Error Correction for Reliable Quantum Computing

Error Correction for Reliable Quantum Computing

by

Patricio Fuentes Ugartemendia

A dissertation submitted to the
TECNUN - SCHOOL OF ENGINEERING
in partial fulfillment of the requirements for the Degree of
DOCTOR OF PHILOSOPHY

Under the supervision of:
Professor Pedro M. Crespo Bofill

University of Navarra–TECNUN
School of Engineering
2021

Doctoral Dissertation by the University of Navarra-TECNUN

Donostia-San Sebastián

For my parents Armando and Pili, my sister Idoia, and my dog Moose.

For Maria, who I would never have met had I not pursued this PhD.

Acknowledgements

It is with great pride and no small sense of accomplishment that I present to you my PhD dissertation. Although my name is the one that appears on the cover, by no means should the work in this document be attributed only to me. Writing a PhD dissertation is no small feat, in fact I have seen it compared to bearing children (an exaggeration certainly...), and one that cannot possibly be achieved alone. For this reason, it is only appropriate that, prior to diving into the contents of this thesis, all those involved in this process be appropriately thanked and their contributions acknowledged. I must, however, make the disclaimer that I have unchained the literary demon that lives within me to write this section. Readers, proceed at your own peril.

It should come as no surprise that the first person to whom I must express my gratitude is my supervisor Professor Pedro Crespo Bofill. He has been present throughout all my academic career-defining moments; having supervised my bachelor thesis, my master's thesis and now my PhD thesis. I have actually known Pedro since I first set foot in Tecnun, as, by what I now believe to have been fate, he became my academic advisor in my first year as an undergraduate student. In all this time, Pedro has been nothing but a joyful and motivating presence, his support and belief in me never wavering. I also find it impossible to overstate his commitment to his students; even at the most improper of times, he has always been willing to spare a moment to give me guidance and advice. His technical expertise is folklore at our university, but only as a PhD student have I truly come to comprehend the depths of his genius; how one can know so much about such diverse fields of science I know not. For giving me the chance to pursue my academic dreams and for so many other things, thank you Pedro.

It is only fitting that the second person I thank be my colleague and co-author Josu Etxezarreta Martinez. He is the pioneer of quantum information at our university and the one person that managed to sell me on this

fascinating realm. Conducting our research and our dissertations together has been one of the highlights of my PhD journey and one of the primary reasons for my academic success. Besides his scholarly contributions to my work, I must also thank Josu for an innumerable amount of non-technical things. Because there are too many to mention herein, I will simply say the following: May our academic prowess continue to increase as a function of our coffee consumption, and may you never cease to "*yayeet*" and "*yeah buddy!*". For your invaluable aid and for becoming one of my closest friends, thank you Josu.

To finish with my fellow co-authors, I must now thank Professor Javier Garcia Frias at the University of Delaware. This dissertation would be nowhere near its current form if not for his involvement. I also feel obliged to thank Xabier Insausti and Marta Zárraga at my own department (department of Mathematical Principles). They have given me the chance to learn concepts completely outside the scope of quantum information and have allowed me to participate and contribute towards a scientific publication on a topic that was completely alien to me a year back. Moreover, it would be unjust to complete this paragraph without mention of Professor Jesús Gutiérrez. Despite not being directly involved in my thesis, it is because of him that I have been able to work within the department of Mathematical Principles. For your help and patience, I thank you all.

At this point, it is time for me to thank the past and present members of office D15. It is only right that I begin with Imanol Granada, a.k.a, "*the Prophet*" or "*Sir Safemoon*", the man responsible for bringing forth what can only be described as the biggest personal economic bonanza I have ever seen. Know that you are dearly missed in D15 and that I cannot wait to see what lies ahead in our joint blockchain endeavours. For bringing me into the cryptoverse and lowering my aversion to risk, thank you Imanol. Next we have Fernando Rosety, another member who no longer dwells among us in D15. The office is not the same without your humming of classical music. While some leave, others come, and so I now move to the newest member of our office, Toni de Martí. Although I have only known him for a scant three months, I have never met anyone with such a surprising array of stories and anecdotes. I must also mention Iñigo Barasoain and Fran Velásquez. The former I thank for his respect (despite having ascended to the higher plane of mathematics he continues to engage with us lowling engineers) and the latter for his cheerfulness and his innate ability to take jokes.

Along these lines, I must now mention all those friends and acquaintances at the university who, perhaps unbeknownst to them, have helped me along the way. First comes Paul Zabalegui. Our friendship began in summer swim school when we were about ten years old and although we lost touch after it finished, fate would have us meet at university once again. There are few people with whom I share so many of my passions and I feel fortunate to count him among my dearest friends. Next we have Unai Ayucar. Although it may border on insanity at times, his untethered imagination has always motivated me to revise my opinion of what is possible in life. This unfettered creativity also extends to his cooking, as only he could possibly think it a good idea to fry *paella* in litres of olive oil. Last among my estranged classmates is Daniel Talan. Despite being in Switzerland, our voice calls and discussions have become a welcome and enjoyable pastime. Thank you.

I continue at the risk of making these acknowledgements longer than the dissertation itself, but it would be an injustice to leave the following people without mention. All my friends and teammates at Txuri-Urdin. Through the rough patches of our first few seasons to winning 3 league titles in a row, I will forever cherish the memories of our time together. In particular, I want to thank Luis Gimenez, Lucas Serna, Pablo Zaballa, Mikel Mendizabal, and Borja Aizpurua personally. You have done more for this thesis than you know. In similar fashion, I also want to give thanks to my teammates, coaches, and staff at the Spanish national team. Actually, I should just thank the game of hockey itself. Even after twenty two years of playing, nothing makes me feel more alive than stepping on the ice. The game has blunted the edge of many a sharp knife in my life and its capacity to distract me has been invaluable to my success as a researcher. For all that you have given me, thank you hockey.

The game took me to Saint Andrew's College and so we now turn to that marvelous place. I would not be who I am today had I not spent two years at SAC. True to its motto, the place molded me into a significantly more mature individual and taught me skills that have shined with brilliance during my time as a PhD student. To all my teachers, coaches, and friends at SAC, thank you. Special shoutouts to my friends in 1st Hockey and the so-called "Dawgz": Graham, Humza, David, YoungWoo, West, and Andy. Although we have not seen each other in a long time, I know our friendship still runs as deep as it did when we gamed away our days at the Manor. Following this theme of childhood friends, thank you to Isma, Guille, Luken, and Andrey. From the day we first met in kindergarten at

Saint Patrick's, our passion for sports and later on fitness kept us united. I pray that our games of *pádel* never stop being so competitive.

Barring chance or extremely good fortune, those I will thank next will likely never read this dissertation. Thank you to R. A Salvatore, Steven Erikson, Brandon Sanderson, Patrick Rothfuss, and all those other authors from whom I have learned so much. Thank you also to Satoshi Nakamoto. I do not doubt that your creation will change the world for the better. I must also thank the Counting Crows, Rise Against, Machine Gun Kelly, and the Chainsmokers for motivating me in all those instances when my discipline evaporated. In relation to this, I must also thank the people at Youtube and Nespresso (and Iñigo Gutierrez by association); procrastination via the classic video and black coffee combo will never get old. Finally, a special shoutout to all those online meetings, home workouts, and quarantine periods brought to me by the COVID-19 pandemic. Somehow, throughout this entire ordeal, my work has not been impeded at any point in time. I have been extremely fortunate.

Having left the best for last, I must now turn to those closest to my heart. After spending days thinking about how to thank my parents appropriately, I found it best to use the following quote by the writer Chuck Palahniuk: "First your parents, they give you your life, but then they try to give you their life". Despite this, I am still left with a feeling of insufficiency. I guess my parents have done so much for me that it is not possible to put my gratitude into words. *Gracias attatto y amatxo por todo.* The same goes for my sister Idoia, to whom I owe more than can be expressed. Thank you for always being there, for always having time to talk, and most of all, for always being willing to listen. Through thick and thin, your presence has never failed to remind me that there is always light at the end of the tunnel. Lastly, thank you Moose for being such an energetic furball and for greeting me at the door everyday as if you had not seen me in years.

Alas, it is now time for me to thank one final person. Maria, you, above anyone else, have lived through the highs and the lows of my PhD. You have seen the extent to which manuscript revisions can frustrate me and how much ideas can consume me. You remained at my side through it all, and only through your intervention have I managed to complete my work. *Maria, tesian gertatu zaidan gauzarik onena zara.*

That was a lengthy acknowledgements section, I know. Even so, I cannot help but feel like I have not thanked all the people that have helped me along the way. In any case, I think that it is high time for me to stop torturing the reader with my sorry attempts at Shakesperean prose. To all

those I have named herein and those who I have surely forgotten, thank you for helping me navigate the trials and tribulations of this voyage, I could not have done it without you.

Abstract

Quantum computers herald the arrival of a new era in which previously intractable computational problems will be solved efficiently. However, quantum technology is held down by decoherence, a phenomenon that is omnipresent in the quantum paradigm and that renders quantum information useless when left unchecked. The science of quantum error correction, a discipline that seeks to combine and protect quantum information from the effects of decoherence using structures known as codes, has arisen to meet this challenge. Stabilizer codes, a particular subclass of quantum codes, have enabled fast progress in the field of quantum error correction by allowing parallels to be drawn with the widely studied field of classical error correction. This has resulted in the construction of the quantum counterparts of well-known capacity-approaching classical codes like sparse codes and quantum turbo codes. However, quantum codes obtained in this manner do not entirely evoke the stupendous error correcting abilities of their classical counterparts. This occurs because classical strategies ignore important differences between the quantum and classical paradigms, an issue that needs to be addressed if quantum error correction is to succeed in its battle with decoherence. In this dissertation we study a phenomenon exclusive to the quantum paradigm, known as degeneracy, and its effects on the performance of sparse quantum codes. Furthermore, we also analyze and present methods to improve the performance of a specific family of sparse quantum codes in various different scenarios.

Research Papers

This thesis is the culmination of two and a half years of work within the *Mathematical Principles group* of the Department of Biomedical Engineering and Sciences at the *Tecnun - School of Engineering (University of Navarra)*. Throughout this time, I have published a number of research papers, detailed below in chronological order. In terms of their relationship to this dissertation, this thesis is mostly comprised of the results obtained in those articles that I have first-authored myself (shown in blue). To provide context, I have included a brief summary of the other works that I have co-authored in Chapter 8.

- **P. Fuentes**, J. Etxezarreta Martinez, P. M. Crespo, and J. Garcia-Frias, "Approach for the construction of non-Calderbank-Steane-Shor low-density-generator-matrix based quantum codes," *Phys. Rev. A*, vol. 102, pp. 012423, 2020. doi:10.1103/PhysRevA.102.012423.

- **P. Fuentes**, J. Etxezarreta Martinez, P. M. Crespo, and J. Garcia-Frías, "Performance of non-CSS LDGM-based quantum codes over the Misidentified Depolarizing Channel," *IEEE International Conference on Quantum Computing and Engineering (QCE20)*, 2020. doi:10.1109/QCE49297.2020.00022.

- J. Etxezarreta Martinez, **P. Fuentes**, P. M. Crespo, and J. Garcia-Frias, "Pauli Channel Online Estimation Protocol for Quantum Turbo Codes," *IEEE International Conference on Quantum Computing and Engineering (QCE20)*, 2020. doi: 10.1109/QCE49297.2020.00023.

- J. Etxezarreta Martinez, **P. Fuentes**, P. M. Crespo, and J. Garcia-Frias, "Approximating Decoherence Processes for the Design and Simulation of Quantum Error Correction Codes in Classical Computers," *IEEE Access*, vol. 8, pp. 172623-172643, 2020. doi: 10.1109/ACCESS.2020.3025619.

- **P. Fuentes**, J. Etxezarreta Martinez, P. M. Crespo, and J. Garcia-Frias, "Design of LDGM-based quantum codes for asymmetric quantum channels," *Phys. Rev. A*, vol. 103, pp. 022617, 2021. doi: 10.1103/PhysRevA.103.022617.

- **P. Fuentes**, J. Etxezarreta Martinez, P. M. Crespo, and J. Garcia-Frias, "Degeneracy and its impact on the decoding of sparse quantum codes," *IEEE Access*, vol. 9, pp. 89093-89119, 2021. doi: 10.1109/ACCESS.2021.3089829.

- J. Etxezarreta Martinez, **P. Fuentes**, P. M. Crespo, and J. Garcia-Frias, "Time-varying quantum channel models for superconducting qubits," *npj Quantum Information*, vol. 7, no. 115, 2021. doi: 10.1038/s41534-021-00448-5.

- **P. Fuentes**, J. Etxezarreta Martinez, P. M. Crespo, and J. Garcia-Frias, "On the logical error rate of sparse quantum codes," submitted to *IEEE Trans. on Quantum Eng.*, 2021. arXiv: 2108.10645v2.

- J. Etxezarreta Martinez, **P. Fuentes**, P. M. Crespo, and J. Garcia-Frias, "Quantum outage probability for time-varying quantum channels," submitted to *Phys. Rev. A*, 2021. arXiv:2108.13701.

Glossary

A list of used acronyms is provided below.

BER	*Bit Error Rate*
BP	*Belief Propagation*
BSC	*Binary Symmetric Channel*
BWML	*Bit-Wise Maximum Likelihood*
CSS	*Calderbank-Shor-Steane*
EFB	*Enhanced Feedback*
ECC	*Elliptic Curve Cryptography*
LDGM	*Low Density Generator Matrix*
LDPC	*Low Density Parity-Check*
LLR	*Log-Likelihood Ratio*
LUT	*Look-Up Table*
NP	*Non-deterministic Polynomial*
OSD	*Ordered Statistics Decoder*
PCM	*Parity Check Matrix*
QCC	*Quantum Convolutional Code*
QEC	*Quantum Error Correction*
QLDGM	*Quantum Low Density Generator Matrix*
QLDPC	*Quantum Low Density Parity Check*
QMLD	*Quantum Maximum Likelihood Decoding*
QPCM	*Quantum Parity Check Matrix*
QSC	*Quantum Stabilizer Code*
QTC	*Quantum Turbo Code*
RSA	*Rivest, Shamir and Adleman*
SP(A)	*Sum-Product (Algorithm)*
WER	*Word Error Rate*
i.i.d.	*independent and identically distributed*

Notation

Although all symbols are defined at their first appearance, some are repeated throughout the dissertation. A list of the most frequent symbols is provided below.

$\mathcal{H}_2$	Complex Hilbert space of dimension 2.
$\otimes$	Tensor product.
I	Single qubit Pauli matrix (identity).
X	Single qubit Pauli matrix (bit flip).
Z	Single qubit Pauli matrix (phase flip).
Y	Single qubit Pauli matrix (bit & phase flip).
$\Pi^{\otimes N}$	Set of N-fold tensor products of single qubit Pauli operators.
$\mathcal{G}_N$	N-fold Pauli group.
$\bar{\mathcal{G}}_N$	Effective N-fold Pauli group.
ξ	Quantum channel.
ξ_P	Pauli channel.
$\lvert \cdot \rvert$	Absolute value of a number or carnality of a set.
$\mathcal{S}$	Stabilizer set defined over the Pauli group.
$\bar{\mathcal{S}}$	Stabilizer set defined over the effective Pauli group.
k	Length of information word.
n	Length of a codeword.
$\mathcal{C}(\bar{\mathcal{S}})$	Stabilizer code.
$\lvert\psi\rangle$	Quantum Information state.
$\lvert\bar{\psi}\rangle$	Encoded quantum state.
$\mathbb{F}_2^{2N}$	Set of length $2N$ binary vectors.
$\odot$	Symplectic product.
$\oplus$	Modulo 2 sum.
$\circledast$	Modulo 2 product.
$\mathbf{w}$	Quantum syndrome.
$\cdot$	Group operation over the Pauli Group.
$\star$	Group operation over the effective Pauli group.

β	Symplectic isomorphism/map.
$\overline{\mathcal{Z}}(\overline{\mathcal{S}})$	Effective centralizer of a stabilizer.
$\overline{\mathcal{N}}(\overline{\mathcal{S}})$	Effective normalizer of a stabilizer.
$\mathbf{S}_v$	Stabilizer operator.
$\mathbf{T}_i$	Pure error operator & effective centralizer coset representative.
$\mathbf{L}_j$	Logical operator & stabilizer coset representative.
$C^{\perp}$	Dual of a an error correction code.

Contents

1 Introduction **1**
- 1.1 Motivation and Objectives . . . 5
  - 1.1.1 Performance of QLDPC codes across the landscape of quantum channels . . . 6
  - 1.1.2 Understanding and exploiting degeneracy . . . 7
- 1.2 Outline and Contributions of the Thesis . . . 7
  - 1.2.1 Chapter 2: Why quantum? . . . 8
  - 1.2.2 Chapter 3: Preliminaries on Quantum Information and Classical Error Correction . . . 8
  - 1.2.3 Chapter 4: Degeneracy and its impact on Decoding (Degeneracy I) . . . 10
  - 1.2.4 Chapter 5: Detecting Degeneracy and Improved Decoding Strategies (Degeneracy II) . . . 10
  - 1.2.5 Chapter 6: Non-CSS QLDPC codes (QLDPC I) . . . 11
  - 1.2.6 Chapter 7: Performance of QLDPC codes over Pauli channels (QLDPC II) . . . 12
  - 1.2.7 Chapter 8: Quantum Turbo Codes and Time-varying quantum channels . . . 13
  - 1.2.8 Chapter 9: Conclusion and Future Work . . . 14
- 1.3 How to read this thesis . . . 14

2 Why Quantum? **17**

3 Preliminaries **27**
- 3.1 Quantum Information . . . 27
  - 3.1.1 Postulates of Quantum Mechanics . . . 28
  - 3.1.2 The Qubit . . . 29
    - 3.1.2.1 The Bloch Sphere . . . 32
  - 3.1.3 Entanglement . . . 34

3.1.4 Quantum Noise 35
3.1.4.1 Unitary operators 36
3.1.4.2 Diagrammatic notation of quantum systems 44
3.1.4.3 N-qubit generalization & The Pauli Group 45
3.1.4.4 Error discretization 47
3.1.4.5 Quantum Channels 49
3.2 Classical Error Correction 53
3.2.1 Linear Block Codes 53
3.2.2 Low Density Parity Check codes 55
3.2.2.1 Factor Graphs 55
3.2.2.2 Syndrome-based decoding of LDPC Codes 63

4 Degeneracy and its impact on Decoding 67
4.1 Stabilizer codes 69
4.1.1 The effective Pauli Group 69
4.1.2 The Symplectic Representation 73
4.2 Stabilizer-based error correction 75
4.2.1 The Stabilizer Group 76
4.2.1.1 Partition of $\mathbb{F}_2^{2N}$ into cosets 77
4.2.1.2 Partition of $\mathbb{F}_2^{2N}$ into cosets of $\Gamma_{\mathcal{R}}$ 78
4.2.1.3 Partition of $\Gamma_{\mathcal{R}}$ into cosets of $\mathcal{R}$ 79
4.2.1.4 Partition of $\mathbb{F}_2^{2N}$ into cosets of $\mathcal{R}$ 80
4.2.2 Partition of $\overline{\mathcal{G}}_N$ into cosets 80
4.2.2.1 Partition $\overline{\mathcal{G}}_N/\overline{\mathcal{Z}}(\overline{\mathcal{S}})$ 82
4.2.2.2 Partition $\overline{\mathcal{Z}}(\overline{\mathcal{S}})/\overline{\mathcal{S}}$ 82
4.2.3 Construction of stabilizer codes 84
4.2.3.1 Pure Errors, Logical Operators & Encoded Pauli operators 84
4.2.3.2 Error detection using stabilizer codes . . . 87
4.2.3.3 Quantum Parity Check Matrix of a Stabilizer Code 89
4.2.4 Decoding Quantum Stabilizer codes 91
4.2.4.1 Optimal decoding of quantum stabilizer codes 92
4.2.4.2 Optimal vs SPA decoding of quantum stabilizer codes 94
4.3 Degeneracy and why it arises 97
4.3.1 Degeneracy in sparse quantum codes 98
4.3.2 End-to-end errors 99
4.3.3 A useful example 102
4.4 Chapter Summary 110

5 Detecting Degeneracy and Improved Decoding Strategies 111
5.1 Performance metrics . . . 112
5.1.1 Physical and Logical Error rate . . . 112
5.1.2 Discriminating between different types of end-to-end errors . . . 115
5.2 An algebraic perspective on end-to-end degenerate errors . 116
5.2.1 Classical coding inspired strategy . . . 118
5.2.2 Detecting end-to-end degenerate errors using encoded Pauli operators . . . 119
5.3 Frequency of each type of end-to-end error . . . 121
5.4 Improved decoding strategies for QLDPC codes . . . 125
5.5 Chapter summary . . . 128

6 Non-CSS QLDPC codes 129
6.1 Preliminaries . . . 131
6.1.1 Pauli channels . . . 132
6.1.1.1 Depolarizing channel . . . 132
6.1.1.2 i.i.d. X/Z Channel . . . 132
6.1.2 The Hashing bound . . . 133
6.1.2.1 Distance to the Hashing bound . . . 133
6.1.3 CSS codes . . . 134
6.1.4 Systematic classical LDGM codes . . . 134
6.2 CSS LDGM-based codes . . . 136
6.3 Design of non-CSS LDGM-based codes . . . 141
6.3.1 Proposed procedure for the construction of non-CSS QLDGM codes . . . 142
6.3.1.1 Method 1: Syndrome node combination . . 142
6.3.1.2 Method 2: Syndrome node combination + removal of s_A nodes . . . 145
6.3.1.3 Non-CSS QPCM . . . 146
6.3.1.4 Mixture of both methods . . . 147
6.3.2 Decoding non-CSS QLDGM codes . . . 147
6.3.3 Rate considerations . . . 148
6.4 Simulation Results . . . 149
6.4.1 i.i.d. X/Z channel - Non-CSS codes based on individual regular LDGM codes . . . 150
6.4.1.1 Non-CSS codes derived using method 1 . . 151
6.4.1.2 Non-CSS codes derived using method 2 . . 151
6.4.2 i.i.d. X/Z Channel - Non-CSS codes based on the parallel concatenation of LDGM codes . . . 154

6.4.3 Depolarizing channel 160
6.4.3.1 Distance to the theoretical limit 160
6.4.3.2 Comparison with existing QLDPC schemes 163
6.5 Chapter summary 166

7 Performance of QLDPC codes over Pauli channels 167
7.1 Performance of non-CSS QLDPC codes over the misidentified depolarizing channel 168
7.1.1 Quantum Channel Identification 169
7.1.1.1 Off-line Estimation Method 171
7.1.1.2 Online Estimation Method 174
7.2 Design of asymmetric QLDGM codes 177
7.2.1 Realistic Pauli Channel model 179
7.2.2 Performance comparison with other QLDPC codes . 180
7.2.3 Asymmetric QLDGM CSS codes 183
7.2.3.1 Adaptation of symmetric CSS QLDGM codes to the Pauli channel 184
7.2.4 Simulations 188
7.2.4.1 Performance over the asymmetric Pauli Channel 189
7.2.4.2 Optimization of the Z decoder 197
7.2.4.3 Simulations and adaptation to other asymmetric parameter values 206
7.2.4.4 Distance to the Hashing bound of the Pauli channel model for asymmetry 209
7.3 Chapter summary 213

8 QTCs and Quantum Channels 215
8.1 Online Estimation Protocol for QTCs over Pauli channels . 216
8.2 Approximating decoherence processes for the design of QECCs on classical computers 217
8.3 Time-varying quantum channel models for superconducting qubits 220
8.4 Quantum outage probability for time-varying quantum channels 221

9 Conclusion and Future Work 227

Appendices 231

A Syndrome-based Decoding of LDPC codes **233**

B Improved Decoding Strategies for QLDPC codes **239**
B.1 Correlation Exploiting Decoder 239
B.2 Freezing Decoder 241
B.3 Random Perturbation Decoder 242
B.4 Collision Decoder 242
B.5 Enhanced Feedback Decoder 243
B.6 Supernode Decoder 245
B.7 Adjusted Decoder 246
B.8 Augmented Decoder 247
B.9 Combined Decoder 248
B.10 Ordered Statistics Decoder 249
B.11 Refined Belief Propagation Decoding 250

C Monte Carlo Simulations **253**

References **257**

CHAPTER 1

Introduction

"Veris in numeris"
Satoshi Nakamoto.

