTECCs versus the quantum depolarizing probability is portrayed in Fig. 7.14, where the code parameters are given in Table 7.11. We also include the upper bound of the QTECC performance of Eq. (7.31) in Fig. 7.14. It can be observed that the upper bounds match with the QTECCs performance recorded without considering the degenerate error patterns.

As we mentioned earlier, there are two sources of degenerate error patterns at the output of the decoder. Firstly, the degenerate error pattern that was imposed directly by the quantum channel, where the error exhibits an identical pattern to the stabilizer operator S_i. Secondly, the degenerate error pattern generated by the recovery operator $\mathcal{R}$, when it tries to recover the legitimate physical qubits, as illustrated in Fig. 7.12. The second case is more dominant than the first one. The reason can be explained as follows. Let us consider the $\mathbf{Z}$ stabilizer operators of the distance-3 surface code given in Table 7.6. There are six $\mathbf{Z}$ stabilizer operators corresponding to the $2^6 = 64$ possible syndrome vector, including the error-free scenario. Remember that the distance-3 surface code can only correct a single qubit error within a block of 13 physical qubits, where each of the single-qubit error patterns is associated with a single syndrome vector. In other words, amongst all of the 64 possible syndrome vectors, there are only 13 syndrome vectors used for uniquely distinguishing the correctable error patterns, while the rest of the syndrome vectors are associated with an error pattern ambiguity, as exemplified in Fig. 7.12 and 7.13. For this reason, QTECCs are highly degenerate QSCs. Hence, the upper bound of the QBER performance matches the simulation-based performance recorded without considering the degeneracy, since it considers the first case of degeneracy, where we only consider a portion of all valid stabilizer operators $S_i \in \mathcal{S}$ in Eq. (7.31). As expected, by accommodating both of the degeneracy cases, the QBER performance of QTECCs is improved, as portrayed in Fig. 7.14.

7.4.2 QBER Versus Distance from Hashing Bound

Increasing the minimum distance of a given QSC construction, which directly improves its per-codeword error correction capability (t), is achieved by increasing the number of physical qubits (n) or by decreasing the quantum coding rate. Specifically for QTECCs, increasing the minimum distance means increasing the number of physical qubits (n) and decreasing the quantum coding rate (r_Q) simultaneously. Naturally, the goal of increasing the minimum distance of the QSC is to achieve a better QBER performance. However, a QBER improvement can only be observed below a certain depolarizing probability (p), which may be referred to as the threshold probability (p_{th}). Using the QBER upper bound performance of Eq. (7.31), we plot the QBER curves for colour, rotated-surface, surface and toric codes in Fig. 7.15. For each of the QTECC constructions, we portray the upper bound QBER performance for the minimum distance values of $d = \{3, 5, 7, 9, 11\}$. The threshold probability of each code is represented by the crossover points, which are marked by dashed lines. Qunatitatively, the threshold probability of colour codes, rotated-surface, surface and toric codes are



(a) Colour codes.

(b) Rotated-surface codes.

(c) Surface codes.

(d) Toric codes.