The behaviour and composition of matter in its most reduced scale has long been pondered by the scientific community. In fact, the concept of the atom dates back to the 5th century BCE, which is when the Greek philosophers Leucippus and Democritus first brought up the idea. Since then, understanding of the topic has progressed immensely, especially with the development of quantum mechanics. Unfortunately, despite our advancements, many areas in the field of physics still defy human understanding. This is, in no small part, due to the incapacity of classical computers to simulate the time evolution of subatomic systems. It was precisely for this reason that, in his revolutionary work [1], Richard Feynman posited that without devices that obeyed the eldritch laws of quantum mechanics (the fundamental theory in physics that describes the behaviour of subatomic particles) it would not be possible to accurately portray the behaviour of matter. Thenceforth, research has shown that quantum constructs have myriads of applications beyond Feynman's original proposal and that they are especially well suited to efficiently solve certain tasks which are computationally unmanageable for classical instruments. The advantages provided by these machines stem from the quantum nature of their most basic component: the quantum bit (qubit). While classical bits can only exist in

one of two states, 0 or 1, qubits manifest as a superposition of these two states, which means that they are both 0 and 1 simultaneously.

The implications that the superposition property of qubits has on the field of computation are vast. Whereas a classical N-bit register stores a single N-bit value, superposition allows an N-qubit quantum register to store 2^N states concurrently. Then, through the devious application of global function optimization techniques, the 2^N states can be evaluated *in parallel*[1] for a cost analogous to that of a single classical evaluation [2, 3]. For this reason, specific problems that are computationally hard in classical terms, such as the factorization of large numbers or performing a search through an unstructured database, become significantly less complex on quantum machines capable of running quantum algorithms [4, 5, 6]. For instance, Shor's algorithm for the factorization of prime numbers runs in polynomial time while the best known classical algorithm for this same purpose runs in exponential time [4]. Other currently known notable tasks that are better addressed using quantum technology are the discrete logarithm problem [4], Byzantine agreement [7], or parallel computation in communication networks [8, 9, 10]. Aside from the field of computation science, numerous other scientific fields stand to gain from the development of the quantum framework. A good example is the area of communication security, where the journey towards quantum secure cryptographic schemes has already begun with the proposal of the BB84 [11] and the E91 [12] protocols.

This staggering theoretical potential has transformed quantum technology into the harbinger of a new era in the fields of computation and communications, and its capability to outperform classical methods in the areas of information processing, storage, and transmission can no longer be disputed. Unfortunately, despite the scientific community's unwavering commitment to the construction of a full-scale quantum computer, devices capable of realizing the promise of quantum information science have not yet become a reality. Mostly, this can be attributed to the phenomenon

[1] The analogy with parallel computing is useful to understand the advantages that quantum computers provide. However, it must be stated that quantum computing and parallel computing are not one and the same. Quantum computing exploits the superposition property of qubits to consider the entire solution space of a particular problem simultaneously. Parallel computing makes use of multiple processors to evaluate each possible solution independently on each of them. The former is limited by the amount of qubits it can employ while the latter is limited by the number of processors at its disposal.

known as decoherence [13, 14, 15], which describes the process by which the quantum objects that store quantum information lose coherence as a consequence of their interaction with the environment. The only way to indefinitely maintain coherence requires the perfect isolation of a quantum state, a process that prohibits any interaction or manipulation of said state, which means that decoherence is unavoidable when working with quantum information. In consequence, for quantum computers to be useful, they must guarantee sufficiently long quantum information coherence time periods for practical applications. Satisfying this requirement is no simple feat, as it implies that quantum processors must function correctly even when their elemental information units suffer from decoherence effects. It is to find an answer to this dilemma that the scientific discipline known as the theory of Quantum Error Correction (QEC) has arisen; to find ways to ensure that quantum technology can operate in time intervals that are long enough for the advantages of quantum computing to come to light. In fact, the corruptive power of decoherence is strong enough to make many experts believe that, without appropriate error correction strategies, quantum computing itself hangs in the balance.

This widespread concern with regard to the achievability of quantum computing in the absence of error correction has caused the field of QEC to experience a drastic surge in popularity since the first quantum code was introduced in [13]. Significant breakthroughs have been made during this rise to fame, of which (arguably) the most important is the formulation of Quantum Stabilizer Codes (QSCs) in Gottesmans PhD thesis [16]. In said work, Gottesman shows how, by casting existing groups of classical codes into the framework of QSCs, quantum counterparts of these classical designs can be derived, effectively allowing the development of quantum coding schemes from existing classical strategies. This formalism has enabled an almost seamless transition from classical error correction to QEC and has led to the construction of many QEC code families like Quantum Reed-Muller codes [17], sparse quantum codes or Quantum Low Density Parity Check (QLDPC) codes [18, 110, 20, 21, 22, 23, 24], Quantum Convolutional Codes (QCC) [25, 26, 27], Quantum Turbo Codes (QTC) [28, 29, 30, 31] and Quantum Topological Codes [32, 33, 34, 35].

It is also pivotal for the mechanisms through which a QEC code bestows quantum information with a robust defence against quantum decoherence to be of reasonable complexity. The encoding and decoding requirements of QEC codes are of paramount importance since the quantum gates which

implement error correction operations are also faulty and may induce additional errors in the quantum information. This gives rise to the concept of fault-tolerance or fault-tolerant computing [36, 37, 38, 39], which is a term used to refer to the notion of a quantum apparatus functioning correctly despite the fact that its most basic components may sometimes be faulty themselves.

Among the aforementioned quantum code families, the quantum counterparts of sparse codes stand out as being especially well suited to implement fault-tolerant error correction methods. The sparsity of their decoding matrices implies that only a few quantum interactions per qubit are necessary in the error correction procedure, and ensures that additional quantum gate errors are avoided. In consequence, this field is evolving rapidly and numerous new construction and design methods for sparse quantum codes are being proposed [40, 41, 42, 43, 44, 45]. These codes, which are also known as QLDPC codes, can be defined as stabilizer codes with sparse generators and they can be constructed using a variety of different methods. One of the most commonly employed design strategies consists in taking classical LDPC codes as the starting point and adapting them so that they can be used in the quantum paradigm [20].

Classical LDPC codes, along with turbo codes [46], represent forms of sparse or random-like codes that can be decoded probabilistically and that are capable of approaching the theoretical communication limits of a communication channel with a reasonable decoding complexity. This stems from the fact that they provide sufficient structure for the decoding process to function correctly, while, simultaneously, the randomness involved in the design itself guarantees excellent performance. Decoding is performed by means of the Sum Product Algorithm (SPA) [47], which is a generic message passing algorithm that computes various marginal functions associated with a global function. Related decoding methodologies for probabilistic codes, such as Belief Propagation (BP) [48] or the Viterbi algorithm [49], have been shown to be specific instances of the SPA [50]. The SPA operates over tree-like graphs known as factor graphs, which are used to represent a complicated "global" function of many variables as a product of simpler "local" functions, each depending on a subset of the variables. Factor graphs express which variables are arguments of which local functions and the SPA derives its computational efficiency by exploiting the way in which the global function factors into those products of "local" functions. When the factor graph is a tree, the SPA converges to the exact solution in a time

bounded by the tree's depth. In scenarios where the algorithm does not converge, i.e., when the factor graph has loops, it still represents a good heuristic method to implement sub-optimal decoding if the loops are long enough. In [51] and [52], it was shown that decoding stabilizer QEC codes on memoryless quantum channels can be defined as the execution of the SPA over a typically loopy factor graph.

Being linear block codes, classical LDPC codes are designed by defining a set of parity check equations that involve information bits. To guarantee their low density, each equation involves a small number of bits, and each particular bit is involved in a reduced number of equations. This set of constraints is defined by means of a Parity Check Matrix (PCM) where each row denotes a parity check equation and each column denotes a coded bit. The PCM can also be represented by means of a factor graph, where two types of interconnected nodes, variable nodes and parity check nodes, represent each of the columns and each of the rows of the PCM, respectively. Because of the low density requirements imposed in the design of these codes, the corresponding factor graph will have a small number of loops, ensuring good performance when decoded using the SPA. Given their capacity-approaching performance under SPA decoding, as well as their potential upside to implement effective error correcting strategies, deriving good quantum sparse codes is germane to the field of QEC.

1.1 MOTIVATION AND OBJECTIVES

Two primary issues arise when designing sparse quantum codes. On the one hand, most randomly generated LDPC codes are not suitable for the quantum paradigm and so the number of good classical codes applicable to the quantum domain is reduced. On the other, decoding based on the SPA is affected by a quantum phenomenon known as *degeneracy* [53, 54, 55, 56], which has no classical equivalent. In the literature, the first problem is addressed by using constructions with stringent requirements, like the one simultaneously proposed by Calderbank, Shor, and Steane in [57, 58] (CSS codes). Unfortunately, these methods introduce particular drawbacks that further complicate the design of QLDPC codes and that, as of yet, have not been completely resolved. The second issue, which pertains to degeneracy, poses a quandary that is more difficult to give an answer to. This happens because the decoding algorithm [47, 48] used to decode degenerate QLDPC codes is designed, in principle, for a classical environment in

which degeneracy is not present, and so it will be completely blind to this phenomenon[2].

In this thesis we attempt to tackle both of these concerns: improving the performance of QLDPC codes and studying degeneracy, by establishing the following key objectives:

1.1.1 PERFORMANCE OF QLDPC CODES ACROSS THE LANDSCAPE OF QUANTUM CHANNELS

Due to the ease with which quantum codes can be built from their classical counterparts based on the CSS construction, most of the existing QLDPC codes are built using this methodology. CSS codes are a particular subset of the QSC family that provide a straightforward method to design quantum codes via existing classical codes. Although the construction introduces additional code design challenges (the classical codes must comply with a specific algebraic condition), the method ensures that the resulting quantum code is applicable in any quantum environment. Thus, codes built in this manner can be used across the entire spectrum of quantum channel models, which means that they can be studied and optimized for different practical scenarios.

Unfortunately, the CSS construction is not without its faults. Classical codes that can be employed in CSS schemes must meet specific requirements that limit the performance that is attainable with the resulting quantum code. This impediment to the performance of CSS codes, known as the *CSS lower bound* [62], implies that the best possible theoretical performance over a quantum channel cannot be met when using a CSS code. This has inspired the search for non-CSS constructions, as they are not limited by the CSS lower bound and should be able to outperform CSS codes provided that they are designed optimally. Non-CSS LDPC-based codes were proposed in [63] and [64]. Despite showing promise, these codes failed to outperform existing CSS QLDPC codes for comparable block lengths.

[2]The effects of degeneracy are mitigated in entanglement assisted schemes [59, 60, 61]. Such strategies make use of pre-shared Bell states to reduce the degenerate content of a quantum code to the point that, depending on the amount of pre-shared information, the scenario becomes increasingly similar to that of classical decoding [29]. However, unassisted coding strategies that cannot make use of entanglement experience the full-fledged impact of this phenomenon.

Against this backdrop, it is clear that there is ample room for scientific growth within the niche of CSS-based design of sparse quantum codes. However, the overarching nature of our first objective requires that we addresses its constituent parts: the design of non-CSS codes and the performance of CSS codes over different quantum channel models, separately. For this reason, in this dissertation we restrict the study of non-CSS QLDPC designs to the framework of the depolarizing channel (the most extended quantum channel model) and we use the widely-established CSS construction technique to build codes and optimize them for less conventional quantum channel models.

1.1.2 UNDERSTANDING AND EXPLOITING DEGENERACY

When quantum stabilizer codes built from sparse classical codes are employed in the quantum paradigm, they are impacted by a phenomenon known as degeneracy [53, 54, 55, 56], which has no classical equivalent. This causes stabilizer codes to exhibit a particular coset structure in which multiple different error patterns act identically on the transmitted information [58, 65, 66]. Although the manifestation of degeneracy in the design of sparse quantum codes and its effects on the decoding process has been studied extensively [16, 55, 56, 67, 68, 69, 70, 71], especially for QTCs and quantum topological codes [29, 68, 72, 73, 74], it remains a somewhat obscure topic in the literature. This can be attributed to the varying and sometimes inconsistent notation and the oft confusing nature of the notion of degeneracy itself. In consequence, although degeneracy should theoretically improve performance, limited research exists on how to quantify and exploit this phenomenon in the framework of QLDPC codes.

For these reasons, the second objective of this thesis is to accurately characterize the phenomenon of degeneracy as it pertains to sparse quantum codes. In a similar manner to the first objective, we will employ a two pronged strategy to realize this goal: first, we seek to completely describe degeneracy and the mechanisms that govern its behaviour, and then we use this framework to devise methods to diagnose and exploit its manifestation.

1.2 OUTLINE AND CONTRIBUTIONS OF THE THESIS

The contents of this thesis are structured in a way that, according to the authors view, provides the best possible reading experience. This means

that instead of following the chronology of the research, the chapters of the dissertation have been placed so that understanding of each particular chapter is facilitated by those that precede it. A timeline of how the research that comprises this thesis actually evolved can be seen in Figure 1.1.

The thesis begins with a brief commentary on quantum computers and what they excel at in Chapter 2. This chapter discusses concepts related to complexity theory that shed light on the importance of quantum computing and quantum error correction. In Chapter 3, an introduction to fundamental concepts and preliminaries related to quantum information science and classical error correction is provided. From here on out, the remaining chapters of the dissertation can be grouped into two distinct parts, each one related to the objetives described previously in subsection 1.1: Part 1 (Chapters 4 and 5) discusses the degeneracy phenomenon and its impact on sparse quantum codes, and Part 2 (Chapters 6 and 7) focuses on the design of non-CSS QLDPC codes and the optimization of CSS QLDPC schemes for different quantum channel models. Then, in the final two chapters of the thesis, a summary of co-authored research (Chapter 8) and possible future work (Chapter 9) is provided.

1.2.1 CHAPTER 2: WHY QUANTUM?

Chapter 2 seeks to clearly present the arguments in favour of quantum computing. For this purpose, it discusses why quantum computers are better suited than their classical counterparts to perform specific tasks like the factorization of prime numbers. It also delves into the technologies that are currently being employed to physically implement quantum computers and provides an overview of the major players in the field of experimental quantum computing.

This chapter is meant as a cursory presentation on the advantages of quantum computing. People that are familiar with the field can skip this section.

1.2.2 CHAPTER 3: PRELIMINARIES ON QUANTUM INFORMATION AND CLASSICAL ERROR CORRECTION

Herein we lay the groundwork necessary to follow the discourse in subsequent chapters by introducing basic concepts of quantum information

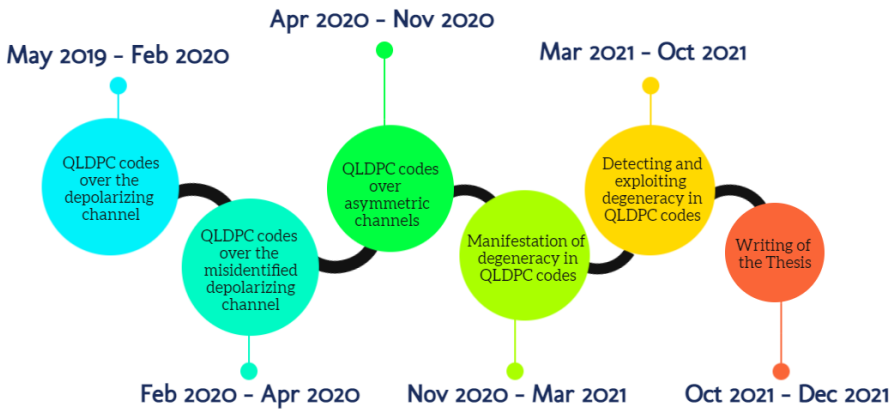


Figure 1.1: *Timeline of the research carried out in this thesis.*

theory and classical error correction. The chapter is divided into two sections: section 3.1 which is devoted to quantum information theory and section 3.2 which is dedicated to classical error correction. The chapter commences in section 3.1.2 with a discussion on the qubit. Then, in Section 3.1.3, we introduce the concept of entanglement, which has become almost folklore in the quantum information community. Following this, we present the notions of unitary operators and gates in the quantum domain in section 3.1.4. This section on quantum information theory is concluded with an introduction to the Pauli group and a brief overview of the most common quantum channel models.

Following section 3.1 we turn to the realm of classical error correction. Section 3.2 is comprised of subsection 3.2.1, which introduces the concept of linear block codes, and subsection 3.2.2, which presents the family of LDPC codes and other basic communication and graph theory notions like factor graphs and iterative decoding.

Although this chapter seeks to make the dissertation self-contained, many other important concepts related to quantum information and classi-

cal error correction have not been included. If needed, we refer the reader to [75] and [76] for further detail on quantum information and classical error correction, respectively. Furthermore, because scientists from various different fields of study are involved in quantum computing, it may be that either section 3.1 or section 3.2 of this chapter is well known to many readers. In such a case, those familiar with a particular section should skip it and move on to the next section or to Chapter 4.

1.2.3 CHAPTER 4: DEGENERACY AND ITS IMPACT ON DECODING (DEGENERACY I)

Chapter 4 studies the phenomenon of degeneracy from the perspective of group theory with the purpose of completely characterizing it in the context of sparse quantum codes. The chapter begins by casting the notion of stabilizer codes into a group theoretical framework and using it to perform necessary distinctions between ideas that are sometimes misunderstood in the field of QEC. Following this, the classical and quantum decoding problems are presented, and their similarities and differences are discussed. Then, we proceed by studying the emergence of degeneracy and the impact that disregarding its existence has on the decoding process. The chapter is closed with a detailed example that serves to illustrate many of these concepts.

The contents of this chapter are based on the following journal paper:

- P. Fuentes, J. Etxezarreta Martinez, P. M. Crespo, and J. Garcia-Frias, "Degeneracy and its impact on the decoding of sparse quantum codes," *IEEE Access*, vol. 9, pp. 89093-89119, 2021. doi: 10.1109/ACCESS.2021.3089829.

1.2.4 CHAPTER 5: DETECTING DEGENERACY AND IMPROVED DECODING STRATEGIES (DEGENERACY II)

This chapter considers the issue of detecting the presence of degeneracy when using sparse quantum codes. We begin the chapter by discussing the reasons for which limited research exists on how to quantify the true impact that the degeneracy phenomenon has on QLDPC codes. Then, we discuss why two different performance assessment metrics have been used in the literature of sparse quantum codes, and we show how only one of them

provides an accurate portrayal of performance. Following this, we devise a method to assess the effects of degeneracy on sparse quantum codes and we explain another previously existing strategy to do so. Finally, we use our strategy to analyze the frequency with which different types of errors occur when using sparse quantum codes and we provide insight on how the design and decoding of these codes can be improved.

The method proposed in this chapter as well as most of its contents have been published in the following journal paper:

- P. Fuentes, J. Etxezarreta Martinez, P. M. Crespo, and J. Garcia-Frias, "On the logical error rate of sparse quantum codes," submitted to *IEEE Trans. on Quantum Eng.*, 2021. arXiv: 2108.10645v2.

1.2.5 CHAPTER 6: NON-CSS QLDPC CODES (QLDPC I)

Most QLDPC codes are built by casting classical LDPC codes in the framework of stabilizer codes, which enables the design of quantum codes from any arbitrary classical binary and quaternary codes. Some of the best performing QLDPC codes are obtained by combining the CSS construction with Low Density Generator Matrix (LDGM) codes, which are a particular type of LDPC code. However, CSS constructions are limited by an unsurpassable bound, which has inspired the search for non-CSS constructions as they should theoretically be able to outperform CSS codes. In this chapter, we show how the nature of CSS designs and the manner in which they must be decoded limits the performance that they can achieve. Then, we introduce a non-CSS quantum code construction that we derive from the best CSS QLDGM construction that can be found in the literature. We close the chapter by showing how codes designed using this method outperform CSS QLDGM codes and most other QLDPC codes of comparable complexity.

The work that appears in this chapter has been published in the following journal paper:

- P. Fuentes, J. Etxezarreta Martinez, P. M. Crespo, and J. Garcia-Frias, "Approach for the construction of non-Calderbank-Steane-Shor low-density-generator-matrix based quantum codes," *Phys. Rev. A*, vol. 102, pp. 012423, 2020. doi:10.1103/PhysRevA.102.012423.

1.2.6 CHAPTER 7: PERFORMANCE OF QLDPC CODES OVER PAULI CHANNELS (QLDPC II)

Most of the research related to QLDPC codes has been conducted under the tacit premise that perfect knowledge of the quantum channel in question is available. In reality, such a scenario is highly unlikely, which makes it necessary to analyze the change in the behaviour of these codes as a function of the existing information about the quantum channel. In the first section of this chapter, section 7.1, we study the behaviour of the non-CSS QLDGM codes introduced in Chapter 6 under the umbrella of channel mismatch, a term that makes reference to a scenario in which the true channel information and that which is known is different.

Generally, it has also been the norm in the literature of QEC to consider only the depolarizing channel: the symmetric instance of the generic Pauli channel, that incurs bit-flips, phase-flips, or a combination of both with the same probability. However, because of the behaviour of the materials they are built from, it is not appropriate to employ the depolarizing channel model to represent specific quantum devices. Instead, they must be modelled using a different quantum channel capable of accurately representing asymmetric scenarios in which the likelihood of a phase-flip is higher than that of a bit-flip. Thus, in the second section of this chapter, section 7.2, we study ways in which to adapt the design of the CSS LDGM-based codes discussed in Chapter 6 to asymmetric quantum channels.

The work that comprises this chapter has been published in the following papers:

- P. Fuentes, J. Etxezarreta Martinez, P. M. Crespo, and J. Garcia-Frías, "Performance of non-CSS LDGM-based quantum codes over the Misidentified Depolarizing Channel," *IEEE International Conference on Quantum Computing and Engineering (QCE20)*, 2020. doi:10.1109/QCE49297.2020.00022.

- P. Fuentes, J. Etxezarreta Martinez, P. M. Crespo, and J. Garcia-Frias, "Design of LDGM-based quantum codes for asymmetric quantum channels," *Phys. Rev. A*, vol. 103, pp. 022617, 2021. doi: 10.1103/PhysRevA.103.022617.

1.2.7 CHAPTER 8: QUANTUM TURBO CODES AND TIME-VARYING QUANTUM CHANNELS

In this chapter, we provide a summary of other research that the author has co-authored and participated in during this PhD dissertation. Although outside the niche of sparse quantum codes, this research is also related to QEC. It is primarily the work of the author's colleague and first author of the journal papers, Josu Etxezarreta Martinez. For this reason, only a succinct overview is contained within this chapter and readers are referred to the first author's own PhD dissertation for discussions that do this work justice. Chapter 8 is comprised of four sections, each one devoted to a specific topic: Section 8.1 discusses contributions that have been made to the field of QTCs, section 8.2 looks at various mathematical tools that can be used to describe the effects of the decoherence phenomenon, section 8.3 goes over the idea of time-varying quantum channels, and finally, section 8.4 looks at the theoretical limits of error correction in the context of time-varying quantum channels.

The research that appears in this chapter has been published in the following journal and conference papers:

- J. Etxezarreta Martinez, P. Fuentes, P. M. Crespo, and J. Garcia-Frias, "Pauli Channel Online Estimation Protocol for Quantum Turbo Codes," *IEEE International Conference on Quantum Computing and Engineering (QCE20)*, 2020. doi: 10.1109/QCE49297.2020.00023.

- J. Etxezarreta Martinez, P. Fuentes, P. M. Crespo, and J. Garcia-Frias, "Approximating Decoherence Processes for the Design and Simulation of Quantum Error Correction Codes in Classical Computers," *IEEE Access*, vol. 8, pp. 172623-172643, 2020. doi: 10.1109/ACCESS.2020.3025619.

- J. Etxezarreta Martinez, P. Fuentes, P. M. Crespo, and J. Garcia-Frias, "Time-varying quantum channel models for superconducting qubits," *npj Quantum Information*, vol. 7, no. 115, 2021. doi: 10.1038/s41534-021-00448-5.

- J. Etxezarreta Martinez, P. Fuentes, P. M. Crespo, and J. Garcia-Frias, "Quantum outage probability for time-varying quantum channels," submitted to *Phys. Rev. A*, 2021. arXiv:2108.13701.

1.2.8 CHAPTER 9: CONCLUSION AND FUTURE WORK

This final chapter concludes our discourse by summarizing the conclusions of our work and analyzing possible routes that the research conducted in this thesis may follow in the future.

1.3 HOW TO READ THIS THESIS

An outline of the contents of this thesis is shown in Figure 1.2. The different parts of this dissertation (enclosed by dotted rectangles in Figure 1.2) need not be read sequentially, as they are mostly independent from each other. It is worth noting, however, that the notation and particular contents of the chapters of this thesis differ from the original journal articles they are based on. This has been done for the sake of clarity and to maintain the integrity of the notation employed herein. For this reason, it is the author's belief that readers will benefit most from reading each chapter as it is presented (reading the dissertation from start to finish). Thus, we believe that going through this thesis by following the orange path shown in Figure 1.2 will result in the best possible reading experience.

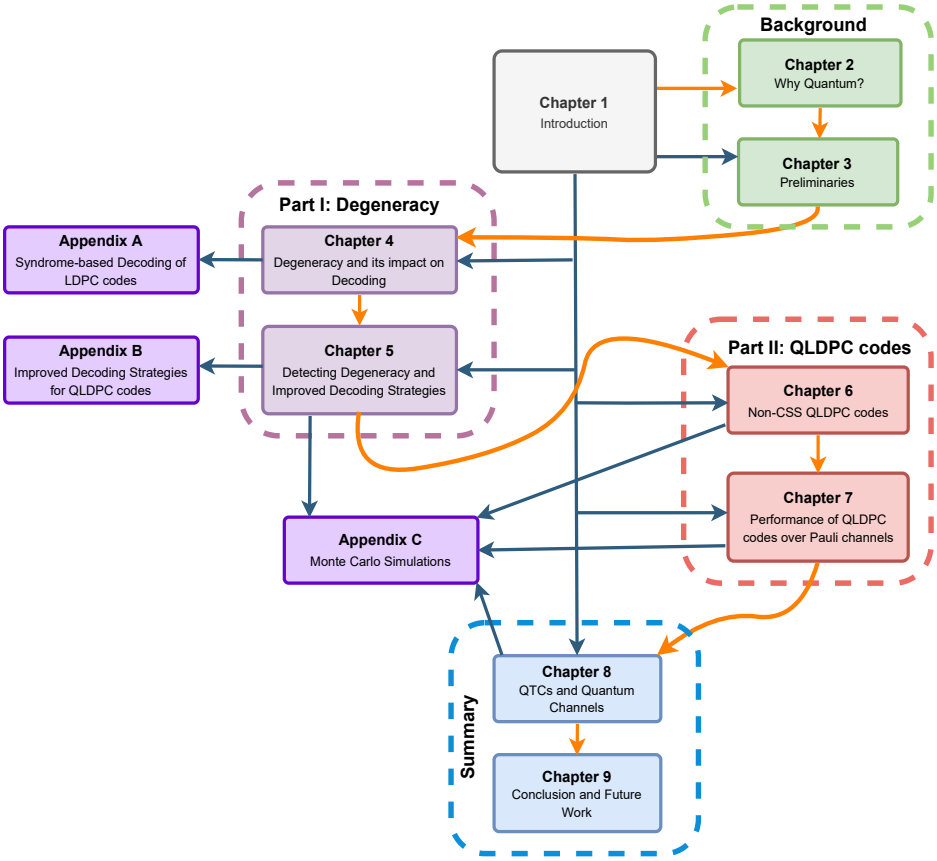


Figure 1.2: *Block diagram detailing the dependencies between chapters.*

CHAPTER 2

Why Quantum?

"Insanity is doing the same thing over and over and expecting different results"
Albert Einstein.

The advent of computers in the late twentieth century has profoundly transformed society. Originally designed to facilitate the computation of mathematical calculations, these machines have transcended their primary purpose and are now present in almost all aspects of modern life. Computers are essentially omnipresent, running everything from national monetary frameworks to power grids, serving as the gateway to the vast digital world we refer to as the internet, and being the most valuable asset in the life of many individuals and companies. However, even though computers have enabled mankind to address issues that could not even be conceived prior to their invention, problems that cannot be solved using the worlds most powerful machines still remain. This inability of classical computers to solve specific problems has both negative and positive implications. For instance, in the realm of drug design where complex optimization problems are commonplace, the incapacity of classical computers to solve these problems slows down the process of drug discovery and is perceived as something negative. In contrast, in the field of cryptography, the fact that

certain problems can not be solved with classical machines is what makes electronic devices and protocols secure.

One might assume that based on technological advancement and the constant improvement of electronic components, those complex problems that cannot be currently solved using classical means may potentially be solvable with future classical methods. In other words, what is impossible in today's computers may not be so in those of the future. However, we know this not to be the case thanks to the strong Church-Turing thesis [77]. The thesis tells us that every physical implementation of universal computation can simulate any other implementation with only a polynomial slowdown. Essentially, this means that while future classical computers may be better than current ones at attempting to solve these problems, the difficulty of solving the problems scales with the size of the input in the same way on both hardwares, i.e, the complexity of solving the task can be understood as being independent of the computer it is run on. Therefore, we can say that there is a subset of problems that, no matter how advanced electronic computational methods become, will be impossible to solve by classical computers in a reasonable amount of time. To better understand this concept, we need to look at it through the lens of classical complexity theory, which is the scientific discipline tasked with classifying computational problems according to their difficulty.

In complexity theory, so-called easy, or classically tractable, problems can be solved by computer algorithms that run in polynomial time; i.e., for a problem of size N, the time or number of steps needed to find the solution is a polynomial function of N. Algorithms for solving hard, or intractable, problems, on the other hand, require times that are exponential functions of the problem size N. Polynomial-time algorithms are considered to be efficient, while exponential-time algorithms are considered inefficient, because the execution times of the latter grow much more rapidly as the problem size increases. Based on this perspective, complexity theorists refer to the aforementioned clasically-intractable problems as Non-deterministic Polynomial (NP) time problems, a term that represents a class of computational problems for which no efficient solution algorithm has been found. Problems are said to belong to the NP class if their solution can be guessed and verified in polynomial time, and are labelled as non-deterministic because no particular rule is followed to make the guess. Thus, although a solution to an NP problem can be verified "quickly" (in polynomial time), there is no known way to find a solution rapidly. That is, the time required to

solve the problem using any currently known algorithm increases exponentially as the size of the problem grows. It is for this reason that the search for a polynomial time algorithm capable of solving NP problems, called the *P versus NP problem*, is one of the fundamental unsolved problems in computer science today.

Numerous well-known problems in computer science and mathematics belong to the family of NP problems. Among them, the most popular are the Traveling salesman problem, the factorization of numbers into primes or the discrete logarithm problem. The first problem is common in optimization scenarios and consists in finding the minimum cyclic path connecting N points with specified distances between them. The latter two problems are prevalent in cryptography. For instance, the security of the Rivest Shamir Adleman (RSA) public key cryptography protocol [78], which is widely employed in traditional finance, relies on the fact that factoring numbers into their prime components is an NP problem. Similarly, many public-private key pair generation cryptographic schemes such as Elliptic Curve Cryptography (ECC) [79], prevalent in blockchain technology based protocols like Bitcoin [80], are secure due to fact that the discrete logarithm problem also belongs to the class of NP problems. Based on this discussion, it is clear that the development of a technology capable of solving classically-unapproachable NP problems will have a disruptive effect on many scientific fields. In fact, given that NP problems are actually quite frequent, it is likely that such a technology will become the catalyst for a societal upheaval similar to the one that occurred when classical computers first burst onto the scene.

It is for these reasons that quantum computing has garnered so much attention during the past decade, as these revolutionary computers are the scientific communities best bet to tackle some of the NP problems that have so thwarted all previous classical solution attempts. The concept of quantum computing was first proposed by Feynman [1], a realization that came to him after devoting time to studying a particular NP problem: the simulation of quantum systems. The difficulty of simulating quantum phenomena using classical methods is best explained by Gottesman in [16], but the main takeaway is that keeping track of a quantum state in a classical computer requires exponential classical resources. More specifically, while an N-bit classical computer has 2^N possible states, its state space is only N-dimensional, since a state can be described by a binary vector with N components. In contrast, an N-qubit quantum computer has a 2^N dimen-

sional state space, since a complex vector with 2^N components is necessary to describe any given state. The same reason that makes the simulation of quantum systems an NP problem on classical computers led Feynman to conjecture that computers that obeyed the laws of quantum mechanics would have the capacity to bypass classical computational limits. Although it may not have seemed so at the time, this statement is groundbreaking. It implies that the classification of problems into complexity classes does not apply to quantum computers, which, aside from foreshadowing that some classically intractable problems may actually become tractable on quantum machines, also suggests that the strong Church-Turing thesis is wrong. Since Feynman's proclamation, quantum algorithms capable of solving classical NP problems in polynomial time have been discovered, further cementing the promise of quantum computing. The best known examples are Shor's algorithm [4], which can factor numbers into their prime components in polynomial time, and Grover's algorithm [5, 6], which provides a quadratic speedup when searching for an entry in an unordered database made up of a finite number of objects[1].

In light of the astonishing promise and breathtaking potential of operational quantum computers, agents in both academia and the private sector are racing towards the development and construction of quantum computers sophisticated enough to run quantum algorithms. For a quantum information processing system to be useful, it requires long-lived quantum states and a viable way to interact with them. Although there are different ways of doing so, we generally consider these systems to be comprised of a number of two-level subsystems called qubits[2]. For quantum computers to be good, their constituent qubits must exhibit the following traits:

- **Long coherence times** → The quantum mechanical properties of qubits can only be leveraged whilst they remain coherent, i.e, while they are in a state of superposition. Qubits lose coherence when they interact with the outside world, an inevitable phenomenon baptized (unsurprisingly) as decoherence. Thus, the coherence time of a qubit is a measure of how long it stays in a workable superposition state, i.e, how much time passes before it "decoheres". Clearly, longer coherence times will allow for longer and more complex quantum computation.

[1] It's complexity is $O(\sqrt{N})$, whereas the best known classical algorithms scale as $O(N)$, where N denotes the number of objects in the database.

[2] The term comes from "quantum bit", since qubits are the quantum analogue of bits in a classical computer.

However, qubit coherence times can only be increased by minimizing the interaction of subatomic quantum particles with the outside world, an extremely complicated task that makes building quantum computers a remarkable feat of engineering.

- **High connectivity** → It is desirable for the qubits of a quantum computer to be highly connected, as this allows operations to act on specific qubits simultaneously. Because decoherence arises when qubits interact with their environment (which includes other qubits), achieving high connectivity between qubits while ensuring long coherence times is a complex task.

- **High fidelity gate operations** → Quantum computation is achieved through the execution of sequences of operations known as quantum gates. Physical implementations of these gates are not perfect and they are faulty by nature (this varies depending on the technology that is employed), which results in excess "noise" being added to quantum information when it is processed with quantum gates. Naturally, higher fidelity gate operations will lead to better quantum computing.

- **High scalability** → In broad terms, the power of a quantum computater is determined by the number of qubits that make it up. Thus, it is important for qubits to be scalable so that increasing numbers of them can be employed to construct quantum machines.

Based on the above list, it is easy to see that satisfying these requirements essentially comes down to the capacity of a quantum computer to handle decoherence-induced noise, as this phenomenon manifests with time, connectivity, and when performing operations. As mentioned previously in the introduction, QEC is (arguably) the best and only way to tackle this issue and minimize the negative impact of decoherence. Despite the theoretical nature of most of the current work on QEC, error correcting codes are indispensable for quantum computers to evolve beyond their presently reduced applications and achieve true and indisputable quantum supremacy [81]. Before delving into the realm of QEC in subsequent chapters, it is worthwhile to briefly discuss some of the most relevant technologies that are being explored as possible physical realizations of qubits. At the time of writing, the most advanced technologies for the construction of qubits are:

1. **Superconducting Circuits**: Qubits can be built based on superconductors by placing a resistance-free current in a superposition state, using a microwave signal, and making it oscillate around a circuit loop. The defining trait of superconducting qubits and the primary reason for them being so well-known is that they have to be kept at cryogenic temperatures (below 100mK, or 0.1 degrees above absolute zero), a necessary requirement for the resistance of the superconductor to vanish. This technology is advantageous for mainly two reasons: It has a faster quantum gate time than other technologies, allowing for much faster computation, and the technology behind superconducting qubits can take advantage of proven existing electronic circuit design methods and processes (such as printable circuits) to tackle the scalability issue of quantum computing. Unfortunately, superconducting qubit technology is not without its faults. Qubits built in this manner have short coherence times and low connectivity. Furthermore, this technology requires the achievement and maintenance of cryogenic temperatures, which can be expensive and cumbersome, as well as individual calibration (each superconducting qubit is slightly different).

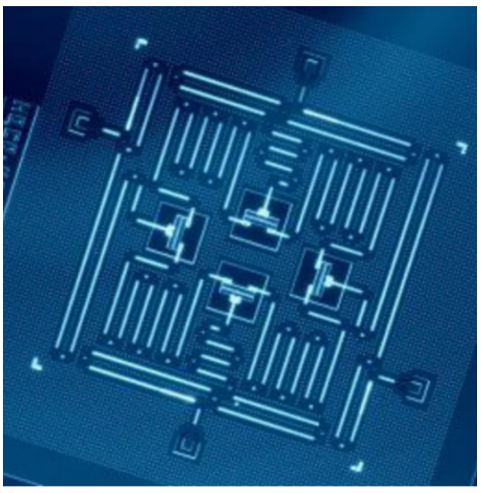


Figure 2.1: 4 *qubit superconducting processor fabricated at IBM [82].*

2. **Trapped Ions**: Ion Trap quantum computers work by trapping ions (charged atoms) using electric fields and holding them in place. Then, the outermost electron orbiting the nucleus can be put in different states and used as a qubit. The main appeal of trapped ion technology is its stability; trapped ion qubits have much longer coherence times than their superconducting counterparts. Additionally, although ions need to be cooled to perform optimally, the temperature requirement is much less prohibitive than for superconducting technology. Another important advantage is that ion trap qubits can be reconfigured, which allows for high qubit connectivity and avoids some of the issues and computational overhead found with other technologies. The downside to trapped ion quantum computing is its significantly slower operation time in comparison to other implementations. Other important drawbacks are the fact that ions need to be kept in high vacuum and that the technology involved in creating trapped ions, which requires the integration of techniques from a wide range of scientific domains, is not yet mature.

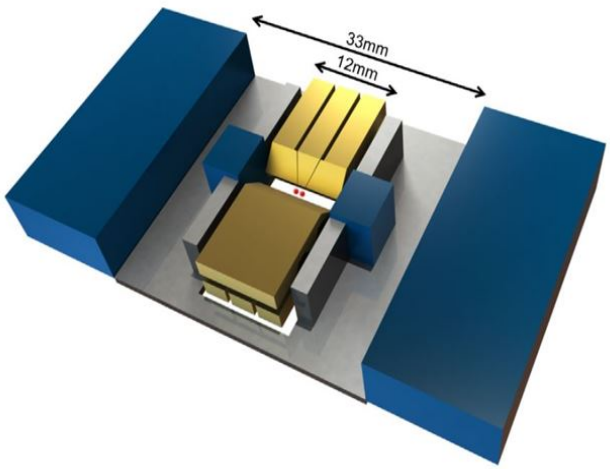


Figure 2.2: *Schematic of the linear Paul ion trap (yellow) fitted with four permanent magnets (blue), arranged to create a strong magnetic field gradient along the trap axis [84].*

3. **Photonics**: Photons (particles of light) operating on silicon chip pathways can be used to construct qubits. The primary advantage of this technology is that it does not require extreme cooling, allowing for more energy-efficient and less cumbersome quantum computing [85]. In a similar manner to superconducting qubits, because it is based on the use of silicon chips, this approach to quantum computing can exploit existing semiconductor industry infrastructure, which makes it highly scalable. Given that this technology is still nascent, important issues such as qubit connectivity remain to be proven.

4. **Neutral Atoms**: The neutral atom approach to quantum computing is similar to that of ion traps but instead of using charged particles, neutral atoms are used as qubits. Aside from exhibiting long coherence times, neutral atoms have the additional advantage of being configurable into arrays of single neutral atoms, which has the potential of becoming a very powerful and scalable technology to build and manipulate thousands of qubits [83]. Despite its promise, as is the case with many of these qubit implementation methods, the principal concerns regarding the use of this technology are that it is still in its first stages of development.

5. **Nitrogen-Vacancy Center**: One of the most recent qubit construction methods is based on the use of an electron spin inside a Nitrogen-Vacancy (NV) centre in a diamond lattice. NV centres are point defects in a diamond lattice characterized by having a nearest-neighbor pair of a nitrogen atom, which substitutes for a carbon atom, and a lattice vacancy (see Figure 2.3). Qubits built in this manner have long coherence times and can work at a large variety of temperatures. Once more, as of yet, there are few experimental results related to this approach to quantum computating.

This wide range of available technologies to physically implement quantum computers presents an unprecedented opportunity for entrepreneurship. In fact, the allure of the field is so strong it has attracted world-renowned companies and inspired the creation of many startups. Figure 2.4 classifies the biggest players in quantum computing according to the technology they have chosen to implement their quantum processors. Among them, D-Wave Systems, a company based in Canada, was the first to offer commercial access to quantum computers. Currently, IBM, which boasts a 127 qubit superconducting processor, QuEra, who claim to have built a

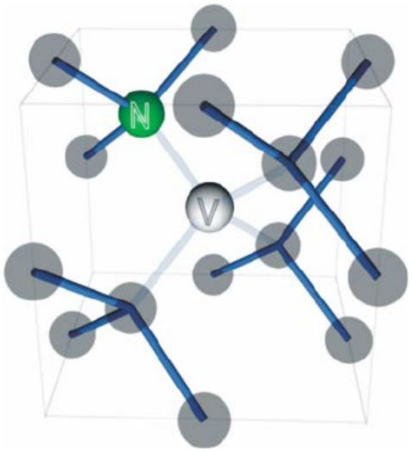


Figure 2.3: *Schematic representation of the nitrogen vacancy (NV) centre structure [86].*

256 neutral atom qubit computer, and IonQ, with its high fidelity 32 qubit trapped ion chip, appear to be leading the charge towards fully operational universal quantum computers.

An important remark

It is irrefutable that quantum computers have the potential to transform the world as we know it. However, it is important to remain grounded and to understand that they will likely "only" deliver tremendous speed-ups for particular types of problems. Fortunately, because some of these problems, like those related to optimization, are present in almost every aspect of society, quantum computing will possibly impact many areas of human life. Nonetheless, it should be stressed that quantum computers are not the *be-all and end-all* of science. We should also remember that, because the field of quantum computing is still in its infancy, we do not yet fully comprehend which problems are suited for quantum speed-ups and how to develop algorithms to demonstrate them.

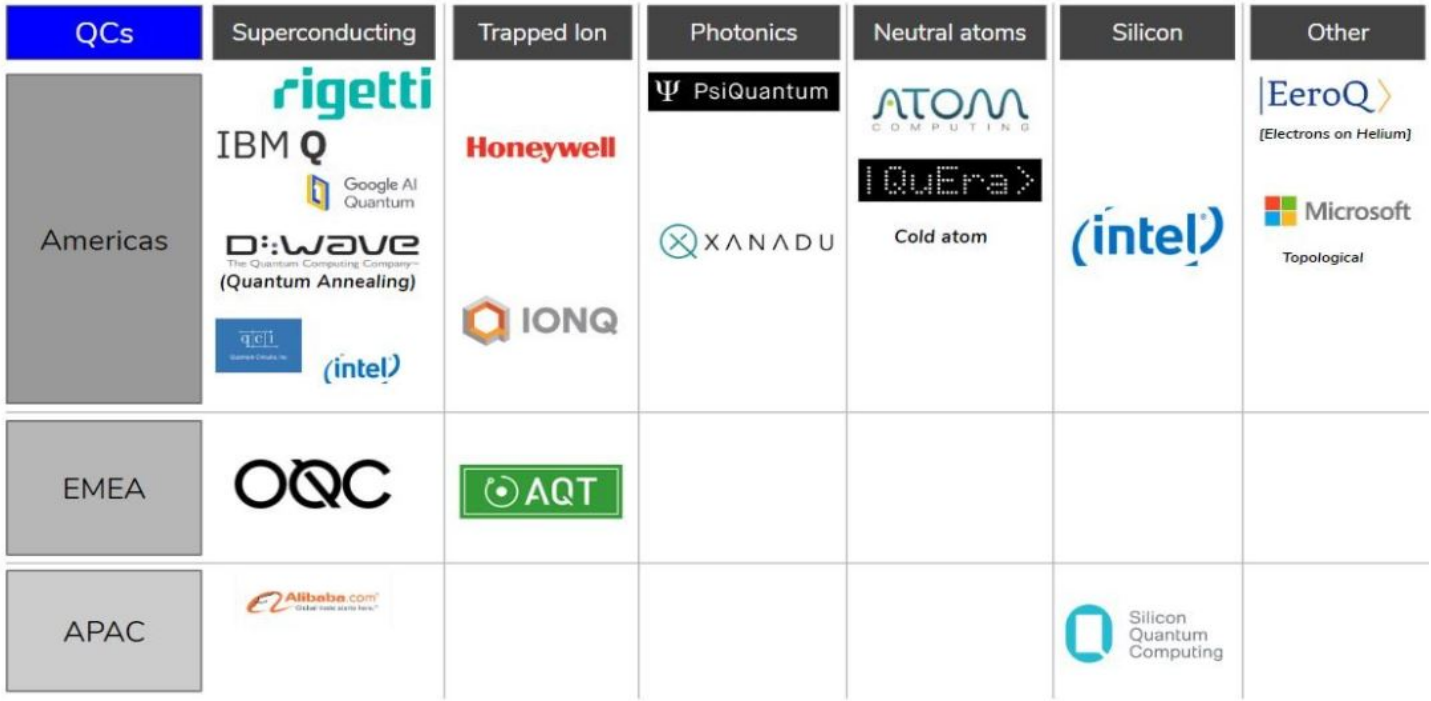


Figure 2.4: *Notable companies involved in quantum computing classified by their chosen technology and geographical location.*

CHAPTER 3

Preliminaries

"Life before death. Strength before weakness. Journey before destination"
Brandon Sanderson.

This chapter serves as a basic introduction to the realms of quantum information theory and classical error correction. It includes background material (terminology and notation) that aims to facilitate the reading and understanding of the rest of this dissertation. The chapter is divided into two major sections: section 3.1 dedicated to quantum information theory and section 3.2 devoted to classical error correction. Because the people involved in the field of quantum information come from a wide variety of scientific disciplines, some readers may find one (or both) sections familiar. In such a case, the appropriate sections should be skipped, although we do suggest reading section 3.1.4.3 as it introduces notation that differs slightly from the one employed in the literature.

3.1 QUANTUM INFORMATION

This section commences with an overview of the postulates of quantum mechanics and a discussion on the qubit and its various representations. Then, in subsections 3.1.3 and 3.1.4.1, we go over important aspects such

as entanglement and the no-cloning theorem. Following this, we introduce the concept of quantum noise and unitary operators in section 3.1.4. We conclude this introduction to quantum information by presenting the Pauli group in subsection 3.1.4.3 and providing a brief overview of the most common quantum channel models in section 3.1.4.5.

3.1.1 POSTULATES OF QUANTUM MECHANICS

The postulates of quantum mechanics provide us with the necessary tools to study the behaviour of subatomic particles. As stated in [75], they are the result of a long process of trial and (mostly) error, which involved a considerable amount of guessing and fumbling by the originators of the theory. Essentially, these postulates are the axioms on which the theory and mathematical framework of quantum mechanics is built. The motivation and reasoning behind them is not always clear (even to experts), but knowing what they represent can be helpful to understand other quantum mechanical concepts. For the sake of simplicity, in what follows we simply state the basic postulates of quantum mechanics. We will then explain and reference them as needed throughout this chapter. Because discussions regarding the origin and physical meaning of these postulates is beyond the scope of this dissertation, the reader is referred to [75, 88] for a rigorous discourse on this topic. The basic postulates of quantum mechanics are:

- **Postulate 1 - State Space**: Associated to any isolated physical system is a complex vector space with inner product (that is, a Hilbert space) known as the state space of the system. The system is completely described by its state vector, which is a unit vector in the system's state space.
- **Postulate 2 - Evolution**: The evolution of a closed quantum system is described by a unitary transformation. That is, the state $|\psi\rangle$ of the system at time t_1 is related to the state $|\psi'\rangle$ of the system at time t_2 by a unitary operator U which depends only on the times t_1 and t_2,

$$|\psi'\rangle = U\,|\psi\rangle\,.$$

- **Postulate 3 - Measurement**: Quantum measurements are described by a collection $\{M_m\}$ of measurement operators. These are operators acting on the state space of the system being measured. The index m refers to the measurement outcomes that may occur in

the experiment. If the state of the quantum system is $|\psi\rangle$ immediately before the measurement then the probability that result m occurs is given by

$$\mathrm{P}(m) = \langle\psi| M_m^\dagger M_m |\psi\rangle ,$$

and the state of the system after the measurement is

$$\frac{M_m |\psi\rangle}{\sqrt{\langle\psi| M_m^\dagger M_m |\psi\rangle}}.$$

The measurement operators satisfy the completeness equation,

$$\sum_m M_m^\dagger M_m = I,$$

where I is the identity matrix.

- **Postulate 4 - Composite systems**: The state space of a composite physical system is the tensor product of the state spaces of the component physical systems. Moreover, if we have systems numbered 1 through N, and system number i is prepared in the state $|\psi_i\rangle$ then the joint state of the total system is $|\psi_1\rangle \otimes |\psi_2\rangle \otimes \ldots \otimes |\psi_N\rangle$, where $\otimes$ denotes the tensor product.