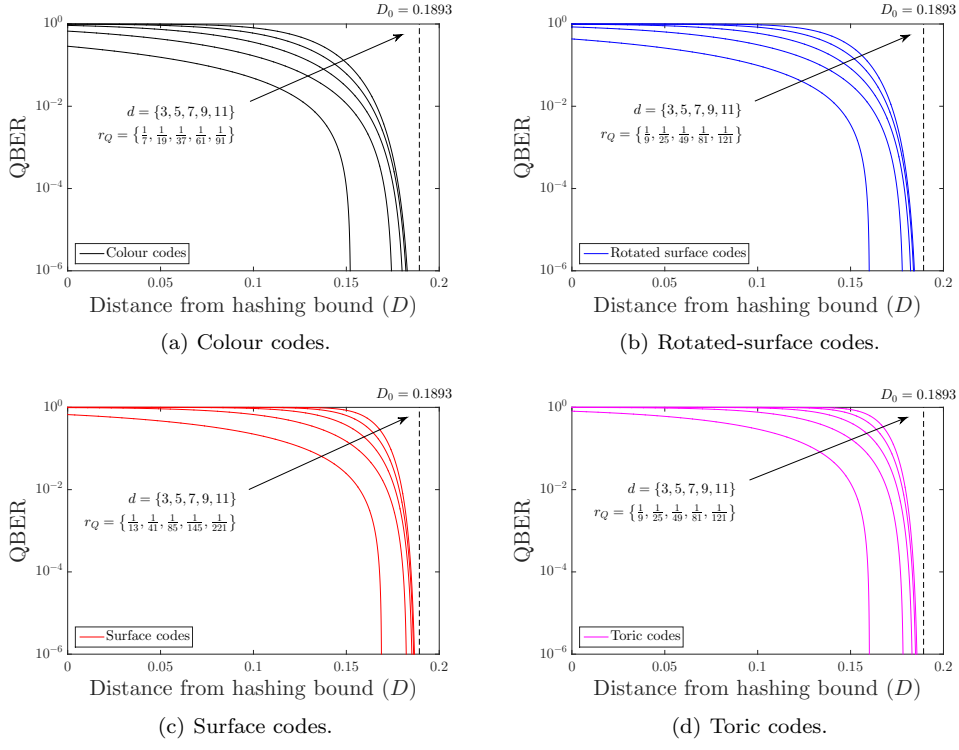
Figure 7.15: Upper bound QBER performance of QTECCs for the minimum distance of $d = \{3, 5, 7, 9, 11\}$ based on Eq. (7.31) and code parameters given in Table 7.8. The crossover amongst the QBER curves represents the threshold probability (p_{th}), which are portrayed in dashed line. ©Chandra *et al.* [174]

1.83×10^{-2}, 1.34×10^{-2}, 6.28×10^{-3} and 6.77×10^{-3}, respectively.

However, presenting the performance of QTECCs over quantum depolarizing channels by portraying the QBER curves versus the depolarizing probability (p) does not take the quantum coding rate (r_Q) into consideration. As we mentioned earlier, we can simply decrease the quantum coding rate further and further to increase the error correction capability of the QTECCs. Nonetheless, for the sake of depicting a fair comparison upon reducing the quantum coding rate, we have to scrutinize how much performance improvement we obtain upon decreasing the quantum coding rate. Therefore, to demonstrate how much performance improvement we attain compared to how much we decrease the quantum coding rate, we normalize the QBER performance by incorporating the quantum hashing bound. More explicitly, the quantum hashing bound can be expressed as follows [270]:

$$C_Q(p) = 1 - H(p) - p \cdot \log_2(3), \tag{7.32}$$

where $H(p)$ is the binary entropy of p. More specifically, the quantum hashing bound of Eq. (7.32) dictates that a random quantum code $\mathcal{C}$ having a sufficiently long codeword and a quantum coding rate $r_Q \leq C_Q(p)$ may yield an infinitesimally low QBER for a given depolarizing probability p. Alternatively, we can refer to $C_Q(p)$ as the hashing limit for the quantum coding rate r_Q associated with a given depolarizing probability p. In terms of its classical dual pair, the value of C_Q is similar to the capacity limit. Similarly, for a given coding rate r_Q, we can find a value of p^* satisfying $r_Q = C_Q(p^*)$, where p^* denotes the maximum value of depolarizing probability p so that a quantum code $\mathcal{C}$ having quantum coding rate of r_Q can operate at an infinitesimally low QBER. The value of p^* may be referred to as the hashing limit for the depolarizing probability of p associated



(a) Colour codes. (b) Rotated-surface codes.

(c) Surface codes. (d) Toric codes.