3.1.2 THE QUBIT

The simplest quantum mechanical system and the basic unit in quantum information is known as the qubit. In contrast to classical bits, which can exist in only one of two possible states, 0 or 1, qubits exhibit a unique property, known as quantum superposition, that allows them to exist as a linear combination of these states. This means that, while a classical bit is an element of the binary field $\mathbb{F}_2$, a qubit is an element of the two dimensional complex Hilbert space $\mathcal{H}_2$. From the first postulate of quantum mechanics we know that a quantum mechanical system is described using the state vector formulation, also known as Braket or Dirac notation [75, 87, 88]. Thus, the superposition state of a qubit can be written as

$$|\psi\rangle = \alpha |0\rangle + \beta |1\rangle , \tag{3.1}$$

where $\alpha,\ \beta \in \mathbb{C}$ and $|\alpha|^2 + |\beta|^2 = 1$. The vectors $|0\rangle$ and $|1\rangle$ are orthonormal basis states that span $\mathcal{H}_2$ and are jointly referred to as the

computational basis states of a qubit. The complex numbers α and β are known as the qubit amplitudes.

Quantum measurement

The third postulate of quantum mechanics tells us that the probability that a quantum state $|\psi\rangle$ is in the state $|x\rangle$ is given by

$$\mathrm{P}(|x\rangle) = \langle\psi| M_x^\dagger M_x |\psi\rangle . \tag{3.2}$$

In quantum mechanics, the symbol $|\cdot\rangle$ represents a column vector known as a ket, and the symbol $\langle\cdot|$ represents a row vector known as a bra (hence why this is known as Braket notation). For every ket $|a\rangle$ there is a bra $\langle a|$ and they can be easily obtained from each other by computing the conjugate transpose, i.e, $|a\rangle = \langle a|^\dagger$. If we choose the measurement operator $M_x = |x\rangle \langle x|$ and introduce it in (3.2), we obtain

$$\begin{aligned}\mathrm{P}(|x\rangle) &= \langle\psi| M_x^\dagger M_x |\psi\rangle = \langle\psi| (|x\rangle \langle x|)^\dagger (|x\rangle \langle x|) |\psi\rangle \\ &= \langle\psi| (|x\rangle \langle x|)(|x\rangle \langle x|) |\psi\rangle = \langle\psi|x\rangle \langle x|x\rangle \langle x|\psi\rangle \\ &= \langle\psi|x\rangle \langle x|\psi\rangle = |\langle x|\psi\rangle|^2,\end{aligned} \tag{3.3}$$

where we have used $\langle x|x\rangle = 1$ and the notation $\langle\cdot|\cdot\rangle$ represents the inner product between the vectors $|\cdot\rangle$ and $\langle\cdot|$. If we now write the computational basis using vector notation as

$$|0\rangle = \begin{bmatrix}1\\0\end{bmatrix}, \ |1\rangle = \begin{bmatrix}0\\1\end{bmatrix},$$

we can use the measurement rule derived in (3.3) to see that the probability that the state $|\psi\rangle$ is in either of the base states is related to the amplitudes as:

$$\begin{aligned}\mathrm{P}(|0\rangle) &= |\langle 0|\psi\rangle|^2 = |\langle 0| (\alpha |0\rangle + \beta |1\rangle)|^2 \\ &= |\alpha \langle 0|0\rangle + \beta \langle 0|1\rangle|^2 = |\alpha|^2, \\ \mathrm{P}(|1\rangle) &= |\langle 1|\psi\rangle|^2 = |\langle 1| (\alpha |0\rangle + \beta |1\rangle)|^2 \\ &= |\alpha \langle 1|0\rangle + \beta \langle 1|1\rangle|^2 = |\beta|^2.\end{aligned} \tag{3.4}$$

Based on (3.4), we can understand why we previously stated that the amplitudes should satisfy $|\alpha|^2+|\beta|^2 = 1$. This comes from the completeness equation for the measurement operators given in the third postulate of quantum mechanics, which ensures that all probabilities add up to 1:

$$1 = \sum_m \mathrm{P}(m) = \sum_m \langle\psi| M_m^\dagger M_m |\psi\rangle .$$

This outcome, also known as the normalization condition, guarantees that we will always obtain a measurement outcome when measuring a qubit. Thus, we can re-define the qubit as a continuum of the states of the computational basis, that, upon measurement, will collapse to either one of the base states with a probability $|\alpha|^2$ or $|\beta|^2$, respectively.

Global phase

Another important consequence of the measurement postulate of quantum mechanics is the fact that the global phase of a quantum state has no observable consequence. Based on what we discussed previously, we can easily compute the probability of measuring a quantum state $|w\rangle = i\,|0\rangle$ in a specific state $|x\rangle$ as

$$\mathrm{P}(|x\rangle) = |\langle x|w\rangle|^2 = |\langle x|\,(i\,|0\rangle)|^2 = |i\,\langle x|0\rangle|^2 = |\langle x|0\rangle|^2,$$

where $i^2 = -1$. Notice how the probabilities for the state $i\,|0\rangle$ are identical to the probabilities for the state $|0\rangle$, i.e, $|\langle x|\,(i\,|0\rangle)|^2 = |\langle x|0\rangle|^2$. Because quantum measurement is the only possible way we have to extract information from a qubit, this means that the states $i\,|0\rangle$ and $|0\rangle$ are equivalent in all relevant physical ways. More generally, it can be said that quantum states that differ only by the overall factor $e^{-i\gamma}$ where γ is a real number, which we refer to as the global phase, are physically indistinguishable. This means that, for an arbitrary quantum state $e^{-i\gamma}\,|a\rangle$, the global phase has no observable consequence

$$|\langle x|\,(e^{-i\gamma}\,|a\rangle)|^2 = |e^{-i\gamma}\,\langle x|a\rangle|^2 = |\langle x|a\rangle|^2. \tag{3.5}$$

Pure states and Mixed states

An important distinction that can be made when studying qubit states is that of *pure* or *mixed* states. A pure qubit state is defined as a coherent

superposition of the basis states, meaning that it can be described as a linear combination of $|0\rangle$ and $|1\rangle$. Thus, pure qubit states are completely specified by a single ket and can be written as shown in (3.1). Mixed qubit states are defined as the statistical combination or incoherent mixture of different pure states that cannot be represented using the Dirac notation (mixed quantum states cannot be written as a single ket). Instead, they are represented in terms of the density matrix formulation of quantum mechanics, which is useful to describe qubits whose state is not completely known in state vector terms.

3.1.2.1 The Bloch Sphere

A practical way of visualizing the two-dimensional complex Hilbert space that defines a qubit is to represent it using a unit-radius sphere. In the jargon of quantum mechanics, this particular sphere is known as the Bloch sphere[1]. The top and bottom of the sphere on the Z-axis are generally chosen to correspond to the $|0\rangle$ and $|1\rangle$ base states, which in turn can represent the physical spin-up and spin-down states of an electron. In fact, some literature actually represents the computational basis using the notation $|\uparrow\rangle = |0\rangle$ and $|\downarrow\rangle = |1\rangle$.

The superposition state given in (3.1) can be rewritten using polar coordinates as

$$|\psi\rangle = r_\alpha e^{i\phi_\alpha} |0\rangle + r_\beta e^{i\phi_\beta} |1\rangle , \tag{3.6}$$

where the parameters $r_\alpha, \phi_\alpha, r_\beta$, and ϕ_β are real numbers. Knowing that the global phase $e^{i\gamma}$ has no observable consequence[2], we can multiply our state by $e^{-i\phi_\alpha}$, which yields

$$\begin{aligned} |\psi'\rangle &= e^{-i\phi_\alpha} |\psi\rangle = r_\alpha |0\rangle + r_\beta e^{-i\phi_\alpha} e^{i\phi_\beta} |1\rangle \\ &= r_\alpha |0\rangle + r_\beta e^{i(\phi_\beta - \phi_\alpha)} |1\rangle = r_\alpha |0\rangle + r_\beta e^{i\phi} |1\rangle , \end{aligned} \tag{3.7}$$

[1] It is so named as a tribute to the work of Swiss physicist Felix Bloch on the quantum theory of solids.

[2] In polar coordinates this can be shown as $|e^{i\gamma}\alpha|^2 = (e^{i\gamma}\alpha)^\dagger e^{i\gamma}\alpha = (e^{-i\gamma}\alpha^\dagger)(e^{i\gamma}\alpha) = \alpha^\dagger\alpha = |\alpha|^2$.

where $\phi = \phi_\beta - \phi_\alpha$. If we write the complex number $r_\beta e^{i\phi}$ in cartesian coordinates as $x + iy$ and considering that the normalization condition $\langle\psi'|\psi'\rangle = 1$ must hold, then

$$\begin{aligned}\langle\psi'|\psi'\rangle &= |r_\alpha|^2 + |x+iy|^2 \\ &= |r_\alpha|^2 + (x+iy)^*(x+iy) = r_\alpha^2 + x^2 + y^2 = 1.\end{aligned} \tag{3.8}$$

Notice that the expression shown in (3.8) is the equation of a unit radius sphere with cartesian coordinates r_α, x, and y. By introducing spherical coordinates we can write the state $|\psi\rangle$ (recall that because the global phase has no observable consequence the states $|\psi\rangle$ and $|\psi'\rangle$ are equivalent in all relevant physical manners) as

$$|\psi\rangle = \cos\theta' |0\rangle + e^{i\phi} \sin\theta' |1\rangle . \tag{3.9}$$

Now, in order for each point on the sphere to be identified by a unique set of spherical coordinates, we must restrict their range. For instance, note how for $\theta' = 0 \rightarrow |\psi\rangle = |0\rangle$ and for $\theta' = \pi \rightarrow |\psi\rangle = -|0\rangle$. This means that the north and south poles of the sphere are physically the same state, since we know the global phase factor $-1 = e^{-i\pi}$ to be irrelevant. Thus, we apply the restriction $0 \leq \theta' \leq \frac{\pi}{2}$ to uniquely identify all the points on the sphere. If we introduce $\theta = 2\theta'$, then we can write

$$|\psi\rangle = \cos\left(\frac{\theta}{2}\right) |0\rangle + e^{i\phi} \sin\left(\frac{\theta}{2}\right) |1\rangle , \tag{3.10}$$

where $0 \leq \theta \leq \pi, 0 \leq \phi \leq 2\pi$ are the coordinates of the points of the Bloch sphere. Using $\theta = 2\theta'$ is a useful convention, as it ensures that the basis state $|0\rangle$ corresponds to the north pole of the Bloch sphere and that the basis state $|1\rangle$ corresponds to the south pole of the sphere. A graphical representation of the Bloch sphere is shown in Figure 3.1. Aside from providing a visual tool to understand the qubit, the Bloch sphere also makes it easier to interpret the concept of pure quantum states and mixed quantum states: pure states can be understood as points on the surface of the sphere and mixed quantum states can be defined as points within the sphere.

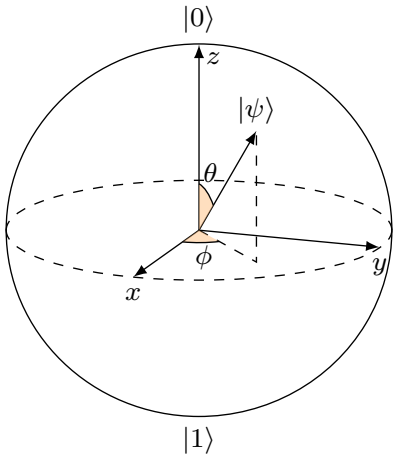


Figure 3.1: *Graphical representation of a qubit $|\psi\rangle$ on the Bloch sphere.*

3.1.3 ENTANGLEMENT

Entanglement is (arguably) the most notorious phenomenon in quantum mechanics. Described by Einstein as "*spooky action at a distance*" [89], it defines a property that links separated qubits and it is the source behind the power of quantum computers. A bipartite quantum system comprised of two single-qubit systems, $|\psi\rangle_{\mathrm{A}} = \alpha\,|0\rangle_{\mathrm{A}} + \beta\,|1\rangle_{\mathrm{A}}$ and $|\psi'\rangle_{\mathrm{B}} = \alpha'\,|0\rangle_{\mathrm{B}} + \beta'\,|1\rangle_{\mathrm{B}}$, can be formulated as

$$\begin{aligned} |\psi\rangle_{\mathrm{A}} \otimes |\psi'\rangle_{\mathrm{B}} &= (\alpha\,|0\rangle_{\mathrm{A}} + \beta\,|1\rangle_{\mathrm{A}}) \otimes (\alpha'\,|0\rangle_{\mathrm{B}} + \beta'\,|1\rangle_{\mathrm{B}}) \\ &= \alpha\alpha'\,|00\rangle_{\mathrm{AB}} + \alpha\beta'\,|01\rangle_{\mathrm{AB}} + \beta\alpha'\,|10\rangle_{\mathrm{AB}} + \beta\beta'\,|11\rangle_{\mathrm{AB}}\,, \end{aligned} \tag{3.11}$$

where the notation $|ij\rangle = |i\rangle \otimes |j\rangle$, i.e, it represents the tensor product between two qubits in states $|i\rangle$ and $|j\rangle$, respectively. Using vector notation, the base states of this two qubit system can be written as

$$|00\rangle = \begin{bmatrix}1\\0\\0\\0\end{bmatrix},\ |01\rangle = \begin{bmatrix}0\\1\\0\\0\end{bmatrix},\ |10\rangle = \begin{bmatrix}0\\0\\1\\0\end{bmatrix},\ |11\rangle = \begin{bmatrix}0\\0\\0\\1\end{bmatrix}.$$

Now consider a superposition state of just two of these basis states $|\sigma\rangle_{\mathrm{AB}} = \tau(|00\rangle_{\mathrm{AB}} + |11\rangle_{\mathrm{AB}})$. Because this state cannot be written as the tensor product of each of its constituent qubits, i.e, it cannot be obtained from a tensor product $|\psi\rangle_{\mathrm{A}} \otimes |\psi'\rangle_{\mathrm{B}}$, it is said to be entangled. Thus, entanglement is defined as the phenomenon by which composite quantum systems can be in states that cannot be written as a product of states of their constituent qubits.

The intricate connection that entangled qubits are bestowed makes it a useful property in myriads of quantum computing applications, the most prominent of which are Quantum Key Distribution (QKD) [90, 91, 92], quantum teleportation [93, 94] and superdense coding [95, 96]. Another common way of describing quantum entanglement is by means of the Bell states or Einstein-Podolsky-Rosen (EPR) pairs [75]. Bell states are useful because of the relative simplicity with which they can be generated (only two quantum gates are required to create an entangled Bell state), hence why most 2-qubit entanglement protocols rely on them. These states are generally denoted as

$$\begin{aligned}|\Phi^+\rangle &= \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle),\ |\Phi^-\rangle = \frac{1}{\sqrt{2}}(|00\rangle - |11\rangle)\\ |\Psi^+\rangle &= \frac{1}{\sqrt{2}}(|01\rangle + |10\rangle),\ |\Psi^-\rangle = \frac{1}{\sqrt{2}}(|01\rangle - |10\rangle).\end{aligned} \tag{3.12}$$

3.1.4 QUANTUM NOISE

Qubits can lose their coherence and become corrupted through a multitude of different mechanisms. Quantum states can be subjected to noise from a communications channel, they can deteriorate with the passage of time due to undesired interactions with their surroundings, and they can even be exposed to faulty error-inducing operations on a quantum computer. All these phenomena are grouped under a single term, decoherence,

which describes the process by which the coherent superposition of the basis states that compose the quantum information state of interest become perturbed [97]. This framework is useful because it allows us to model the coherence loss of qubits regardless of the technology used to implement them[3].

The effects of decoherence can be conveniently described using the mathematical abstraction of quantum channels, which map an input quantum state to a "noisy" output quantum state [98]. The goal of quantum error correction is to revert the action of these quantum channels and restore the output quantum state into the input quantum state. A pure input quantum state can result in a pure output state, which may be equal to the pure input state (in the extremely unlikely scenario that no channel corruption has occurred), or a mixed output state, if our input qubits have become entangled with the environment. Although it might seem that correcting a mixed output state will be harder than acting on a pure output, this is not the case. The mixed output state can be interpreted as an ensemble of pure states, which implies that if each of the constituent pure states can be corrected back to its original form, we will have recovered the full mixed state. From the perspective of the density matrix formulation of quantum mechanics, the actions of quantum channels can be understood as the application of a so-called *superoperator* on the input density matrix that describes our data qubits [97]. This superoperator can be diagonalized and written as the sum of a variety of different matrices that act on the input states with different probability. If a QEC code can correct any of the possible matrices, it will also be capable of correcting the full superoperator [16]. Although these individual matrices need not be unitary, the effects of decoherence on qubits are often modelled using combinations of specific unitary matrices.

3.1.4.1 Unitary operators

A linear operator whose hermitian conjugate (adjoint) is also its inverse is known as a unitary operator. More explicitly, a unitary operator U acts linearly on a quantum state as

$$U(|\psi\rangle) = U(\alpha\,|0\rangle + \beta\,|1\rangle) = U(\alpha\,|0\rangle) + U(\beta\,|1\rangle)\ ,$$

[3]Depending on the specific technology with which qubits are constructed, decoherence will arise due to different physical phenomena.

and it fulfils

$$U^{-1} = U^{\dagger} \ .$$

From the evolution postulate of quantum mechanics we know that unitary operators play an integral role in this theory. Explicitly, this postulate tells us that a closed quantum system in state $|\psi\rangle$ at time t_1 will be related to its state $|\psi'\rangle$ at time t_2 by a unitary operator U that depends strictly on the times t_1 and t_2 as

$$|\psi'\rangle = U\,|\psi\rangle\,. \tag{3.13}$$

This means that the evolution of a closed quantum system is described by a unitary transformation. A direct consequence of this is that the quantum analogues of single bit classical logic gates, single-qubit quantum gates, are described by unitary 2×2 matrices. These matrices must be unitary because of the normalization condition given in (3.8). Recall that any quantum state $|\psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle$ must fulfil $|\alpha|^2 + |\beta|^2 = 1$, which for quantum state $|\psi'\rangle = U\,|\psi\rangle$ holds only if the matrix U is unitary.

The No-cloning Theorem

A peculiar phenomenon in quantum mechanics, and one that also results from the second postulate of quantum mechanics, is the fact that quantum information cannot be copied or cloned. This concept will be alien to those with a classical information background, since the action of copying or replicating information is a tenet in many classical error correction and information storage strategies. Conventionally referred to as the No-Cloning theorem, the impossibility of copying quantum states arises as a result of the linearity of quantum mechanics. To show this, assume we have an operator C that copies the arbitrary quantum states $|\psi\rangle$ and $|\sigma\rangle$ as

$$C\,|\psi\rangle = \psi \otimes \psi,\ C\,|\sigma\rangle = \sigma \otimes \sigma \ .$$

Then if we wish to copy the sum of both states we perform

$$C(|\psi\rangle + |\sigma\rangle) = (|\psi\rangle + |\sigma\rangle) \otimes (|\psi\rangle + |\sigma\rangle) \ .$$

Because the transformation C must be linear (all unitary operators are bounded linear operators), then

$$C(|\psi\rangle + |\sigma\rangle) = C(|\psi\rangle) + C(|\sigma\rangle) = \psi \otimes \psi + \sigma \otimes \sigma.$$

However, it is obvious that $(|\psi\rangle+|\sigma\rangle)\otimes(|\psi\rangle+|\sigma\rangle) \neq \psi\otimes\psi+\sigma\otimes\sigma$, which means that our copying operator C has failed to copy the state $|\psi\rangle + |\sigma\rangle$. Thus, the primary and most relevant consequence of the no cloning theorem is that it is impossible to correctly copy superpositions of the basis states. This means that, in contrast to classical methods, quantum error correction strategies cannot rely on backup copies to preserve information. Instead, they must protect the original quantum state from errors for as long as required[4].

The Pauli Matrices

The second postulate of quantum mechanics hides a significant caveat: it does not tell us which unitary operators act on the quantum state, it simply tells us that the closed quantum system will evolve as shown in (3.13). In fact, it turns out that it is possible for any unitary transformation, i.e, any possible $U \in \mathbb{C}^{2\times2}$, to act on a qubit[5] [75]. Fortunately, the phenomenon of *error discretization*, which will be presented later on, tells us that a QEC code only needs to consider a basis of $\mathbb{C}^{2\times2}$ to be able to correct all the possible unitary transformations that can act on a qubit. For now, we present the most common single-qubit unitary operators in quantum mechanics.

The Pauli matrices are primordial elements in the fields of quantum computation and information. They are a set of 2×2 complex matrices that are both Hermitian[6] and unitary. Because any complex unitary 2×2 matrix is also a single-qubit quantum gate, the Pauli matrices can also be referred to as Pauli gates, i.e, the terms are interchangeable in this context. Although notation in the literature varies, they are commonly represented as follows:

[4]The longer that qubits are protected the longer their coherence time will be, which should allow the execution of more complex quantum algorithms.

[5]This also means that any unitary matrix can act as a quantum gate.

[6]A matrix A is Hermitian if $A^\dagger = A$.

- **Pauli Identity Matrix/Gate**: The Pauli identity matrix is defined as

$$I \equiv \sigma_0 \equiv \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} .$$

The action of the Pauli Identity gate on any quantum state $|\psi\rangle = \alpha |0\rangle + \beta |1\rangle$ is

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

- **Pauli X Matrix/Gate**: The Pauli X matrix is defined as

$$X \equiv \sigma_1 \equiv \sigma_x \equiv \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} .$$

The Pauli X gate, also known as the bit flip gate, acts by swapping the probabilities of the computational basis states. On the Bloch sphere, this can be seen as a 180° rotation about the x-axis (See Figure 3.2a). More explicitly, the action of the Pauli X gate on any quantum state $|\psi\rangle = \alpha |0\rangle + \beta |1\rangle$ can be written as

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

- **Pauli Y Matrix/Gate**: The Pauli Y matrix is defined as

$$Y \equiv \sigma_2 \equiv \sigma_y \equiv \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix} .$$

On the Bloch sphere, the Pauli Y gate acts by rotating a quantum state about the y-axis by 180° (See Figure 3.2b). This means that it swaps the amplitudes of the base states and introduces a phase shift of π between them. Mathematically, the action of the Pauli Y gate on any quantum state $|\psi\rangle = \alpha |0\rangle + \beta |1\rangle$ can be written as

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

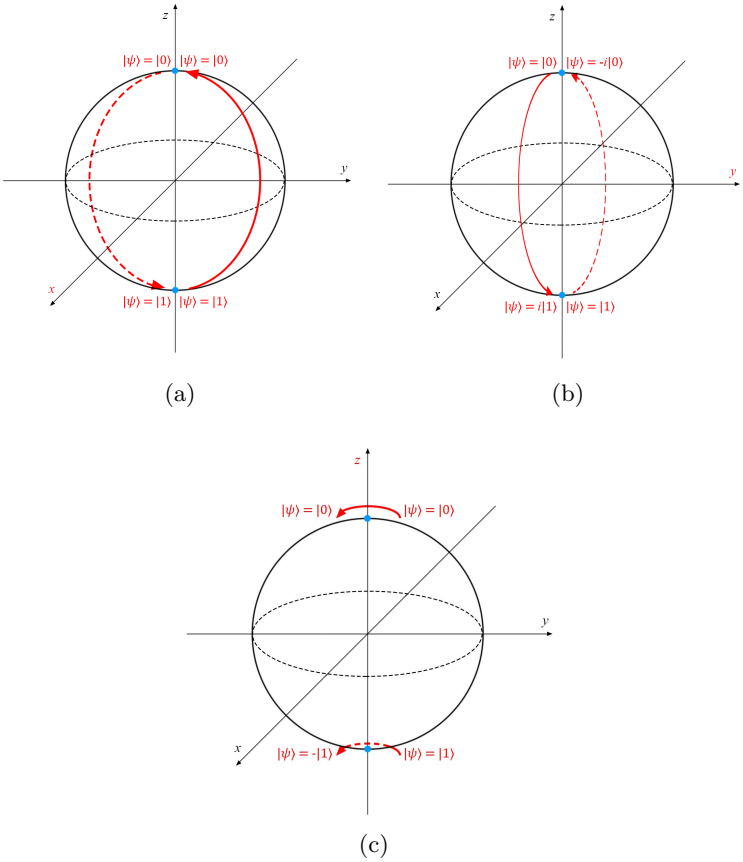


Figure 3.2: ***(a)*** *Effects of the Pauli X gate represented on the Bloch sphere.* ***(b)*** *Effects of the Pauli Y gate represented on the Bloch sphere.* ***(c)*** *Effects of the Pauli Z gate represented on the Bloch sphere.*

- **Pauli Z Matrix/Gate**: The Pauli Z matrix is defined as

$$Z \equiv \sigma_3 \equiv \sigma_z \equiv \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} .$$

The Pauli Z gate, also known as the phase-flip gate, rotates a quantum state about the z-axis by 180° (see Figure 3.2c). This can be understood as the gate leaving the basis state $|0\rangle$ unchanged and changing the sign of the $|1\rangle$ basis state, or as the introduction of a phase

shift of π (a phase flip) between the basis states. More explicitly, the action of the Pauli Z gate on any quantum state $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$ can be written as

$$|\psi'\rangle = Z|\psi\rangle = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \begin{pmatrix} \alpha \\ \beta \end{pmatrix} = \alpha|0\rangle - \beta|1\rangle \ .$$

Based on these definitions, it is clear that the three Pauli matrices are related. We can express this relationship mathematically as $Y = iXZ$. This means that the effects of the Pauli Y gate are those of applying a Pauli X gate followed by a Pauli Z gate, hence why the Pauli Y gate is sometimes referred to as the bit-and-phase flip gate. We can write the products between Pauli matrices in a more general manner as

$$\sigma_a \sigma_b = I\delta_{a,b} + i\sum_{c=1}^{3} \epsilon_{abc}\sigma_c, \tag{3.14}$$

where δ represents the Kronecker delta and

$$\epsilon_{abc} = \begin{cases} 1 & \text{if } (a,b,c) = \{(1,2,3),(3,1,2),(2,3,1)\}, \\ -1 & \text{if } (a,b,c) = \{(3,2,1),(1,3,2),(2,1,3)\}, \\ 0 & \text{if } a = b \text{ or } b = c \text{ or } a = c. \end{cases}$$

By carefully inspecting the product relationships shown in (3.14) we find that the Pauli matrices also exhibit the algebraic property of anticommuting with themselves. This can be written as,

$$\{\sigma_i, \sigma_j\} = \sigma_i\sigma_j + \sigma_j\sigma_i = 0 \ ,$$

where $\{\cdot\}$ denotes the anticommutator[7], $i \neq j$, and $i, j \in \{x, y, z\}$.