Figure 7.16: Upper bound performance of QTECCs in term of the QBER versus the distance D from the hashing bound. The dashed lines portray the ultimate distance to the quantum hashing bound of $D_0 = 0.1893$. ©Chandra *et al.* [174]

with a given quantum coding rate of r_Q. In the classical domain, the value of p^* is similar to the noise limit. Therefore, in general, the aim is that of finding a QSC that is capable of performing as close as possible to the quantum hashing bound. For example, let us consider the distance-3 and distance-5 rotated-surface codes having quantum coding rates of $r_Q = 1/9$ and $r_Q = 1/25$, respectively. By substituting $C_Q = 1/9$ and $C_Q = 1/25$ into the Eq. (7.32), we obtain the noise limits of $p^* = 0.160$ and $p^* = 0.179$, respectively. It can be concluded that the noise limit is higher for the quantum code exhibiting a lower quantum coding rate. To incorporate the quantum hashing bound into the QBER performances of QTECCs, we define the distance from the hashing bound as follows:

$$D \triangleq p(r_Q) - p. \tag{7.33}$$

In other words, by changing the horizontal axis from the depolarizing probability p to the distance D from the hashing bound, we shift all the QBER curves according to their hashing bounds, so that all the hashing bounds are at the reference point of $D = 0$.

Several pertinent questions arise from the quantum hashing bound formulation. Firstly, is there a noise limit, where no QSC constructions are capable of achieving a satisfactorily low QBER? Indeed, the answer is yes. By substituting $C_Q = 0$ into Eq. (7.32), which is the lowest possible value of achievable quantum coding rate, we arrive at the ultimate hashing bound of $p(0) \approx 0.1893$. Secondly, what is the farthest possible distance from the quantum hashing bound for any QSC construction? To answer this question, we have to consider the worst-case scenario, where a QSC exhibiting a near-zero quantum coding rate ($r_Q \approx 0$) achieves an infinitesimally low QBER at near-zero quantum depolarizing probability ($p \approx 0$). By substituting the value

of $r_Q = 0$ and $p = 0$ into Eq. (7.33), we define the ultimate distance D_0 from the hashing bound as

$$\begin{aligned} D_0 &= p(0) - p \\ &= 0.1893 - 0 \\ &= 0.1893. \end{aligned} \tag{7.34}$$

Therefore, the desirable performance of any QSCs quantified in terms of the QBER versus distance from the quantum hashing bound is represented by the curves exhibiting a reasonably low QBER as close as possible to the reference point of $D = 0$. Naturally, this implies having a low QBER as far as possible from the ultimate distance of $D_0 = 0.1893$ wrt the hashing bound. In simpler terms, any QSCs can only operate at a reasonably low QBER withing the hashing bound range of $0 \leq D \leq D_0$. Furthermore, we should only consider the quantum coding rate reduction of r_Q to be beneficial, if the associated QBER performance curve moves closer to the reference point of $D = 0$. Otherwise, it is more advisable to find a better code construction exhibiting an identical quantum coding rate. Alternatively, we may invoke a more powerful decoding scheme, for example by utilizing a soft-decision-aided decoder. The QBER performance of QTECCs versus their distances from the quantum hashing bound are portrayed in Fig. 7.16. It can be observed in Fig. 7.16 that increasing the minimum distance of the QTECCs yields a performance improvement in terms of their QBER versus depolarizing probability p snd also in terms of their distance from the hashing bound D. However, at low QBERs the curves become crowded in the vicinity of the ultimate hashing bound distance of D_0. Moreover, the results show an agreement with the quantum coding rate versus minimum distance evolution of QTECCs seen in Fig. 7.11. The improvement of the minimum distance – which is directly linked to the error correction capability – is not fast enough upon reducing the quantum coding rate to compensate for the rapidly increasing number of physical qubits. Therefore the QTECCs may be recommended for short to moderate codeword lengths.