Hadamard Gate

The Hadamard gate is a single qubit gate defined as

$$H \equiv \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \ .$$

[7]The anticommutator of two operators $A \in \mathbb{C}^{N\times N}$ and $B \in \mathbb{C}^{N\times N}$ is defined as $\{A, B\} = AB + BA$, where if $\{A, B\} = 0$ the operators are said to anticommute.

It acts by mapping the pure basis states of the computational basis onto superpositions of these states as shown below

$$H|0\rangle = \frac{1}{\sqrt{2}}\begin{pmatrix}1 & 1\\ 1 & -1\end{pmatrix}\begin{pmatrix}1\\0\end{pmatrix} = \frac{1}{\sqrt{2}}\begin{pmatrix}1\\1\end{pmatrix}$$
$$= \frac{1}{\sqrt{2}}\left[\begin{pmatrix}1\\0\end{pmatrix} + \begin{pmatrix}0\\1\end{pmatrix}\right] = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) = |+\rangle \ ,$$

$$H|1\rangle = \frac{1}{\sqrt{2}}\begin{pmatrix}1 & 1\\ 1 & -1\end{pmatrix}\begin{pmatrix}0\\1\end{pmatrix} = \frac{1}{\sqrt{2}}\begin{pmatrix}1\\-1\end{pmatrix}$$
$$= \frac{1}{\sqrt{2}}\left[\begin{pmatrix}1\\0\end{pmatrix} + \begin{pmatrix}0\\-1\end{pmatrix}\right] = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle) = |-\rangle \ .$$

The states $|+\rangle$ and $|-\rangle$ are known as the Hadamard basis states. On the Bloch sphere, the operation of the Hadamard gate can be visualized as a 180° rotation around the x-axis followed by a 90° rotation around the y-axis. The effects of applying the Hadamard gate on the base state $|0\rangle$ are shown on the Bloch sphere in Figure 3.3a.

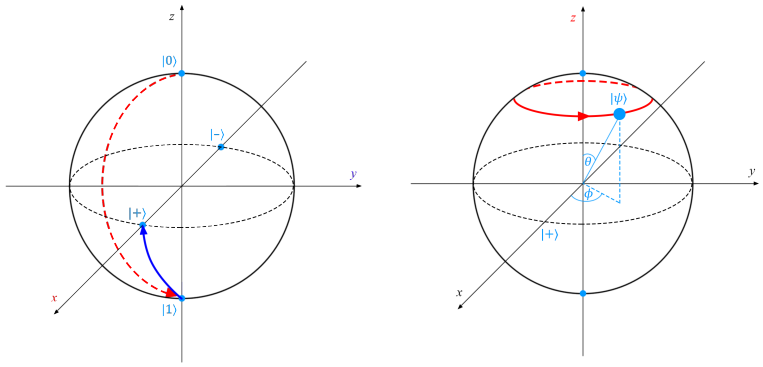
Phase shift gate

The Phase shift gate is a single qubit gate defined as

$$R_\phi \equiv \begin{pmatrix}1 & 0\\ 0 & e^{i\phi}\end{pmatrix} \ .$$

It acts by changing the phase of the basis state $|1\rangle$ while leaving the basis state $|0\rangle$ intact, i.e, it rotates the quantum state $|\psi\rangle$ around the z-axis by an amount determined by ϕ (see Figure 3.3b). For instance, if we set $\phi = \frac{\pi}{2}$, we will shift the quantum state 90° around the z-axis. The most common phase shift gates are the S and T gates, which are shown below and are obtained by setting $\phi = \frac{\pi}{2}$ and $\phi = \frac{\pi}{4}$, respectively. Note that the Pauli Z gate is a phase shift gate with $\phi = \pi$.

$$S \equiv \begin{pmatrix}1 & 0\\ 0 & i\end{pmatrix} \ , \ T \equiv \begin{pmatrix}1 & 0\\ 0 & e^{i\frac{\pi}{4}}\end{pmatrix} \ .$$



(a) (b)

Figure 3.3: ***(a)*** *Representation on the Bloch sphere of* $H|0\rangle = |+\rangle$*. Recall that the operation of the Hadamard gate can be visualized on the Bloch sphere as a* $180°$ *rotation around the* x*-axis (red arrow) followed by a* $90°$ *rotation around the* y*-axis (blue arrow).* ***(b)*** *Phase shift operation shown on the Bloch sphere.*

Multiqubit Gates

Multiqubit gates are quantum gates that act on two or more qubits (up to this point we have only seen single-qubit gates). Below we present two examples.

- **Swap gate**: The swap gate is defined as

$$\text{SWAP} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} .$$

  As befits its name, the swap gate re-orders the input qubits, i.e, $\text{SWAP}\,|0\rangle\,|1\rangle = |1\rangle\,|0\rangle$.

- **Controlled NOT (CNOT) gate**: The CNOT gate is a quantum analogue of the classical XOR gate. As with any controlled gate, the

CNOT gate has an input control qubit and an input target qubit. If the control qubit is in the base state $|1\rangle$, the gate will perform a NOT operation (a Pauli X) on the target qubit. Otherwise, the target qubit is left unchanged. The CNOT gate is defined by the matrix

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

More explicitly, the CNOT gate acts on two qubit quantum states as follows

$$\text{CNOT}\,|00\rangle \to |00\rangle\,,\ \text{CNOT}\,|01\rangle \to |01\rangle\,,$$
$$\text{CNOT}\,|10\rangle \to |11\rangle\,,\ \text{CNOT}\,|11\rangle \to |10\rangle\,,$$

where the first qubit is the control qubit and the second qubit is the target qubit.

3.1.4.2 Diagrammatic notation of quantum systems

Quantum information theory allows us to model a quantum computation as a sequence of quantum gates. We do this by drawing graphical depictions of quantum circuits based on the Penrose graphical notation [99], which was originally conceived to visually represent tensors in physics. These diagrams must be read from left-to-right and are composed of wires, drawn as straight lines, and quantum gates. These wires do not necessarily correspond to physical wires; instead, they may correspond to the passage of time, or to a physical particle such as a photon moving from one location to another through space. A single-qubit unitary operator U can be represented on a quantum circuit as a box placed over the corresponding wire (which represents the qubit the operator is acting on). Thus, we can represent any of the gates discussed in the previous section by simply substituting U by the appropriate symbol, i.e, X for the Pauli X gate or H for the Hadamard gate. This is shown in Figure 3.4a.

Similarly, we can represent multiqubit gates by joining the wires that model the evolution of the qubits in our quantum system. The quantum circuit representation of the SWAP gate and the CNOT gate is shown in

figures 3.4b and 3.4c. This diagrammatic notation of quantum systems will come in handy later on in the dissertation, especially when representing QEC circuits.

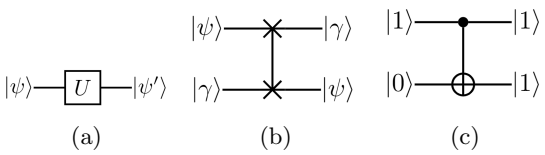


Figure 3.4: ***(a)*** *Schematic of the single-qubit unitary operator U acting on the input state $|\psi\rangle$.* ***(b)*** *Schematic of the SWAP gate in quantum circuits.* ***(c)*** *Schematic of the CNOT gate in quantum circuits.*

3.1.4.3 N-qubit generalization & The Pauli Group

So far in the dissertation we have only concerned ourselves with single qubits. Despite being useful when first presenting the framework of quantum information, quantum computers require a much larger number of qubits to run quantum algorithms. This requires that we generalize the framework that we have seen up to this point to quantum systems composed of N-qubits.

N-qubit systems

Previously, we employed the first postulate of quantum mechanics to establish that a qubit is an element of the two dimensional complex Hilbert space. Now, by applying the fourth postulate of quantum mechanics, we can define an N-qubit composite system as an element of the 2^N-dimensional Hilbert space, $\mathcal{H}_2^{\otimes N}$, which can be conveniently described as the tensor product of each of the individual qubits. Recall that we have actually already applied this concept to describe a 2-qubit system in section 3.1.3.

Based on this same principle, we can represent the state of an N-qubit system using a vector defined as the tensor product of N independent vectors, where each of these N vectors belongs to a different two-dimensional Hilbert space (each vector represents a single qubit). The basis states of

this 2^N-dimensional Hilbert space will be given by tensor products of the form $|v_1\rangle \otimes \ldots \otimes |v_N\rangle = |v_1 \ldots v_N\rangle$, where $v_i = \{0, 1\}$ and $i = 1, \ldots, N$. Therefore, we can generalize the superposition state of N-qubits as

$$\alpha_0 |00\ldots0\rangle + \alpha_2 |00\ldots1\rangle + \ldots + \alpha_{2^N-1} |11\ldots1\rangle \ ,$$

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

The Pauli Group

We will later see that the key to the functioning of most QEC strategies lies in the ingenious application of group theory. This requires that we introduce the concept of the Pauli group.

Let $\Pi = \{I, X, Y, Z\}$ denote the set of single-qubit Pauli matrices defined in section 3.1.4.1. Based on the product relationships between these matrices (see expression (3.14)) and including the phase factors $\{\pm 1, \pm i\}$, we define the set $\widetilde{\Pi}$ as

$$\begin{aligned} \widetilde{\Pi} &= \{\Theta_1 I, \Theta_2 X, \Theta_3 Y, \Theta_4 Z\} \\ &= \{\pm I, \pm iI, \pm X, \pm iX, \pm Y, \pm iY, \pm Z, \pm iZ\} \ , \end{aligned} \tag{3.15}$$

where $\Theta_l = \{\pm 1, \pm i\}$ and $l = 1, 2, 3, 4$. This set, together with the product defined in (3.14)), which we will denote by $\times$ to ensure notation clarity, compose a non-abelian[8] group known as the single-qubit Pauli group. We will write this group as $\mathcal{G}_1 = (\tilde{\Pi}, \times)$.

The single-qubit Pauli group can be extended to N-qubit systems by means of the tensor product. To that end, let $\Pi^{\otimes N} = \{I, X, Y, Z\}^{\otimes N}$ denote the set of N-fold tensor products of the single qubit Pauli operators and let $\widetilde{\Pi}^{\otimes N}$ be the set defined as $\widetilde{\Pi}^{\otimes N} = \cup_{k=0}^{3} i^k \Pi^{\otimes N}$. Note that $\widetilde{\Pi}^{\otimes N}$ is the set of N-fold tensor products of the single qubit Pauli operators and the possible overall factors $\{\pm 1, \pm i\}$. That is,

$$\widetilde{\Pi}^{\otimes N} = \{\Theta_1 I, \Theta_2 X, \Theta_3 Y, \Theta_4 Z\}^{\otimes N} \ , \tag{3.16}$$

[8] An abelian group, also called a commutative group, is a group in which the result of applying the group operation to two group elements does not depend on the order in which they are written. That is, the group operation is commutative.

where $\Theta_l = \{\pm 1, \pm i\}$ and $l = 1, 2, 3, 4$.

Notice that the cardinality of the sets $\Pi^{\otimes N}$ and $\widetilde{\Pi}^{\otimes N}$ is $|\Pi^{\otimes N}| = 2^{2N}$ and $|\widetilde{\Pi}^{\otimes N}| = 2^{2N+2}$, respectively. Let us now define the operation $\cdot$ as

$$\forall\ \mathrm{A}, \mathrm{B} \in \widetilde{\Pi}^{\otimes N},\ \ \mathrm{A} \cdot \mathrm{B} = A_1 \times B_1 \otimes B_2 \times A_2 \otimes \ldots \otimes A_N \times B_N, \quad (3.17)$$

where the products of single qubit operators are given by the product defined in (3.14). This operation together with the set $\widetilde{\Pi}^{\otimes N}$ compose what is known as the N-fold or N-qubit Pauli group, which we will denote by $\mathcal{G}_N = (\widetilde{\Pi}^{\otimes N}, \cdot)$. Note that any two operators in $\mathcal{G}_N$ either commute or anticommute, hence why this group is not abelian.

3.1.4.4 Error discretization

Previously we discussed how, as a consequence of the second postulate of quantum mechanics, it is possible for any operator to act on a qubit so long as it is unitary. This means that there is a continuum of errors that may affect a qubit, i.e, qubits may suffer an almost negligible error such as a phase shift of $\frac{\pi}{263}$ or they could be impacted by an apparently catastrophic error that may remove the qubit entirely [75]. Extraordinarily, it can be shown that the entire continuum of possible errors can be corrected by correcting only a discrete subset of those errors. This is known as error discretization and it is the central tenet of quantum error correction (should the entire continuum of possible errors have to be considered, it is obvious that error correction would not be feasible).

The discretization of errors in the quantum paradigm is a direct result of the *theorem for the necessary and sufficient conditions for error correction.* In homage to its discoverers, this theorem is often referred to as the Knill-Laflamme theorem [100], although the conditions for error correction where also formulated by Bennett at a similar time [101]. The theorem plays a pivotal role in QEC because it defines the conditions that an error correcting code must fulfil in order to protect against decoherence.

Theorem 1. *(Knill-Laflamme Theorem for the necessary and sufficient conditions of quantum error correction) Let* C *be a quantum error correcting code defined as a subspace of* $\mathcal{H}_2^{\otimes N}$ *and let* $\mathcal{E} \subset \mathbb{C}^{2^N \times 2^N}$ *denote a set of errors. Then* C *will be able to correct all the errors in* $\mathcal{E}$ *if and only if*

$$\langle\psi_i| E_a^\dagger E_b |\psi_j\rangle = c_{ab}\delta_{ij} \ ,$$

where $|\psi_i\rangle, |\psi_j\rangle$ *are a basis for the subspace defined by* C*, i.e, they run over all possible basis codewords,* $E_a, E_b \in \mathcal{E}$*,* $c_{ab} \in \mathbb{C}$ *is independent of* i *and* j*, and* δ *denotes the Kronecker delta.*

Let us discuss how the theorem actually provides the necessary and sufficient conditions for quantum error correction. In order for a code to correct two different errors, E_a and E_b for instance, it must be capable of distinguishing the action of E_a on a basis codeword $|\psi_i\rangle$ from the action of E_b on a different basis codeword $|\psi_j\rangle$. The only possible way that this can be done is if $E_a |\psi_i\rangle$ is orthogonal to $E_b |\psi_j\rangle$, which we can write as $\langle\psi_i| E_a^\dagger E_b |\psi_j\rangle = 0$ and $i \neq j$. However, if a code is to function as a quantum error correcting code, we must also ensure that when making measurements to diagnose an error we learn nothing about the encoded information. This is due to the third postulate of quantum mechanics, which tells us that we cannot directly measure quantum systems, else the superposition states will collapse. We learn about errors that our quantum information may have suffered by computing $\langle\psi_i| E_a^\dagger E_b |\psi_i\rangle$, and since we must not learn anything about the state of the code itself, then these measurements must be the same for all basis codewords: $\langle\psi_i| E_a^\dagger E_b |\psi_i\rangle = \langle\psi_j| E_a^\dagger E_b |\psi_j\rangle$. If we now reconsider the single equation given in the Knill-Laflamme theorem, we can see that it encompasses both of the conditions we have just explained.

The discretization of errors is perhaps the most critical result of the Knill-Laflamme Theorem, because it allows us to correct all possible errors by only considering a subset of them. It is generally formulated as a corollary of the theorem as follows.

Corollary 1. *(Discretization of errors) For any quantum error correcting code* C *defined as a subspace of* $\mathcal{H}_2^{\otimes N}$*, if* C *is capable of correcting all the errors in a set* $\mathcal{E} \subset \mathbb{C}^{2^N \times 2^N}$*, then* C *is capable of correcting all the errors in the linear span of* $\mathcal{E}$*.*

To truly understand the implications of this corolary, let us begin by defining a basis for $\mathbb{C}^{2\times 2}$. For this purpose we need to look no further than the set Π of the single qubit Pauli matrices themselves. Showing that the Pauli matrices are a basis for $\mathbb{C}^{2\times 2}$ is straightforward: Firstly,

the Pauli matrices cannot be written as linear combinations of themselves, which means that they are linearly independent. Secondly, because $\mathbb{C}^{2\times 2}$ is a complex vector space of dimension 4, any element of this space can be written as the linear combination of 4 linearly independent matrices that also belong to $\mathbb{C}^{2\times 2}$. Thus, the Pauli matrices are a basis for $\mathbb{C}^{2\times 2}$.

Having established this, we will now show that the set $\Pi^{\otimes N}$ of N-fold tensor products of single qubit Pauli matrices is a basis for $\mathbb{C}^{2^N\times 2^N}$. Thanks to the properties of the tensor product, we know that given a vector subspace V with basis $|i\rangle$ and a vector subspace W with basis $|j\rangle$, the tensor product of the basis' $|i\rangle \otimes |j\rangle$ is a basis for the tensor product space $V \otimes W$. Then, knowing that the set Π is a basis for $\mathbb{C}^{2\times 2}$ and that $\mathbb{C}^{2^N\times 2^N} = (\mathbb{C}^{2\times 2})^{\otimes N}$, the set $\Pi^{\otimes N}$ is a basis for $\mathbb{C}^{2^N\times 2^N}$.

Now, by focusing on the wording of the error discretization corollary: "if C is capable of correcting all the errors in a set $\mathcal{E} \subset \mathbb{C}^{2^N\times 2^N}$, then C is capable of correcting all the errors in the linear span of $\mathcal{E}$", it becomes clear why this result is so significant. If we can design codes with the capacity to correct errors in the set $\Pi^{\otimes N}$, because this set is a basis for $\mathbb{C}^{2^N\times 2^N}$ ($\Pi^{\otimes N}$ spans $\mathbb{C}^{2^N\times 2^N}$), our codes will also be capable of correcting any error in $\mathbb{C}^{2^N\times 2^N}$. Essentially, this means that for the purposes of error correction, instead of having to work with the entire space $\mathbb{C}^{2^N\times 2^N}$, it is enough to consider only the set $\Pi^{\otimes N}$.

3.1.4.5 Quantum Channels

Earlier we mentioned that decoherence can be understood as the undesired entanglement of a quantum state with its environment. This corrupts quantum superposition states and makes it impossible for quantum computers to function without reverting the effects of this corruption. Although they are caused by various different physical mechanisms, we know from the postulates of quantum mechanics that all decoherence-related processes must be unitary (if the quantum state and the environment are considered as a closed system [75, 88]) and that their effects must be diagnosed without direct measurement of the state of our data qubits (else their superposition would be disturbed). Additionally, decoherence is often assumed to affect each qubit differently, i.e, the noise process that affects each qubit is considered to be independent [102].

To truly comprehend the manner in which decoherence affects quantum information, it is useful to study single-qubit examples. First, let us consider how this phenomenon affects the basis states of the computational basis. The basis sates $|0\rangle$ and $|1\rangle$ can be said to decohere as

$$\begin{aligned} |\psi\rangle_E |0\rangle &\to |a_1\rangle_E |0\rangle + |a_2\rangle_E |1\rangle , \\ |\psi\rangle_E |1\rangle &\to |a_3\rangle_E |0\rangle + |a_4\rangle_E |1\rangle , \end{aligned} \tag{3.18}$$

where $|\psi\rangle_E$ is the state of the environment before any interaction takes place [97] and $\{|a_i\rangle_E\}_{i=1}^4$ are states of the environment. Essentially, the expressions given in (3.18) are a mathematical representation of the undesired entanglement of the basis states with the environment. Based on this description, we can say that a qubit in state $|\psi'\rangle = \alpha |0\rangle + \beta |1\rangle$ will decohere as

$$|\psi\rangle_E |\psi'\rangle \to \alpha(|a_1\rangle_E |0\rangle + |a_2\rangle_E |1\rangle) + \beta(|a_3\rangle_E |0\rangle + |a_4\rangle_E |1\rangle). \tag{3.19}$$

Despite being a useful example, the description given in (3.19) is an impractical model for more than one qubit [102]. Instead, we can use quantum channels, as they provide better and more practical ways of modelling decoherence. To be more precise, quantum channels are mathematical abstractions that describe the effects of decoherence by mapping input quantum states onto "noisy" output quantum states. Among them, the amplitude damping channel, which describes the energy loss suffered by a quantum system as a consequence of its interaction with the environment, and the phase damping channel, which characterizes the loss of quantum information without energy loss, are the most popular. By combining both of these models we obtain the combined phase-and-amplitude damping channel, which provides the most accurate possible model for decoherence. Unfortunately, simulation of these channel models requires an amount of resources that increases exponentially for every additional qubit, which makes it impossible to efficiently simulate quantum channels directly on classical machines for sufficiently large qubit counts. Thankfully, the following theorem provides a solution to this predicament.

Theorem 2. *(Gottesman-Knill Theorem) Consider a quantum computation performed using only the following elements: state preparations in the*

computational basis, Hadamard gates, phase gates, controlled-NOT gates, Pauli gates, and measurements in the computational basis. Such a computation can be efficiently simulated on a classical computer.

The Gottesman-Knill theorem [75, 103] tells us that certain quantum computations that involve complex and highly entangled states (such as keeping track of how multi-qubit quantum states decohere) may actually be tractable on classical computers. Although the theorem does not apply to all possible quantum computations, it resolves the exponential resource dilemma by providing a way to model decoherence without the need for a functioning quantum computer[9]. Additionally, the theorem also holds for all quantum circuits that can be described using the stabilizer formalism, which, as will be shown in the following chapter, encompasses many QEC code families and allows us to simulate QEC codes efficiently on classical computers [97].

It must be mentioned that the classically tractable model for decoherence that can be built thanks to the Gottesman-Knill theorem is actually an approximation of the aforementioned quantum channels (amplitude damping, phase damping, and combined amplitude and phase damping). Although not as precise a model as the combined amplitude and phase damping channel, by means of quantum information theory techniques such as "twirling", this decoherence model has been shown to be a valid approximation. For the sake of brevity, herein we will only discuss the Pauli channel model for decoherence, which is the name given to the decoherence model that classical machines can simulate. For a thorough and in-depth discussion on the topic of quantum channels and their approximations, the reader should refer to [97].

The Pauli channel

The Pauli channel is a classically tractable quantum channel model that represents the decoherence effects suffered by quantum information. The effect of the Pauli channel ξ_P upon an arbitrary single-qubit quantum state with density[10] matrix ρ can be written as

[9] It is worth nothing that the problem is somewhat paradoxical in nature: we actually need the very same machine we are trying to make work, a quantum computer, to completely track the decoherence of qubits.