7.4.3 Fidelity

From an implementational perspective, a quantum gate or quantum channel is often characterized by the so-called fidelity, which represents the closeness of a pure quantum state of $|\overline{\psi}\rangle$ compared to the mixed states having the quantum density operator of ρ. More explicitly, since the quantum channel imposes decoherence on our legitimate quantum states representing the physical qubits $|\overline{\psi}\rangle$, the decoder may not successfully recover the legitimate state. Therefore, the ensemble of all the possible predicted legitimate states of physical qubits $|\widehat{\psi}\rangle$ can be represented using the state of $|\psi_i\rangle$ having a probability of p_i. The fidelity can be formulated as follows [89, 271, 272]:

$$F = \langle \overline{\psi} | \rho | \overline{\psi} \rangle, \tag{7.35}$$

while ρ, which portrays the statistical characteristics of a the mixed states, is defined by

$$\rho = \sum_{i=1}^{N} p_i |\psi_i\rangle\langle\psi_i|, \tag{7.36}$$

where $|\psi_i\rangle$ represents all the possible states in the ensemble and p_i is the probability of having state $|\psi_i\rangle$ in the ensemble, which is subject to the unity constraint of $\sum_{i=1}^{N} p_i = 1$.

In order to demonstrate the benefit of QTECCs in the context of quantum depolarizing channels, we compare the so-called initial fidelity F_{in} and final fidelity F_{out}. The initial fidelity is the fidelity of the pure quantum state of $|\psi\rangle$ over the quantum depolarizing channel $\mathcal{P}$ unprotected by any QSCs scheme. Therefore, the initial fidelity F_{in} can be expressed as follows:

$$F_{\text{in}} = 1 - p. \tag{7.37}$$

The final fidelity is that of the pure state of the desired output $|\psi'\rangle$ protected by the a QSC scheme after the recovery procedure $\mathcal{R}$ and inverse encoder $\mathcal{V}^\dagger$ of Fig. **??**. Therefore, the final fidelity F_{out} of the quantum

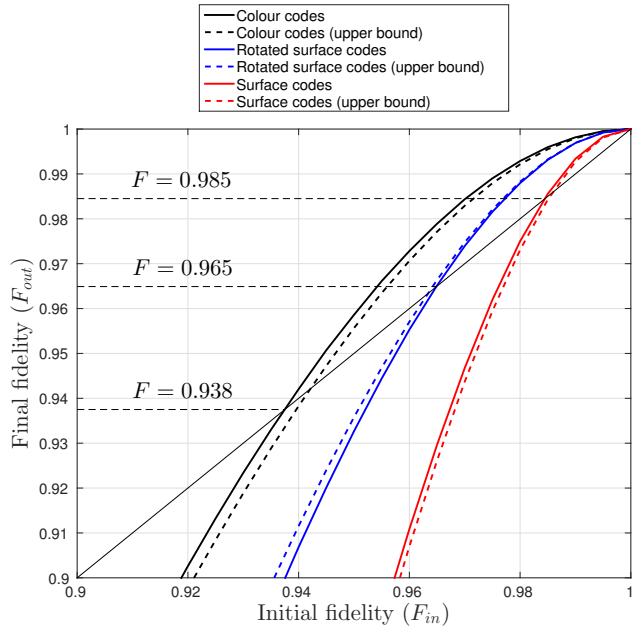


Figure 7.17: The fidelity performance of QTECCs having a minimum distance of 3 in term of fidelity of Eq. (7.35). The colour code reaches the fidelity threshold earlier than the rotated-surface and surface code, since the colour code has the lowest number of physical qubits compared to the rotated-surface code and the surface code. The code parameters are given in Table 7.11. ©Chandra *et al.* [174]

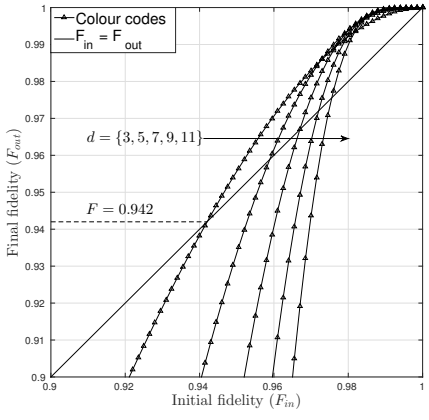
system can be readily described as

$$F_{\text{out}} = 1 - \text{QBER}. \tag{7.38}$$

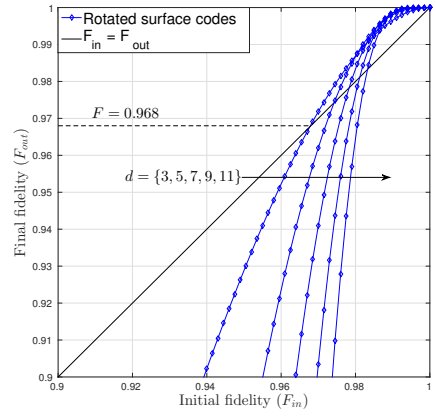
The fidelity performances of the distance-3 QTECCs are depicted in Fig. 7.17. The black solid line represents the condition of $F_{\text{in}} = F_{\text{out}}$. The crossover point between the line of $F_{\text{in}} = F_{\text{out}}$ and fidelity performance curve of QTECCs is the break-even point, which we may refer to as the *threshold fidelity* F_{th}. The break-even point denotes the minimal initial fidelity required for ensuring that we do acquire a fidelity improvement upon the applicaton of the QSC scheme, which is invoked for protecting the state of the physical qubits. The upper bound of threshold fidelity F_{th} for the different types of QTECCs is depicted in Fig. 7.18. It can be observed that different code families having various minimum distances d result in different threshold fidelity F_{th}. For the QSCs utilizing hard-decision syndrome decoding, we derive the upper bound approximation formula for F_{th}. First, from Eq. (7.31) and Eq. (7.38), we arrive at

$$\begin{aligned} F_{\text{out}} &= 1 - \text{QBER}_{\text{upper}} \\ &= 1 - \left(1 - \sum_{i=0}^{t=\lfloor \frac{d-1}{2} \rfloor} \binom{n}{i} p^i (1-p)^{n-i}\right) \\ &= 1 - \sum_{\lfloor \frac{d-1}{2} \rfloor + 1}^{n} \binom{n}{i} p^i (1-p)^{n-i}. \end{aligned} \tag{7.39}$$

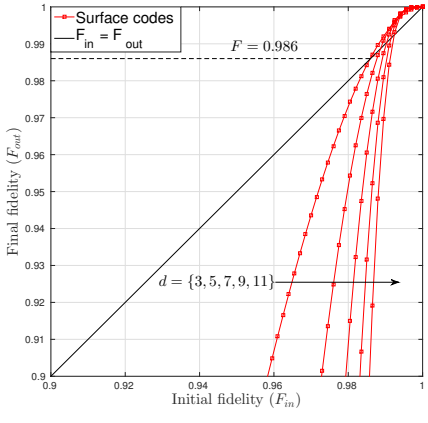
For a low depolarizing probability p, the expression given in Eq. (7.39) can be approximated in order to



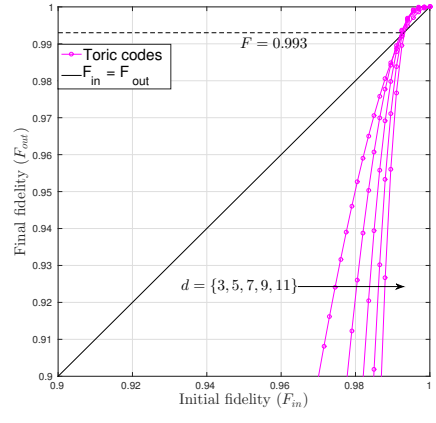
(a) Colour codes.