[10] It is significantly easier to describe quantum channels using the density matrix representation of quantum mechanics (the state vector notation can become too complicated

$$\xi_P(\rho) = (1 - p_x - p_y - p_z)\rho + p_x X\rho X + p_y Y\rho Y + p_z Z\rho Z, \tag{3.20}$$

where $\{p_x, p_y, p_z\}$ are the probabilities that the state ρ suffers each respective operator. More explicitly, a qubit that traverses the Pauli channel experiences a bit-flip (X operator) with probability p_x, a phase-flip (Z operator) with probability p_z or a combination of both (a Y operator) with probability p_y. In summary, the action of the Pauli channel on a quantum state $|\psi\rangle$ can be understood as a mapping onto a linear combination of the original state ($I|\psi\rangle$), the phase flipped state ($Z|\psi\rangle$), the bit flipped state ($X|\psi\rangle$), and the bit-and-phase flipped state ($Y|\psi\rangle$), where the sum is weighted by the probabilities $\{p_x, p_y, p_z\}$. Note that the reason behind the Pauli channel being a manageable decoherence model for classical machines is that it can be completely described by means of the Pauli matrices (see expression (3.20)). In a similar manner to the generalization shown in section 3.1.4.3, the Pauli channel will act on a multi-qubit quantum state by applying an error operator $E \in \mathcal{G}_N$, where E can be understood as the tensor product of single-qubit Pauli matrices.

Based on the nature of the probabilities $\{p_x, p_y, p_z\}$ of each single-qubit Pauli operator, different versions of the general Pauli channel can be derived. The most popular quantum channel model in the QEC literature is the independent depolarizing channel model [21, 22, 23, 28, 30, 31, 55, 63, 64], which is a specific instance of the Pauli channel in which the individual operator probabilities are all equal, i.e., $p_x = p_z = p_y = \frac{p}{3}$. This channel model is completely characterized by the depolarizing probability p. When quantum states of N qubits are considered, the errors that take place belong to the N-fold Pauli group $\mathcal{G}_N$ and they will act independently on each qubit, causing an X, Z, or Y error with probability $p/3$ and leaving the qubit unchanged with probability $(1 - p)$. Later on in this dissertation we will present different channel models based on the properties of the individual probabilities $\{p_x, p_y, p_z\}$.

in some instances). The reader is referred to [75, 97] for a more complete discourse on this topic.

3.2 CLASSICAL ERROR CORRECTION

The field of classical error correction was first conceived following the publishing of Claude E. Shannon's seminal work [104]. Within it, Shannon proved that reliable transmission over a communication channel is not possible if the information transfer rate exceeds a quantity known as the channel capacity. Furthermore, this work also showed that in order to transmit information at rates close to the channel capacity, information protection strategies kown as channel codes are necessary. This gave birth to a new scientific field dedicated to the study and optimization of channel codes, which would later become known as the discipline of Classical Error Correction. In this dissertation we focus on a specific family of classical error correcting codes called linear block codes. Thus, this section begins with an introduction to this family of error correcting codes. Then, we introduce the linear block code family of Low Density Parity Check (LDPC) codes in section 3.2.2. This section also includes basic communication and graph theory notions like factor graphs an iterative decoding. For the sake of simplicity, in this introduction we deal only with binary codes and all the arithmetic will be mod 2. The reader is referred to [105, 106, 107, 108] for discussions on non-binary linear block codes.

3.2.1 LINEAR BLOCK CODES

Suppose we wish to transmit a message made up of k information bits. Then, a linear block code is defined as a linear mapping between a k-bit message vector $[\mathbf{u}]_{1\times k}$ and an N-bit codeword vector $[\mathbf{x}]_{1\times N}$. Linear block codes exhibit the linearity property because linear combinations of codewords also belong to the code (they are codewords). In other words, a binary linear block code constitutes a linear mapping from the space $\mathbb{F}_2^k$ to the space $\mathbb{F}_2^N$ which maps the information word $[\mathbf{u}]_{1\times k}$ to the codeword $[\mathbf{x}]_{1\times N}$ by computing the matrix product $[\mathbf{x}]_{1\times N} = [\mathbf{u}]_{1\times k}[\mathbf{G}]_{k\times N}$. The matrix $[\mathbf{G}]_{k\times N}$ is a size $k \times N$ matrix known as the generator matrix whose columns form a basis for the k-dimensional coding subspace of $\mathbb{F}_2^N$, and represent basis codewords. A linear block code can also be represented using its Parity Check Matrix (PCM) $\mathbf{H}$, which defines a basis for the nullspace of the code, i.e, the product $\mathbf{G}\mathbf{H}^\top$ will always be $\mathbf{0}$, where $\mathbf{0}$ denotes the all zero matrix of size $k \times (N-k)$. Thus, $[\mathbf{H}]_{(N-k)\times N}$ is a size $(N-k)\times k$ matrix whose rows represent the linear constraints to which the codewords of the code are subjected and that ensures $\mathbf{c}\mathbf{H}^\top = \mathbf{0}$ whenever

$[\mathbf{c}]_{1\times N}$ is a codeword. In other words, the PCM provides a simple way of checking if any length N binary vector belongs to the code. For context, the generator and parity check matrix pair of the [7,4,3] Hamming code [109] is shown below,

$$\mathbf{G}_{\text{ham}} = \begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 \end{pmatrix}, \ \mathbf{H}_{\text{ham}} = \begin{pmatrix} 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 1 & 0 & 0 & 1 \end{pmatrix}. \tag{3.21}$$

Linear block codes are conventionally represented using the $[N, k, d]$ notation, where N represents the block length of the code (the size of a codeword), k represents the number of encoded bits, and d represents a concept known as the minimum distance of the code. The rate of the code can be computed from these parameters as $R = \frac{k}{N}$. The distance of a code provides a measure of the error-correcting properties of a code. To better understand what it represents, it is useful to introduce the concept of the Hamming distance. The Hamming distance between two vectors is the minimum number of bits that must be flipped to convert one vector to the other. The distance between two binary vectors $\mathbf{a}$ and $\mathbf{b}$ is equal to the weight (the number of 1s in the vector) of $\mathbf{a}+\mathbf{b}$. This means that, for a code to correct t single-bit errors, it must have distance at least $2t + 1$ between any two codewords. A t bit error will take a codeword exactly a t distance away from its original value, so when the distance between codewords is at least $2t+1$, we can distinguish errors on different codewords and correct them to the proper codewords.

Let us now recall our initial example of wanting to transmit k information bits. Once the information message $\mathbf{u}$ has been encoded into the codeword $\mathbf{x}$ using the generator matrix, the next step is to transmit the codeword through a communication channel. Typically this results in noise being added to the codeword and a noisy vector $[\mathbf{r}]_{1\times N} = [\mathbf{c}]_{1\times N} \oplus [\mathbf{e}]_{1\times N}$ being received, where $[\mathbf{e}]_{1\times N}$ is a length N vector that represents the action of the particular channel in question. The next step would be to perform decoding, which refers to the process of attempting to revert the action of the channel on the transmitted codeword and recovering the original information message $\mathbf{u}$. There is a wide variety of different decoding strategies for the families of linear block codes, but the most basic and

easy to understand is known as syndrome detection. Essentially, when the noisy vector $\mathbf{r}$ is received, the decoder will compute a length $N-k$ binary vector $[\mathbf{z}]_{1\times(N-k)}$, known as the error syndrome, based on the product $[\mathbf{r}]_{1\times N}[\mathbf{H}^\top]_{N\times(N-k)} = [\mathbf{c}\oplus\mathbf{e}]_{1\times N}[\mathbf{H}^\top]_{N\times(N-k)} = [\mathbf{c}]_{1\times N}[\mathbf{H}^\top]_{N\times(N-k)} \oplus [\mathbf{e}]_{1\times N}[\mathbf{H}^\top]_{N\times(N-k)} = [\mathbf{e}]_{1\times N}[\mathbf{H}^\top]_{N\times(N-k)} = \mathbf{z}_{1\times(N-k)}$. If the syndrome is equal to zero, it is likely that the channel has not corrupted the codeword and that $\mathbf{x}$ has been received. If not, then the decoder can use the syndrome in combination with a decoding algorithm to find the noise vector $\mathbf{e}$ that was added to $\mathbf{c}$ and attempt to revert its impact by re-adding $\mathbf{e}$ to $\mathbf{r}$.

3.2.2 LOW DENSITY PARITY CHECK CODES

Sparse codes, generally referred to as Low Density Parity Check codes, are a class of linear block codes whose parity check matrices, as befits their name, are low density[11] (sparse) matrices. LDPC codes were discovered by Gallager in 1961 [110, 111], but quickly fell from grace given that they could not be efficiently decoded. A couple of decades later, following the proposal of BP decoding [48] and a drastic improvement in communications technology, LDPC codes returned to the limelight as one of the best families of capacity achieving error correcting codes [112, 113, 114, 115].

The most common way of describing LDPC codes is by means of factor graph [47] and Tanner[12] graph [116, 117] representations of their PCMs. Factor graphs are generic, edge-connected graphs that are practical to represent mathematical expressions because of the straightforward manner in which they portray the dependencies and factorizations of these expressions. Moreover, LDPC codes can be efficiently decoded by running the SPA algorithm over a factor graph representation of their PCM. For these reasons, in what follows (which is an adaptation of [118]), factor graphs and the SPA algorithm are discussed in detail.

3.2.2.1 Factor Graphs

Algorithms that must deal with complicated multi-variable global functions often exploit the manner in which these functions factor into products of local functions, each of which depends on a subset of the variables. Such

[11] When a matrix is said to be sparse or low density, it will have a significantly larger number of zero entries than non-zero entries.

[12] Tanner graphs have been shown to be particular instances of factor graphs [50].

a factorization can be portrayed by means of a bipartite graph that is commonly known as a factor graph. For instance, let $g(x_1, x_2, x_3, x_4, x_5)$ be a multi-variable global function acting on the set of variables $\{X_1, X_2, X_3, X_4, X_5\}$. This function is represented in Fig. 3.5, where the *function* and *variable nodes* are represented as squares and circles, respectively. Function nodes and variable nodes are connected with an edge if and only if the corresponding variable is an argument of the function. Note that a factor graph is always bipartite, i.e., edges are only allowed between vertices of different types.

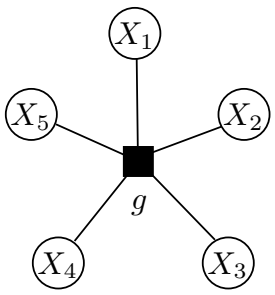


Figure 3.5: *Factor graph representation of the global multi-variable function* $g(x_1, x_2, x_3, x_4, x_5)$.

Factor graphs are most useful to represent the way in which global functions can be factorized as a product of several local functions. Let us assume that the function $g(x_1, x_2, x_3, x_4, x_5)$ can be written as

$$g(x_1, x_2, x_3, x_4, x_5) = f_A(x_1) \cdot f_B(x_2) \cdot f_C(x_1, x_2, x_3) \cdot f_D(x_3, x_4) \cdot f_E(x_3, x_5).$$

In this case, the factor graph can be expanded to represent the local functions f_A, f_B, f_C, f_D, f_E. This expansion is depicted in Fig. 3.6, where the variable nodes comprise the top layer of the graph and the function nodes make up the bottom layer of the graph. Although this bipartite graph is unique, it can be redrawn differently. For convenience, let us rearrange it as in Fig. 3.7.

Aside from the simplicity with which they represent multi-variable functions, the primary appeal of factor graphs comes from the fact that the SPA algorithm [47] can be run over them. It is for this reason that factor graphs

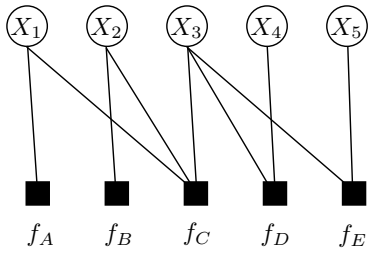


Figure 3.6: *Factorized factor graph of $g(\cdot)$*

are so often used in conjunction with LDPC codes, as the SPA algorithm can be used to decode LDPC codes by using the factor graph representation of their PCMs. In fact, many other decoding algorithms, such as BP or Viterbi decoding [48, 49], have been shown to be specific instances of the SPA [47].

The SPA is a message passing algorithm that exploits the way in which factor graphs represent the factorization of multi-variable global functions into simpler local functions to compute target functions of interest known as marginals. The computational efficiency of the algorithm comes precisely from the way in which factor graphs represent global functions. The algorithm can be summarized as a set of rules and procedures that govern the way in wich message passing takes place over a factor graph. Based on the specific structure of a factor graph, two different scenarios can be encountered:

The Sum-Product Algorithm over cycle-free Factor Graphs

Recall the rearranged factor graph shown in Fig. 3.7 corresponding to the global function $g(\cdot)$. This global function can be factorized as

$$g(x_1, x_2, x_3, x_4, x_5) = f_A(x_1) \cdot f_B(x_2) \cdot f_C(x_1, x_2, x_3) \cdot f_D(x_3, x_4) \cdot f_E(x_3, x_5).$$

Let us now assume that we are interested in calculating the marginal function $\hat{g}_1(x_1)$, given by

$$\hat{g}_1(x_1) = \sum_{x_2, x_3, x_4, x_5} g(x_1, x_2, x_3, x_4, x_5) = \sum_{\sim x_1} g(x_1, x_2, x_3, x_4, x_5). \quad (3.22)$$

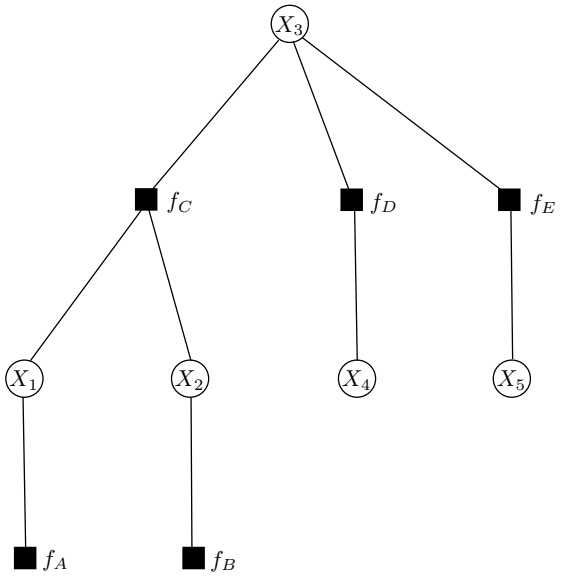


Figure 3.7: *Rearrangement of the graph shown in Figure 3.6.*

If certain criteria are met, then the SPA can be used to compute $\hat{g}_1(x_1)$ based on the exchange of messages between the nodes of the factor graph. The messages exchanged by the factor graph nodes are described as follows:

- We denote by $\psi(X_i)$ the set of function nodes f_j connected to variable node X_i. Similarly, $\psi(X_i) \setminus f_k$ denotes the set of function nodes f_j connected to the variable node X_i excluding the function node f_k. For example, $\psi(X_1) = \{f_A, f_C\}$ and $\psi(X_1) \setminus f_A = \{f_C\}$ in Fig. 3.7.
- We denote by $\mu_{X_i \to f_j}(x_i)$ and $\mu_{f_j \to X_i}(x_i)$ those messages transmitted from the variable node X_i to the function node f_j and from f_j to X_i, respectively. Note that these messages are only transmitted if $f_j \in \psi(X_i)$ and $X_i \in \psi(f_j)$.

These messages are propagated through all edges of the graph based on the operational rules of the algorithm and can be reused to compute different marginal functions. The operational rules define how each message should be computed and are summarized as:

Messages transmitted from variable nodes to function nodes: The message sent from variable node X_i to function node f_j is computed as

$$\mu_{X_i \to f_j}(x_i) = \prod_{f_k \in \psi(X_i) \setminus f_j} \mu_{f_k \to X_i}(x_i). \tag{3.23}$$

Messages transmitted from function nodes to variable nodes: The message sent from function node f_j to variable node X_i is computed as

$$\mu_{f_j \to X_i}(x_i) = \sum_{X_k \in \psi(f_j) \setminus X_i} f_j(x_k) \cdot \mu_{X_k \to f_j}(x_k). \tag{3.24}$$

Computation of the marginal functions: The marginal function $\hat{f}_i(x_i)$ is computed as

$$\hat{f}_i(x_i) = \prod_{f_j \in \psi(X_i)} \mu_{f_j \to X_i}(x_i). \tag{3.25}$$

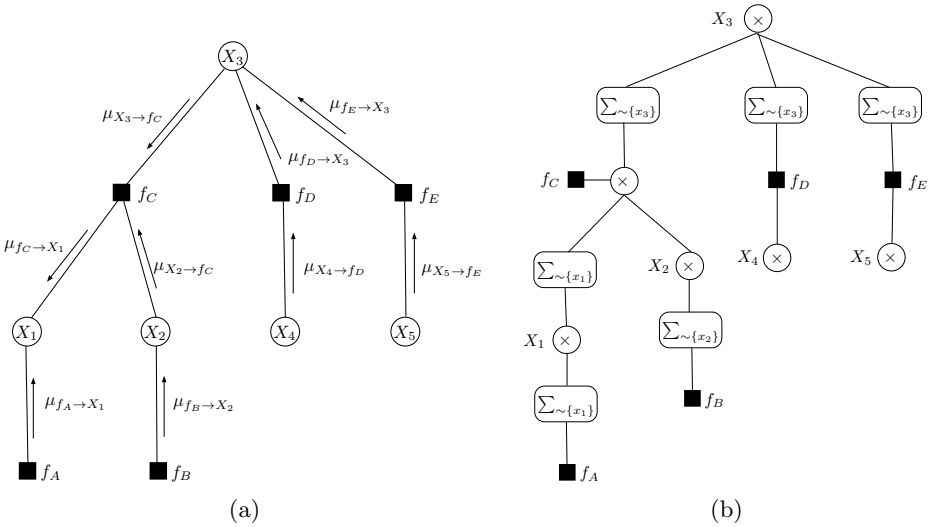


Figure 3.8: ***(a)*** *Messages involved in the marginalization of* $\hat{g}_1(x_1)$. ***(b)*** *Update rules of the marginalization of* $\hat{g}_1(x_1)$.

Figure 3.8 portrays the message exchange process of the SPA when computing the marginal function $\hat{g}_1(x_1)$. This can also be seen by breaking

up the expression given in (3.22) into its compounding factors as

$$\hat{g}_1(x_1) = \sum_{x_2}\sum_{x_3}\sum_{x_4}\sum_{x_5} f_A(x_1)\cdot f_B(x_2)\cdot f_C(x_1,x_2,x_3)\cdot f_D(x_3,x_4)\cdot f_E(x_3,x_5)$$

$$= \underbrace{f_A(x_1)}_{\mu_{f_A\to X_1}(x_1)} \cdot \underbrace{\sum_{x_2}\sum_{x_3} f_C(x_1,x_2,x_3)\cdot \underbrace{f_B(x_2)}_{\mu_{f_B\to X_2}(x_2)}}_{\mu_{X_2\to f_C}(x_2)} \cdot \underbrace{\underbrace{\sum_{x_4} f_D(x_3,x_4)\cdot \underbrace{1}_{\mu_{X_4\to f_D}(x_4)}}_{\mu_{f_D\to X_3}(x_3)} \cdot \underbrace{\sum_{x_5} f_E(x_3,x_5)\cdot \underbrace{1}_{\mu_{X_5\to f_E}(x_5)}}_{\mu_{f_E\to X_3}(x_3)}}_{\mu_{X_3\to f_C}(x_3)}}_{\mu_{f_C\to X_1}(x_1)}, \tag{3.26}$$

which allows us to derive that the marginalization of X_1 is actually given by,

$$\hat{g}_1(x_1) = \mu_{f_A\to X_1}(x_1)\cdot\mu_{f_C\to X_1}(x_1). \tag{3.27}$$

The other marginals $\hat{g}_2(x_2)$, $\hat{g}_3(x_3)$, $\hat{g}_4(x_4)$ and $\hat{g}_5(x_5)$ are computed in a similar fashion, each particular marginal requiring the computation of a different set of messages. All the necessary messages required to compute these marginals are shown in Fig. 3.9.

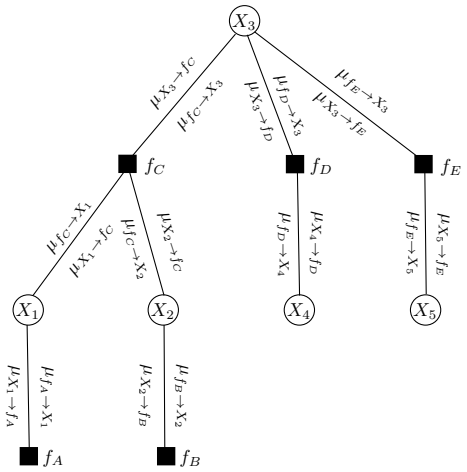


Figure 3.9: *All necessary messages required to compute the marginal functions $\hat{g}_1(x_1), \hat{g}_2(x_2)$, $\hat{g}_3(x_3)$, $\hat{g}_4(x_4)$ and $\hat{g}_5(x_5)$. Messages that propagate upwards/downwards are placed on the right/left side of the edges.*

At this point, it is important to note that execution of the SPA for the computation of these marginals requires a finite number of steps, i.e, the message exchange procedure halts on its own. This occurs because the factor graph representation of the global function $g(\cdot)$ (See Figures 3.5, 3.6, and 3.7) has no cycles. A cycle or loop can be understood as a closed path in the factor graph that begins and ends at the same variable node and that involves the passing of a single message over each of the edges that comprise the path. Cycles can be described by their length, which refers to the number of edges that conform the cycle itself. An example of a length-4 cycle is shown in Figure 3.10. Execution of the SPA over a factor graph with cycles produces an "iterative" algorithm with no natural termination in which messages are passed multiple times on a given edge. This results in the marginalizations computed by the algorithm not being exact function summaries. However, in many of the SPAs practical applications, like the decoding of LDPC codes or turbo codes, execution over graphs with cycles is involved. Fortunately, despite its inexact marginalization of global functions in the presence of cycles, extensive simulation results have shown that SPA-based decoding of very long codes can achieve astonishing performance (within a small fraction of a decibel of the Shannon capacity on a Gaussian channel) [112, 119, 120].

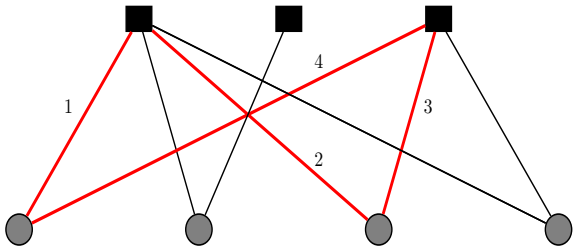


Figure 3.10: *Example of a cycle in a factor graph. The edges that conform the cycle are numbered from left to right.*

The Sum-Product algorithm over factor graphs with cycles

Most of the factor graphs that appear in the context of error correction contain cycles. This makes it impossible to compute an exact marginalization of the global function, as the algorithm can no longer terminate in a finite number of steps [121]. However, this does not mean the SPA cannot be executed over a factor graph with cycles, instead, it requires that

the iterative unending version of the SPA that manifests over loopy graphs be forcibly halted. This intervention or lack of a natural termination is the reason why the SPA is a sub-optimal marginalization algorithm in the presence of cycles.

When running over a loopy factor graph, operation of the SPA can be summarised in the following steps:

1. All factor graph messages are initialized.
2. Messages are updated according to a specific schedule: (3.23) and (3.24). This schedule may vary from step to step.
3. After each step (or a predefined number of steps) the marginal functions are computed: (3.25).
4. The algorithm output is derived from the current value of the marginal function.
5. Based on the output, a decision is made: if the output satisfies a set of conditions then the algorithm is stopped and if not, it continues to run through the previous steps until those conditions are met or until a specific number of iterations is reached.

Decoding of LDPC codes can be understood as a particular example of running the SPA algorithm over a factor graph with cycles.

Factor graph representation of a PCM

The factor graph representation of a linear block code is nothing more than a visual representation of the specific code constraints that the parity check matrix of the code defines. In fact, SPA-decoding of an LDPC code can be summarized succinctly as using the algorithm to check if a received message belongs to the code, which is appropriately represented by a factor graph. We know from the previous discussion that factor graphs are bipartite graphs that contain two distinct types of nodes; variable nodes and function nodes. Deriving such a graph for a linear block code is achieved by relating variable nodes to columns of the PCM and function nodes to the rows of the PCM; there will be a variable node for every column of the PCM and there will be a function node for every row of the PCM, respectively. The graph is completed by connecting variable nodes and function

nodes with a directionless edge[13] for every nonzero entry in the parity check matrix. As is done in the literature and in conventional error correction jargon, we will refer to the function nodes of the factor graph of linear block code as parity check nodes[14].

Figure 3.11 portrays the factor graph representation of a column-permuted version of the PCM of the [7,4,3] classical Hamming code shown in (3.28). Note that the property of linearity ensures that performing linear operations on the PCM of a linear block code does not alter the code itself, hence the PCM of (3.21) and the one shown below are essentially analogous.

$$\mathbf{H}_{\text{ham}} = \begin{pmatrix} 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{pmatrix} \tag{3.28}$$

The matrix shown in (3.28), given its reduced size, is not a low density matrix and so the [7,4,3] Hamming code is not actually an LDPC code. However, because LDPC codes are typically much larger and sparser, utilizing the Hamming code for this example serves the intended purpose. It must also be mentioned that, because the Hamming code is binary, the PCM-to-factor graph mapping is one-to-one [20, 122, 123]. This is not always the case for non-binary codes.