(b) Rotated-surface codes.



(c) Surface codes.



(d) Toric codes.

Figure 7.18: Upper bound fidelity performance of QTECCs. ©Chandra *et al.* [174]

determine the upper bound of the output fidelity as follows:

$$F_{\text{out}} \approx 1 - \binom{n}{\lfloor \frac{d-1}{2} \rfloor + 1} p^{\lfloor \frac{d-1}{2} \rfloor + 1}. \tag{7.40}$$

Since the threshold fidelity satisfies the relationship of $F_{\text{th}} = F_{\text{in}} = F_{\text{out}}$, we can substitute $F_{\text{out}} = F_{\text{th}}$ and $p = 1 - F_{\text{th}}$ into Eq. (7.40). Finally, the upper bound of the threshold probability can be formulated as

$$F_{\text{th}}(n, d) = 1 - \binom{n}{\lfloor \frac{d-1}{2} \rfloor + 1}^{-1/\lfloor \frac{d-1}{2} \rfloor}. \tag{7.41}$$

For example, the threshold for a distance-3 colour code having a quantum coding rate of $r_Q = 1/7$ based on Fig. 7.18(a) is $F_{\text{th}} = 0.942$, while using the upper bound approximation of the fidelity threshold in Eq. (7.41) we have $F_{\text{th}} = 0.952$. For the distance-3 rotated-surface code, surface code and toric code, the threshold fidelity

values based on Fig. 7.18(b), 7.18(c) and 7.18(d) are $F_{\rm th} = 0.968$, $F_{\rm th} = 0.986$ and $F_{\rm th} = 0.993$, respectively. By using the approximation of Eq. (7.41), the fidelity threshold upper bounds are given by $F_{\rm th} = 0.972$, $F_{\rm th} = 0.987$ and $F_{\rm th} = 0.994$, respectively for the distance-3 rotated-surface code, surface code and toric code. Here, we used the family of QTECCs as our representative examples, while the threshold fidelity of Eq. (7.41) is generically applicable for any QSCs using hard-decision syndrome decoding. Ultimately, the QTECCs are capable of reducing the quantum decoherence, which is demonstrated both by the QBER reduction and the fidelity improvement attained.

7.5 Summary and Conclusions

We portrayed the evolution of the topological error correction codes designed in the classical domain to that of their quantum-domain dual pairs. We showed that by arranging the bits of the codeword on a lattice structure in the classical domain provides a beneficial inherent error correction capability. Furthermore, for a long codeword, the classical topological error correction codes correspond to the family of LDPC codes exhibiting attractive properties, such as unbounded minimum distance as a function of the codeword length, structured construction and a coding rate of $r = 1/2$. By contrast, the quantum topological error correction codes are more suitable for applications requiring short to moderate codeword lengths, since the quantum coding rate of QTECCs tends to zero for a long codeword. We characterized the performance of QTECCs in the face of the quantum depolarizing channel in terms of the QBER attained. First, we showed that QTECCs are highly degenerate quantum codes, therefore the classical simulation is only capable of portraying the performance of QTECCs without considering the degeneracy property. Secondly, we demonstrated that increasing the minimum distance of the QTECCs improves the QBER performance. Additionally, we normalized the performance by considering the coding rate, which was achieved by defining the *distance from the hashing bound*. Explicitly, we have shown that the growth of minimum distance of QTECCs upon increasing the codeword length is not fast enough to compensate for the increased codeword length. Consequently, the QBER performance of QTECCs gradually tends to the *ultimate distance from the hashing bound*. Finally, we determined the fidelity threshold for QSCs based on hard-decision syndrome decoding, which represents the minimum fidelity value required for a quantum system to glean benefits from QSCs. Ultimately, the employment of QSCs will improve the reliability of quantum computers.