3.2.2.2 Syndrome-based decoding of LDPC Codes

Consider a classical linear block code with generator matrix $[\mathbf{G}]_{k\times N}$ and parity check matrix $[\mathbf{H}]_{(N-k)\times N}$ such that $\mathbf{G}\mathbf{H}^\top = \mathbf{H}\mathbf{G}^\top = \mathbf{0}$. Assume we encode the information word $[\mathbf{u}]_{1\times k}$ as $[\mathbf{u}]_{1\times k}[\mathbf{G}]_{k\times N} = [\mathbf{x}]_{1\times N}$ and transmit the codeword $[\mathbf{x}]_{1\times N}$ through a classical Binary Symmetric Channel (BSC) completely specified by the flip probability p (the channel flips each input bit with probability p). Upon reception of the noisy channel output sequence $[\mathbf{r}]_{1\times N} = [\mathbf{x}]_{1\times N} \oplus [\mathbf{e}]_{1\times N}$, the receiver computes the error syndrome, $[\mathbf{z}]_{1\times(N-k)}$, as $\mathbf{z} = \mathbf{x}\mathbf{H}^\top = (\mathbf{c}\oplus\mathbf{e})\mathbf{H}^\top = \mathbf{e}\mathbf{H}^\top$. The error syndrome can be understood as a Boolean representation of which code constraints are fulfilled by the received noisy sequence and which ones are

[13] Nodes of the same type are never connected.

[14] The rows of the PCM represent the equations that define the code. These equations enforce a parity on the code, hence why the function nodes of the factor graph corresponding to a linear code can be seen as performing parity checks.

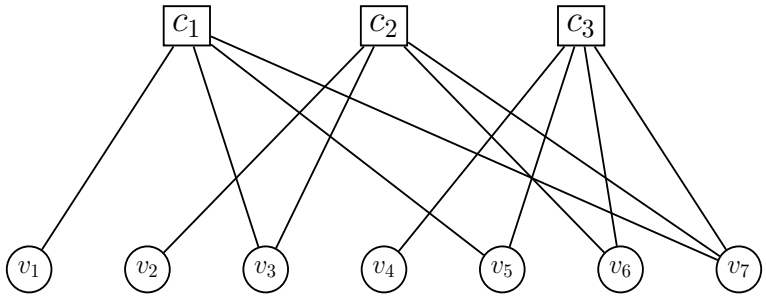


Figure 3.11: *Factor graph representation of the [7,4,3] Hamming code, which corresponds to the parity check matrix shown in equation (3.28).*

not[15]. Having determined the syndrome, the decoder will use it to produce an estimate $\hat{\mathbf{e}}$ of the error pattern induced by the channel. Following this, the inverse of the estimated error pattern, $-\hat{\mathbf{e}} = \hat{\mathbf{e}}$ (recall that arithmetic is mod2), is added to the received noisy vector $\mathbf{r}$. If the error estimate and the channel error match, $\hat{\mathbf{e}} = \mathbf{e}$, the codeword $\mathbf{r}$ will be recovered, and if not, a decoding mistake will have taken place.

Based on this entire process, it is clear that the purpose of the decoding process is to produce an estimate of the most likely error pattern $\hat{\mathbf{e}}$ given the syndrome $\mathbf{z}$ so that the probability of a word[16] error $\mathrm{P}(\hat{\mathbf{e}} \neq \mathbf{e})$, commonly known as the Word Error Rate (WER), is minimized. This is formally referred to as syndrome-based Maximum Likelihood (ML) decoding, which we can express mathematically as

$$\hat{\mathbf{e}}^{\mathrm{ML}} = \arg\max_{\mathbf{e}} \mathrm{P}(\mathbf{e}|\mathbf{z}). \tag{3.29}$$

By applying Bayes' rule, the expression shown in (3.29) can be expanded into

$$\hat{\mathbf{e}} = \arg\max_{\mathbf{e}} \mathrm{P}(\mathbf{e}|\mathbf{z}) = \arg\max_{\mathbf{e}} \frac{\mathrm{P}(\mathbf{z}|\mathbf{e})\mathrm{P}(\mathbf{e})}{\mathrm{P}(\mathbf{z})} \tag{3.30}$$

[15]Recall that the constraints of a classical code are represented by a set of parity check equations, each one associated to a different row of the parity check matrix of the code.

[16]Error patterns or sequences are sometimes referred to as words.

Notice that the computation of expressions (3.29) and (3.30) requires the brute force calculation of the probability of all the possible errors given the obtained syndrome $\mathbf{z}$. In the most simple of scenarios (codes of reduced block length), it is possible to implement the ML decoding rule using a pre-computed Look-Up Table (LUT) that associates error syndromes with error patterns [124]. However, since the set of possible errors grows exponentially as a function of the code block length, brute force and LUT computations quickly become intractable. In fact, decoding a generic code based on equation (3.29) has been shown to be an NP problem[17] [56, 125].

For this reason, decoding is generally approached via a less computationally demanding approximation of the ML decoding rule known as Symbol-Wise Maximum Likelihood (SWML) decoding. The SWML decoding rule can be written as

$$\begin{aligned} \hat{\mathrm{e}}_j^{\mathrm{SW}} &= \arg\max_{\mathrm{e}} P(\mathrm{e}_j = \mathrm{e}|\mathbf{z}) \\ &= \arg\max_{\mathrm{e}_j \in \{0,1\}} \sum_{\mathrm{e}_1,\ldots,\mathrm{e}_{j-1},\mathrm{e}_{j+1},\ldots,\mathrm{e}_N} P(\mathrm{e}_1,\ldots,\mathrm{e}_N|\mathbf{z}), \end{aligned} \tag{3.31}$$

where we have used regular lower case romans to denote the components of an error $\mathbf{e} \in \mathbb{F}_2^N$, i.e., $\mathbf{e} = [\mathrm{e}_1, \ldots, \mathrm{e}_N]$. The reason why SWML (3.31) is less computationally complex than ML (3.29) comes from the fact that SWML maximizes the individual marginal probabilities instead of the global maximum, i.e, it minimizes the symbol-wise error probability $\mathrm{P}(\hat{\mathrm{e}}_k \neq \mathrm{e}_k)$ instead of the WER. For general channels, this difference between both decoding rules can result in SWML yielding different results from those that would be obtained using conventional ML decoding, since the marginal optimum $\hat{\mathbf{e}}^{\mathrm{SW}} = [\hat{\mathrm{e}}_1^{\mathrm{SW}}, \ldots, \hat{\mathrm{e}}_N^{\mathrm{SW}}]$ need not be equal to the optimal sequence $\hat{\mathbf{e}}^{\mathrm{ML}}$. However, for memoryless channels (as are all the channels considered in this dissertation), both criteria agree and decoding based on either the ML or SWML rules will yield the same outcome [126].

The SWML decoding rule is generally implemented by running the SPA algorithm over the factor graph representation of the code. Although this procedure does not directly find a solution for (3.29) (this only occurs for memoryless channels and non-cyclic factor graphs), the SPA algorithm

[17] It is actually an NP-complete problem, which is the particular subclass that contains the hardest NP problems.

provides an efficient manner to perform SWML decoding of LDPC codes. Specifically, for the case of memoryless channels, and assuming that the graphical representation of the code does not contain cycles, the SPA obtains the solution for (3.29). Refer to chapter A in the Appendix for a detailed example on how SWML decoding of LDPC codes is performed by means of the SPA.

CHAPTER 4

Degeneracy and its impact on Decoding

"Not only is the universe stranger than we think, it is stranger than we can think"
Werner Heisenberg.

Oftentimes, there is enough similarity between the classical and quantum paradigms to warrant the application of classical methods to the quantum framework. This allows us to interpret, design, and implement quantum error correction strategies based on tried-and-true classical methods and enables us to make headway in our journey towards error-protected qubits. However, there is a particular mechanism by which quantum error correcting codes can exhibit properties that have never before been seen in classical error correction. In the literature, this phenomenon is known as *degeneracy*, and codes that display this particular trait are known as degenerate codes. Quantum codes are said to be degenerate when different error sequences have the same effect on their codewords. This stands in stark contrast to classical codes who will experience different effects (codewords will be corrupted differently) when subjected to different error sequences.

The manifestation of the degeneracy phenomenon has important implications. The fact that multiple errors can have the same effect on a degenerate code implies that, for this type of code, we will be capable of correcting these errors based on the same recovery operator. Essentially, this means that degeneracy allows quantum codes to "*pack more information*" than classical codes [75], which should ultimately have a positive impact on performance. However, because many QEC strategies make use of classical methods that cannot exploit degeneracy (they were designed for a paradigm in which this phenomenon is absent), this is not always the case. In fact, there are specific instances in which the use of classical error correction stratagems not only neglects the benefits of degeneracy, but also negatively impacts the performance of degenerate quantum codes. Aside from performance related issues, the existence of this strictly-quantum mechanism also disallows the use of classical proof techniques to derive the theoretical bounds on quantum error-correction. Thus, at present, degeneracy presents an interesting conundrum: it should technically improve the performance of quantum codes but it cannot be exploited using classical methods, and it also invalidates the use of classical methodologies to derive the theoretical error correction limits of degenerate quantum codes.

Beyond these observations and although significant progress has been made recently, especially with regard to improving the performance of quantum codes, the true impact of degeneracy on QEC codes is not yet completely understood. Part of this is due to the difficulty of accurately presenting the idea of degenerate codes, as is reflected by the different and varying notation that can be found in the literature. For this reason, in this chapter we attempt to derive an accurate and easy to follow interpretation of degenerate codes. To do so, we apply group theory to the framework of QEC codes and we use it to describe and study the degeneracy phenomenon in a simple and straightforward manner. Furthermore, this group theoretic approach enables us to discuss the increased nuance of the quantum decoding problem when compared to the classical problem and allows us to show how the presence of degeneracy can be both a blessing and a curse depending on the context.

The chapter begins by presenting the well-known stabilizer formalism in the context of our group theoretic approach. Prior to doing so, we introduce important QEC concepts such as the effective Pauli group or the symplectic representation. Alghough some readers may find this content to be better suited to the previous chapter, given that the interpretation and

notation that is employed herein differs from that used in the literature, we believe that including these notions at the beginning of this chapter eases the overall reading experience. Once stabilizer codes have been introduced, we present the quantum decoding problem and discuss its intricate nature. We close the chapter by providing a three-qubit example in which the effects of degeneracy and its impact on the decoding process can be illustrated in a more practical manner.

4.1 STABILIZER CODES

Stabilizer codes, also known as additive quantum codes, are an important class of quantum codes whose construction is analogous to classical linear codes. In fact, the stabilizer theory of quantum error correction allows us to import any binary or quaternary classical code for use as a quantum code, so long as it fulfils a specific condition. This is momentous, as it implies that, if classical codes that satisfy this particular constraint can be found, we may design quantum codes from pre-existing classical codes. Prior to introducing the theory of stabilizer codes, it is worthwhile to extend some of the concepts related to the group structure of Pauli operators. This discussion will come in handy later on, as we will see that stabilizer codes are intricately related to the Pauli group (see section 3.1.4.3).

4.1.1 THE EFFECTIVE PAULI GROUP

Earlier in section 3.1.4.3 we defined the N-qubit Pauli group as $\mathcal{G}_N = (\widetilde{\Pi}^{\otimes}, \cdot)$, where the set $\widetilde{\Pi}^{\otimes}$ is the set of N-fold tensor products of single qubit Pauli operators together with the overall factors $\{\pm 1, \pm i\}$ and $\cdot$ is the group operation defined in (3.17). We also showed in section 3.1.2 that quantum states that differ only by an overall phase factor are physically indistinguishable, i.e, that this phase factor has no observable consequence [127]. This means that it makes physical sense to neglect the phase factors $\{\pm 1, \pm i\}$ included in $\mathcal{G}_N$.

Thus, we can define the *effective N-fold Pauli Group*, $\overline{\mathcal{G}}_N = (\Pi^{\otimes N}, \star)$, where $\Pi^{\otimes N}$ is the set of N-fold tensor products of single qubit Pauli operators (without the overall factors) and $\star$ behaves as in (3.17) but with the operation between single Pauli operator products defined not as in (3.14)

but as

$$\sigma_a \star \sigma_b = I\delta_{a,b} + \sum_{c=1}^{3} |\epsilon_{abc}|\sigma_c, \tag{4.1}$$

where ϵ_{abc} is the same as in (3.14).

Observe that, in contrast to $\mathcal{G}_N$, $\overline{\mathcal{G}}_N$ is an abelian group. This means that the commutation relations exhibited by the elements of $\overline{\mathcal{G}}_N$ under the operation $\cdot$ (3.17) are lost under the $\star$ operation defined in (4.1).

The relationship between the N-qubit Pauli group and its effective counterpart goes beyond a difference in commutation relations. In what follows, we present a set of propositions that further characterize the effective Pauli group and establish an important isomorphism between subsets of the general Pauli group (where the operators in a subset differ only by a phase factor) and the elements of the effective Pauli group.

***Proposition* 1.** *: Let $\mathcal{P} \subset \mathcal{G}_N$ be the abelian subgroup $\mathcal{P} = (\{\pm I, \pm iI\}^{\otimes N}, \cdot)$. Then, for all $\mathrm{A} \in \mathcal{G}_N$, there is a unique operator $\mathbf{A} \in \mathcal{G}_N$ such that $\mathbf{A} \in \Pi^{\otimes N} \subset \widetilde{\Pi}^{\otimes N}$ and*

$$\mathbf{A} = \mathrm{P} \cdot \mathrm{A} = \mathrm{A} \cdot \mathrm{P}, \ \ \textit{for some } \mathrm{P} \in \mathcal{P}, \tag{4.2}$$

where for the sake of clarity, capital romans are used to denote operators that belong to $\mathcal{G}_N$, and capital boldface is used for those particular operators in $\mathcal{G}_N$ that belong to the subset $\Pi^{\otimes N}$. Note that $\mathrm{A} = \mathbf{A}$ iff $\mathrm{A} \in \Pi^{\otimes N} \subset \widetilde{\Pi}^{\otimes N}$.

Proof: We prove it by contradiction. Assume that for $\mathrm{A} \in \mathcal{G}_N$ there are two different operators $\mathbf{A}, \mathbf{B} \in \Pi^{\otimes N}$ in $\mathcal{G}_N$ such that

$$\mathbf{A} = \mathrm{P}_1 \cdot \mathrm{A} \text{ and } \mathbf{B} = \mathrm{P}_2 \cdot \mathrm{A} \text{ for some } \mathrm{P}_1 \text{ and } \mathrm{P}_2 \text{ in } \mathcal{P}.$$

Therefore,

$$\mathbf{A} \cdot \mathbf{B} = \mathrm{P}_1 \cdot \mathrm{P}_2 \cdot \mathrm{A}^2 = \pm \mathrm{P}_1 \cdot \mathrm{P}_2,$$

where we have applied that $\mathrm{A}^2 = \mathrm{A} \cdot \mathrm{A} = \mathrm{A} \cdot \mathrm{A}^\dagger = \pm\, I^{\otimes N}$ (The third step, where we have applied $\mathrm{A} = \mathrm{A}^\dagger$, holds because $\mathrm{A} \in \mathcal{G}_N$). Since $\pm \mathrm{P}_1 \cdot \mathrm{P}_2 \in \mathcal{P}$, then necessarily $\mathbf{A} = \mathbf{B}$.

□

Throughout the rest of this chapter and dissertation, an arbitrary operator $\mathrm{D} \in \mathcal{G}_N$ and its unique corresponding operator in $\mathcal{G}_N$ that belongs

to $\Pi^{\otimes N}$ will be denoted by the same letter, i.e., D and $\mathbf{D}$. Furthermore, we employ the symbolic notation $\mathrm{D} \equiv \mathbf{D}$ to represent the fact that for all physical purposes or from a quantum operator perspective, D and $\mathbf{D}$ will be equal up to a phase. That is,

$$\mathrm{D} \equiv \mathbf{D} \;\Leftrightarrow\; \mathrm{D} = i^k \mathbf{D}, \;\; \text{for some } k \in \{0,1,2,3\}. \tag{4.3}$$

From the definition of the effective N-fold Pauli group $\overline{\mathcal{G}}_N = (\Pi^{\otimes N}, \star)$, the following proposition easily follows.

***Proposition* 2.** *: Given any* $\mathrm{A}, \mathrm{B} \in \mathcal{G}_N$ *and* $\mathbf{A}, \mathbf{B} \in \overline{\mathcal{G}}_N$*, such that* $\mathrm{A} \equiv \mathbf{A}$ *and* $\mathrm{B} \equiv \mathbf{B}$*, then,* $\mathrm{A} \cdot \mathrm{B} \equiv \mathbf{A} \star \mathbf{B}$*.*

Proof: From (4.3), $\mathrm{A} \cdot \mathrm{B} = (i^{k_1}\mathbf{A}) \cdot (i^{k_2}\mathbf{B}) = i^{k_1+k_2}(\mathbf{A} \cdot \mathbf{B})$ for some $k_1, k_2 \in \{0,1,2,3\}$. Based on the group operations, $\cdot$ (3.17) and $\star$ (4.1), we know that $\mathbf{A} \cdot \mathbf{B}$ can only differ from $\mathbf{A} \star \mathbf{B}$ by a phase factor, i.e., $\mathbf{A} \cdot \mathbf{B} = i^{k_3}\mathbf{A} \star \mathbf{B}$, for some $k_3 \in \{0,1,2,3\}$. Then, $\mathrm{A} \cdot \mathrm{B} = i^{k_1+k_2}(\mathbf{A} \cdot \mathbf{B}) = i^{k_1+k_2+k_3}\mathbf{A} \star \mathbf{B} = i^k \mathbf{A} \star \mathbf{B}$, where $k = (k_1 + k_2 + k_3) \mathrm{mod} 4$. Therefore, $\mathrm{A} \cdot \mathrm{B} \equiv \mathbf{A} \star \mathbf{B}$.

□

Let us now define the equivalence relation $\sim_{\mathcal{P}}$ on $\mathcal{G}_N$. For all $\mathrm{A}, \mathrm{B} \in \mathcal{G}_N$, A will be equivalent to B, i.e.,

$$\forall\, \mathrm{A}, \mathrm{B} \in \mathcal{G}_N, \;\; \mathrm{A} \sim_{\mathcal{P}} \mathrm{B}$$

if and only if $\mathrm{B} = \mathrm{P} \cdot \mathrm{A} = \mathrm{A} \cdot \mathrm{P}$ for some $\mathrm{P} \in \mathcal{P}$, where $\mathcal{P}$ is given in (4.2).

Based on this equivalence relation, the equivalence class or coset of $\mathcal{P}$ in $\mathcal{G}_N$ that contains operator $\mathrm{A} \in \mathcal{G}_N$ is defined as

$$\mathcal{P}\mathrm{A} = \{\mathrm{P} \cdot \mathrm{A} : \forall\, \mathrm{P} \in \mathcal{P}\}.$$

We say that A is the representative of coset $\mathcal{P}$A. If we now look back to Proposition 1 and expression (4.2), it is straightforward to derive the following proposition.

***Proposition* 3.** *Given an arbitrary coset* $\mathcal{P}$A *of* $\mathcal{P}$ *in* $\mathcal{G}_N$*, then* $\mathbf{A} \equiv \mathrm{A}$ *is the unique operator in* $\mathcal{G}_N$ *such that* $\mathbf{A} \in \mathcal{P}\mathrm{A} \cap \Pi^{\otimes N}$*, i.e., it belongs to the coset and to the subset* $\Pi^{\otimes N}$*.*

Based on this last proposition, we can take all the elements of $\Pi^{\otimes N}$ as the representatives of all the cosets of $\mathcal{P}$ in $\mathcal{G}_N$. The total number of cosets, which we denote as $|\mathcal{P} : \mathcal{G}_N|$, will thus be given by the cardinality of $\Pi^{\otimes N}$, i.e., 2^{2N}. Proposition 3 also allows us to partition the underlying set of the N-fold Pauli group $\mathcal{G}_N$, $\widetilde{\Pi}^{\otimes N}$, as

$$\widetilde{\Pi}^{\otimes N} = \bigcup_{\mathbf{A} \in \Pi^{\otimes N}} \mathcal{P}\mathbf{A},$$

where $\mathbf{A}$ runs through all the elements of $\Pi^{\otimes N}$.

Let us now consider the set of all cosets of $\mathcal{P}$ in $\mathcal{G}_N$, that is, $Q = \{\mathcal{P}\mathbf{A} \subset \widetilde{\Pi}^{\otimes N} : \forall\, \mathbf{A} \in \Pi^{\otimes N}\}$. The quotient group $\mathcal{G}_N/\mathcal{P}$ is defined as $\mathcal{G}_N/\mathcal{P} = (Q, \bullet)$ where the group operation $\bullet$ is defined as

$$(\mathcal{P}\mathrm{A}) \bullet (\mathcal{P}\mathrm{B}) = \mathcal{P}[\mathrm{A} \cdot \mathrm{B}], \tag{4.4}$$

where $\mathcal{P}[\mathrm{A} \cdot \mathrm{B}]$ denotes the coset that contains the operator $\mathrm{A} \cdot \mathrm{B}$. From Proposition 2, we can use $\mathbf{A} \star \mathbf{B} \in \Pi^{\otimes N}$ as the representative of coset $\mathcal{P}[\mathrm{A} \cdot \mathrm{B}]$. That is to say,

$$(\mathcal{P}\mathrm{A}) \bullet (\mathcal{P}\mathrm{B}) = \mathcal{P}[\mathbf{A} \star \mathbf{B}]. \tag{4.5}$$

Having introduced the quotient group $\mathcal{G}_N/\mathcal{P}$, we can now define an isomorphism between $\mathcal{G}_N/\mathcal{P}$ and $\overline{\mathcal{G}}_N$.

***Definition* 1.** *Let α be the (one to one) mapping:*

$$\alpha : Q \to \Pi^{\otimes N},$$

defined as $\alpha(\mathcal{P} \cdot \mathrm{A}) = \mathbf{A}$. In other words, α maps a coset of $\mathcal{P}$ in $\mathcal{G}_N$ to its unique representative in $\Pi^{\otimes N}$.

***Proposition* 4.** *The mapping α is an isomorphism between $\mathcal{G}_N/\mathcal{P}$ and $\overline{\mathcal{G}}_N$.*

Proof: First, by Proposition 3, α is a bijective (one to one) mapping. Second, we must show that for all $\mathrm{A}, \mathrm{B} \in \mathcal{G}_N$

$$\alpha(\mathcal{P}\mathrm{A} \bullet \mathcal{P}\mathrm{B}) = \alpha(\mathcal{P}\mathrm{A}) \star \alpha(\mathcal{P}\mathrm{B}).$$

From (4.5), $(\mathcal{P}\mathrm{A}) \bullet (\mathcal{P}\mathrm{B}) = \mathcal{P}[\mathbf{A} \star \mathbf{B}]$. Thus,

$$\alpha(\mathcal{P}\mathrm{A} \bullet \mathcal{P}\mathrm{B}) = \alpha(\mathcal{P}[\mathbf{A} \star \mathbf{B}]) = \mathbf{A} \star \mathbf{B} = \alpha(\mathcal{P}\mathrm{A}) \star \alpha(\mathcal{P}\mathrm{B}),$$

as we wanted to prove.

□

The isomorphism α establishes that considering the effective Pauli group $\overline{\mathcal{G}}_N = (\Pi^{\otimes N}, \star)$ is analogous to working with equivalence classes of the Pauli group. This distinction is sometimes neglected in the literature, which may result in confusion when the notation is abused by referencing the Pauli group $\mathcal{G}_N$ instead of the effective Pauli group $\overline{\mathcal{G}}_N$. Because the global phase of a quantum state cannot be measured, quantum error correction will exclusively deal with elements of the effective Pauli group, meaning that quantum codes are capable of correcting qubits only up to a phase. Once again, given that the global phase has no observable consequence, being unable to consider the global phase has no impact on the error correcting capabilities of quantum codes. Instead, this result serves to simplify the framework of QEC codes as, strictly for the purposes of error correction, we only need to consider elements of the effective Pauli group.

Unfortunately, because the group operation $\star$ is not able to convey the commutation properties that exist among Pauli operators under the $\cdot$ product, the effective Pauli group is not sufficient on its own for appropriate QEC design. This will become clearer when we present the concept of quantum syndromes later on in this chapter. For now, assume that it is not possible to perform error correction based only on $\overline{\mathcal{G}}_N$ without recovering the commutation properties that exist in $\mathcal{G}_N$.

4.1.2 THE SYMPLECTIC REPRESENTATION

This issue of lost commutation relations can be overcome by adopting the symplectic representation [75, 128] of the Pauli operators in $\overline{\mathcal{G}}_N$. More specifically, by means of the symplectic mapping $\beta : \Pi^{\otimes N} \to \mathbb{F}_2^{2N}$, which is an isomorphism between the group $\overline{\mathcal{G}}_N = (\Pi^{\otimes N}, \star)$ and the group $(\mathbb{F}_2^{2N}, \oplus)$ of $2N$ binary-tuples under the mod 2 sum operation, we are able to recover the commutation properties that are lost when working with the operators of $\overline{\mathcal{G}}_N$. For clarity, throughout the remainder of this chapter, lower case boldface romans without a subscript will be used to denote $2N$ binary-tuples that belong to $\mathbb{F}_2^{2N}$, and lower case boldface romans with a subscript will be used to denote N binary-tuples that belong to $\mathbb{F}_2^N$.

The symplectic mapping β is defined as

$$\beta(\mathbf{A}) = \mathbf{a} = (\mathbf{a}_x|\mathbf{a}_z),\ \mathbf{a}_x, \mathbf{a}_z \in \mathbb{F}_2^N,$$

where the values of the entries of $\mathbf{a}_x$ and $\mathbf{a}_z$ at position $i = 1, \ldots, N$, are directly dependent on the single qubit Pauli operator, $[\mathbf{A}]_i$, located at the i-th position in the tensor product that makes up $\mathbf{A}$. More specifically,

$$\begin{aligned}
\beta([\mathbf{A}]_i = I) &= ([\mathbf{a}_x]_i = 0 \mid [\mathbf{a}_z]_i = 0) \\
\beta([\mathbf{A}]_i = X) &= ([\mathbf{a}_x]_i = 1 \mid [\mathbf{a}_z]_i = 0) \\
\beta([\mathbf{A}]_i = Z) &= ([\mathbf{a}_x]_i = 0 \mid [\mathbf{a}_z]_i = 1) \\
\beta([\mathbf{A}]_i = Y) &= ([\mathbf{a}_x]_i = 1 \mid [\mathbf{a}_z]_i = 1).
\end{aligned} \tag{4.6}$$

***Proposition* 5.** *The symplectic map β is an isomorphism $\overline{\mathcal{G}}_N \simeq (\mathbb{F}_2^{2N}, \oplus)$.*

Proof: By construction β is bijective and for all $\mathbf{A}, \mathbf{B} \in \overline{\mathcal{G}}_N$, it can be easily checked that $\beta(\mathbf{A} \star \mathbf{B}) = \beta(\mathbf{A}) \oplus \beta(\mathbf{B})$.

□

Note that the commutation properties of the Pauli operators with regard to the $\cdot$ product are not recovered by just defining the isomorphism β (after all, $(\mathbb{F}_2^{2N}, \oplus)$ is an abelian group). For this purpose, we define the symplectic scalar product, $\mathbf{a} \odot \mathbf{b} \in \mathbb{F}_2\ \forall\ \mathbf{a}, \mathbf{b} \in \mathbb{F}_2^{2N}$ as

$$\mathbf{a} \odot \mathbf{b} \triangleq (\mathbf{a}_x \circledast \mathbf{b}_z) \oplus (\mathbf{a}_z \circledast \mathbf{b}_x). \tag{4.7}$$

where $\circledast$ is the standard mod 2 inner product defined on $(\mathbb{F}_2^N, \oplus)$ considered as a vector space over the field $\mathbb{F}_2$.

***Proposition* 6.** *Any two operators $\mathbf{A}, \mathbf{B}$ in $\overline{\mathcal{G}}_N$ will either commute or anticommute (with respect to the group operation $\cdot$ in $\mathcal{G}_N$) if and only if the symplectic scalar product between $\beta(\mathbf{A}) = \mathbf{a}$ and $\beta(\mathbf{B}) = \mathbf{b}$, i.e., $\mathbf{a} \odot \mathbf{b}$, takes the value 0 or 1, respectively* [97, 127].

Fig. 4.1 summarizes the isomorphisms α and β.

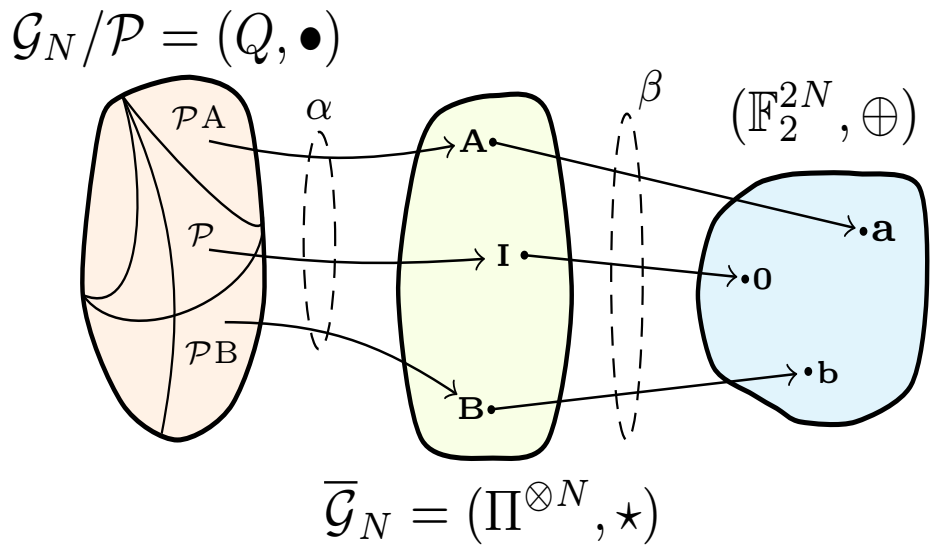


Figure 4.1: *Isomorphism between $\mathcal{G}_N/\mathcal{P}$ and $\overline{\mathcal{G}}_N$ defined by the mapping $\alpha : (Q, \bullet) \longrightarrow (\Pi^{\otimes N}, \star)$ and isomorphism between $\overline{\mathcal{G}}_N$ and $\mathbb{F}_2^{2N}$ defined by the symplectic mapping $\beta : (\Pi^{\otimes N}, \star) \longrightarrow (\mathbb{F}_2^{2N}, \oplus)$. The operator $\mathbf{I}$ represents the N qubit identity operator, i.e., $\mathbf{I} = I^{\otimes N}$.*

4.2 STABILIZER-BASED ERROR CORRECTION

The fact that quantum states cannot be directly measured has been mentioned numerous times throughout this dissertation. However, we also know that it is impossible to perform error correction without any knowledge regarding how decoherence is acting on our quantum information. Thus, in order for QEC to function, a method capable of extracting information about decoherence-related errors without actually measuring the quantum states themselves is necessary. Fortunately, this issue can be circumvented with the stabilizer formalism, which allows us to gleam information about errors without having to look at the actual quantum information by measuring the so-called *quantum syndrome* [88, 75, 97, 127]. Prior to diving into the specifics of stabilizer codes, it is worthwhile to provide a more general view of the stabilizer formalism itself, as its applications ex-

tend far beyond the niche of QEC. This is best achieved using the following example, which was originally introduced in [75].

Consider the EPR state $|\Phi^+\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$ given in (3.12). Now assume that we apply the operations X_1X_2 and Z_1Z_2 to $|\Phi^+\rangle$, where the notation X_iZ_j describes the action of an X Pauli gate on the i-th qubit and a Z Pauli gate on the j-th qubit. Notice how these operations leave the state unchanged, i.e, $X_1X_2\,|\Phi^+\rangle = |\Phi^+\rangle$ and $Z_1Z_2\,|\Phi^+\rangle = |\Phi^+\rangle$. In written form, we say that the state $|\Phi^+\rangle$ is stabilized by the operators X_1X_2 and Z_1Z_2. Although it may not seem advantageous from this small example, being able to represent quantum states by means of the operators that stabilize them is extremely practical. In fact, it becomes even more useful to describe quantum codes, which can be very difficult to represent in state vector notation and are much more compactly described using stabilizers.

After this short introduction to the stabilizer formalism, we move towards to the description of stabilizer coding principles from the perspective of group theory. Those familiar with these concepts will realize that this interpretation deviates from conventional representations and that the employed notation differs slightly from the one usually found in the literature [88, 75, 129, 130, 131]. This is meant to facilitate the comprehension of certain topics in the field of quantum stabilizer codes that are sometimes misunderstood.

4.2.1 THE STABILIZER GROUP

Let $\{\mathbf{s}_1, \mathbf{s}_2, \ldots, \mathbf{s}_{N-k}\}$ be a subset of independent vectors in the vector space $\mathbb{F}_2^{2N}$ over $\mathbb{F}_2$ that satisfies $\mathbf{s}_i \odot \mathbf{s}_j = 0$, $\forall\, i \neq j$. Let us now define the subspace $\mathcal{R}$ as the span of the set $\{\mathbf{s}_1, \mathbf{s}_2, \ldots, \mathbf{s}_{N-k}\}$. Thus, $\mathcal{R}$ has cardinality $|\mathcal{R}| = 2^{N-k}$. Based on these definitions, we can now use the isomorphism β (see proposition 5) to define the stabilizer group.

***Definition* 2.** *The stabilizer set $\overline{\mathcal{S}} \subset \Pi^{\otimes N}$ is defined as $\overline{\mathcal{S}} = \beta^{-1}(\mathcal{R})$, where β^{-1} is the inverse of the symplectic isomorphism in Proposition 5.*

***Proposition* 7.** *The set $\overline{\mathcal{S}}$ together with the $\star$ product, i.e., $(\overline{\mathcal{S}}, \star)$, is an abelian subgroup of $\overline{\mathcal{G}}_N$.*

Proof: That $(\overline{\mathcal{S}}, \star)$ is a subgroup of $\overline{\mathcal{G}}_N = (\Pi^{\otimes N}, \star)$ is straightforward from the fact that $\mathcal{R}$ is a subgroup of $(\mathbb{F}_2^{2N}, \oplus)$ and β is an isomorphism $\overline{\mathcal{G}}_N \backsimeq (\mathbb{F}_2^{2N}, \oplus)$. That $\overline{\mathcal{S}}$ is abelian follows from the fact that $\overline{\mathcal{G}}_N$ is abelian.

□

In what follows, for the purpose of simplicity, we will not distinguish between the group $(\overline{\mathcal{S}}, \star)$ and its underlying set $\overline{\mathcal{S}}$ (the group operation that applies is clear). Similarly, $\mathcal{R}$ will also represent $(\mathcal{R}, \oplus)$.

Note that the stabilizer group $\overline{\mathcal{S}} \subset \overline{\mathcal{G}}_N$ can be generated by the set of stabilizer generators $\{\mathbf{S}_v = \beta^{-1}(\mathbf{s}_v)\}_{v=1}^{N-k}$ in $\overline{\mathcal{G}}_N$. That is to say, for any $\mathbf{S} \in \overline{\mathcal{S}}$, there are $b_v \in \{0, 1\}$, $v = 1, \ldots, N-k$, such that

$$\mathbf{S} = \mathbf{S}_1^{b_1} \star \mathbf{S}_2^{b_2} \star \ldots \star \mathbf{S}_{N-k}^{b_{N-k}}.$$

Observe how, from Proposition 6, all the $\mathbf{S}_v$ operators will commute with respect to the group operation $\cdot$ in $\mathcal{G}_N$, since the symplectic products of their binary counterparts in $\mathbb{F}_2^{2N}$ are always zero.

4.2.1.1 Partition of $\mathbb{F}_2^{2N}$ into cosets

Next, we derive a partition of the group $(\mathbb{F}_2^{2N}, \oplus)$ into cosets. We will later relate this partition to the concepts of stabilizer codes.

***Definition* 3.** *Let* $\Gamma_{\mathcal{R}} \subset \mathbb{F}_2^{2N}$ *be the set*

$$\begin{aligned}\Gamma_{\mathcal{R}} &= \{\mathbf{a} \in \mathbb{F}_2^{2N} : \mathbf{a} \odot \mathbf{s} = 0,\ \forall \mathbf{s} \in \mathcal{R}\} \\ &= \{\mathbf{a} \in \mathbb{F}_2^{2N} : \mathbf{a} \odot \mathbf{s}_v = 0,\ v = 1 \ldots N-k\}.\end{aligned}$$

***Proposition* 8.** *The set* $\Gamma_{\mathcal{R}}$ *together with the modulo-2 sum, i.e.,* $(\Gamma_{\mathcal{R}}, \oplus)$*, is an abelian subgroup of* $(\mathbb{F}_2^{2N}, \oplus)$*.*

Proof: The identity element of $(\mathbb{F}_2^{2N}, \oplus)$, which is $\mathbf{0}$, is in $\Gamma_{\mathcal{R}}$ since $\mathbf{0} \odot \mathbf{s}_v = 0$ for all $v \in \{1 \ldots N-k\}$. On the other hand, one must show that $\mathbf{a} \oplus \mathbf{b} \in \Gamma_{\mathcal{R}}$, that is, we must show that $(\mathbf{a} \oplus \mathbf{b}) \odot \mathbf{s}_v = 0$, for all $v \in \{1 \ldots N-k\}$. Note that if $\mathbf{a}, \mathbf{b} \in \Gamma_{\mathcal{R}}$, then $\mathbf{a} \odot \mathbf{s}_v = 0$ and $\mathbf{b} \odot \mathbf{s}_v = 0$ for all v, so that,

$$\begin{aligned}(\mathbf{a} \oplus \mathbf{b}) \odot \mathbf{s}_v &= \mathbf{a} \odot \mathbf{s}_v \oplus \mathbf{b} \odot \mathbf{s}_v \\ &= 0 \oplus 0 = 0\end{aligned}$$

Therefore, we conclude that $(\Gamma_{\mathcal{R}}, \oplus)$ is an abelian group.

□

Once more, in what follows we will not distinguish between the group $(\Gamma_{\mathcal{R}}, \oplus)$ and its underlying set $\Gamma_{\mathcal{R}}$.

4.2.1.2 Partition of $\mathbb{F}_2^{2N}$ into cosets of $\Gamma_{\mathcal{R}}$

Based on the newly defined subgroup $\Gamma_{\mathcal{R}}$, we can partition the group $(\mathbb{F}_2^{2N}, \oplus)$ into cosets of this set in a similar manner to what was done previously for $\mathcal{G}_N$ with respect to its subgroup $\mathcal{P}$ in section 4.1.1.

***Definition* 4.** *Define the equivalence relation* $\sim_\Gamma$ *in* $\mathbb{F}_2^{2N}$ *as*

$$\mathbf{a} \sim_\Gamma \mathbf{b} \text{ iff } \mathbf{b} = \mathbf{c} \oplus \mathbf{a} \text{ for some } \mathbf{c} \in \Gamma_{\mathcal{R}}.$$

Then, the equivalence class or coset, $\gamma\mathbf{a}$, of $\Gamma_{\mathcal{R}}$ in $\mathbb{F}_2^{2N}$ containing $\mathbf{a}$, is the set

$$\gamma\mathbf{a} \triangleq \Gamma_{\mathcal{R}} \oplus \mathbf{a} = \{\mathbf{c} \oplus \mathbf{a} : \forall\, \mathbf{c} \in \Gamma_{\mathcal{R}}\}.$$

This means that two elements $\mathbf{a}, \mathbf{b} \in \mathbb{F}_2^{2N}$ will belong to the same coset if and only if $\mathbf{a} \oplus \mathbf{b} \in \Gamma_{\mathcal{R}}$.

***Proposition* 9.** *The number of cosets of* $\Gamma_{\mathcal{R}}$ *in* $\mathbb{F}_2^{2N}$ *is* $|\mathbb{F}_2^{2N} : \Gamma_{\mathcal{R}}| = 2^{N-k}$.

Proof: Define the syndrome[1] vector $\mathbf{w} \in \{0,1\}^{N-k}$ (we abuse notation slightly by using a lower case boldface roman without a subscript to denote a vector in $\mathbb{F}_2^{N-k}$) associated to each $\mathbf{a} \in \mathbb{F}_2^{2N}$ as

$$\mathrm{w}_v = \mathbf{a} \odot \mathbf{s}_v,\ v = 1, \ldots, N-k. \tag{4.8}$$

Observe from Definition 3 that a vector $\mathbf{a} \in \mathbb{F}_2^{2N}$ has syndrome zero, $\mathbf{w} = \mathbf{0}$, where $\mathbf{0} = [0_1, \ldots, 0_{N-k}]$, if and only if $\mathbf{a} \in \gamma\mathbf{0} = \Gamma_{\mathcal{R}}$. Furthermore, since two vectors $\mathbf{a}$ and $\mathbf{b}$ will belong to the same coset iff $\mathbf{a} \oplus \mathbf{b} \in \Gamma_{\mathcal{R}}$, all the vectors in a coset have the same syndrome, and two different cosets will have two different syndromes. Consequently, since there are 2^{N-k} different syndromes, there will be 2^{N-k} different cosets.

□

Notice that we have partitioned $\mathbb{F}_2^{2N}$ into cosets of $\Gamma_{\mathcal{R}}$. We denote the corresponding coset representatives as $\{\mathbf{t}_1, \ldots, \mathbf{t}_{2^{N-k}}\} \in \mathbb{F}_2^{2N}$. Each $\mathbf{t}_i$ is associated to a different syndrome vector $\mathbf{w}_i$, so that for all $i = 1, \ldots, 2^{N-k}$,

$$\mathbf{t}_i \odot (\mathbf{s}_1, \ldots, \mathbf{s}_{N-k}) = \mathbf{w}_i. \tag{4.9}$$

[1]The concept of syndromes in quantum error correction is discussed later on in this chapter. Rigorous explanations on this topic can also be found in [75, 97].

Thus, the group $(\mathbb{F}_2^{2N}, \oplus)$ can be partitioned as

$$\mathbb{F}_2^{2N} = \bigcup_{i=1}^{2^{N-k}} \underbrace{\mathbf{t}_i \oplus \Gamma_{\mathcal{R}}}_{\gamma \mathbf{t}_i} = \Gamma_{\mathcal{R}} \bigcup \left[\bigcup_{i=2}^{2^{N-k}} \mathbf{t}_i \oplus \Gamma_{\mathcal{R}} \right], \tag{4.10}$$

where, without loss of generality, the representative of coset $\Gamma_{\mathcal{R}} \oplus \mathbf{t}_1$ has been chosen to be $\mathbf{0}$. We denote this partition of $\mathbb{F}_2^{2N}$ into cosets of $\Gamma_{\mathcal{R}}$ by $\mathbb{F}_2^{2N}/\Gamma_{\mathcal{R}}$.

Since the cardinality of $\mathbb{F}_2^{2N}$ is 2^{2N} and $|\mathbb{F}_2^{2N}| = |\mathbb{F}_2^{2N} : \Gamma_{\mathcal{R}}||\Gamma_{\mathcal{R}}|$, we conclude that $|\Gamma_{\mathcal{R}}| = 2^{N+k}$. Recall that $|\mathbb{F}_2^{2N} : \Gamma_{\mathcal{R}}| = 2^{N-k}$ (Proposition 9).

4.2.1.3 Partition of $\Gamma_{\mathcal{R}}$ into cosets of $\mathcal{R}$

The group $\Gamma_{\mathcal{R}}$ can itself be partitioned into cosets of its subgroup $\mathcal{R} \subset \Gamma_{\mathcal{R}}$ by defining the following equivalence relation.

***Definition* 5.** *Define the equivalence relation* $\sim_{\mathcal{R}}$ *for all the elements* $\mathbf{a}, \mathbf{b} \in \Gamma_{\mathcal{R}}$ *as*

$$\mathbf{a} \sim_{\mathcal{R}} \mathbf{b} \;\; \textit{iff } \mathbf{b} = \mathbf{a} \oplus \mathbf{s}, \textit{ for some } \mathbf{s} \in \mathcal{R}.$$

This coset of $\mathcal{R}$ in $\Gamma_{\mathcal{R}}$ containing $\mathbf{a}$ induced by this relation, $\mathcal{R}\mathbf{a}$, is the subset of $\Gamma_{\mathcal{R}}$

$$\mathcal{R}\mathbf{a} \triangleq \mathbf{a} \oplus \mathcal{R} = \{\mathbf{a} \oplus \mathbf{s}_v : \forall\, \mathbf{s}_v \in \mathcal{R}\}.$$

As with the previous equivalence relation given in Definition 4, two vectors $\mathbf{a}, \mathbf{b} \in \Gamma_{\mathcal{R}}$ belong to the same coset if and only if $\mathbf{a} \oplus \mathbf{b} \in \mathcal{R}$.

***Proposition* 10.** *The number of cosets of* $\mathcal{R}$ *in* $\Gamma_{\mathcal{R}}$ *is given by* $|\Gamma_{\mathcal{R}} : \mathcal{R}| = 2^{2k}$.

Proof: Knowing that $|\Gamma_{\mathcal{R}}| = |\Gamma_{\mathcal{R}} : \mathcal{R}||\mathcal{R}|$, with $|\mathcal{R}| = 2^{N-k}$, and $|\Gamma_{\mathcal{R}}| = 2^{N+k}$, we can conclude that the number of cosets of $\mathcal{R}$ in $\Gamma_{\mathcal{R}}$ is $|\Gamma_{\mathcal{R}} : \mathcal{R}| = 2^{2k}$.

□

If we denote the set of representatives as $\{\mathbf{l}_j \in \Gamma_{\mathcal{R}}\}_{j=1}^{2^{2k}}$, one obtains the following $\Gamma_{\mathcal{R}}/\mathcal{R}$ partition:

$$\Gamma_{\mathcal{R}} = \bigcup_{j=1}^{2^{2k}} \underbrace{\mathbf{l}_j \oplus \mathcal{R}}_{\mathcal{R}\mathbf{l}_j} = \mathcal{R} \bigcup \left[\bigcup_{j=2}^{2^{2k}} \mathbf{l}_j \oplus \mathcal{R} \right], \tag{4.11}$$

where, once again, we have chosen $\mathbf{l}_1 = \mathbf{0}$ for convenience. Note that all these coset representatives satisfy

$$\mathbf{l}_j \odot (\mathbf{s}_1, \ldots, \mathbf{s}_{N-k}) = 0, \tag{4.12}$$

where $j = 1, \ldots, 2^{2k}$.

4.2.1.4 Partition of $\mathbb{F}_2^{2N}$ into cosets of $\mathcal{R}$

By combining the partitions $\mathbb{F}_2^{2N}/\Gamma_{\mathcal{R}}$ in (4.10) and $\Gamma_{\mathcal{R}}/\mathcal{R}$ in (4.11), the partition $\mathbb{F}_2^{2N}/\Gamma_{\mathcal{R}}/\mathcal{R}$ can be obtained. That is, $\mathbb{F}_2^{2N}$ can be partitioned as the union of the cosets of $\mathcal{R}$ in $\mathbb{F}_2^{2N}$ with coset representatives $\{\mathbf{t}_i \oplus \mathbf{l}_j\}$:

$$\mathbb{F}_2^{2N} = \bigcup_{i=1}^{2^{N-k}} \bigcup_{j=1}^{2^{2k}} (\mathbf{t}_i \oplus \mathbf{l}_j) \oplus \mathcal{R}. \tag{4.13}$$

4.2.2 PARTITION OF $\overline{\mathcal{G}}_N$ INTO COSETS

At this point, we are now in a position to use the partition $\mathbb{F}_2^{2N}/\Gamma_{\mathcal{R}}/\mathcal{R}$ and the inverse of the isomorphism β to establish a partition over $\overline{\mathcal{G}}_N$. We begin by defining the effective centralizer group of a stabilizer $\overline{\mathcal{S}}$ in $\overline{\mathcal{G}}_N$.

***Definition* 6.** *The effective centralizer $\overline{\mathcal{Z}}(\overline{\mathcal{S}}) \subset \Pi^{\otimes N}$ of stabilizer $\overline{\mathcal{S}}$ is the set obtained by applying the inverse of the isomorphism β to the set $\Gamma_{\mathcal{R}}$ (refer to Definition 3). That is to say, $\overline{\mathcal{Z}}(\overline{\mathcal{S}}) = \beta^{-1}(\Gamma_{\mathcal{R}})$, where*

$$\beta^{-1}(\Gamma_{\mathcal{R}}) = \{\beta^{-1}(\mathbf{a}) \in \overline{\mathcal{G}}_N : \forall\, \mathbf{a} \in \mathbb{F}_2^{2N},\, \mathbf{a} \odot \mathbf{s}_v = 0,$$

$$v = 1, \ldots, N-k\}.$$

Therefore, $\overline{\mathcal{Z}}(\overline{\mathcal{S}})$ is the set of all operators in $\overline{\mathcal{G}}_N$ that commute with all the stabilizer generators $\mathbf{S}_v \in \overline{\mathcal{S}} \subset \overline{\mathcal{G}}_N$ with regard to the group operation

$\cdot$ defined in $\mathcal{G}_N$. Thus,

$$\overline{\mathcal{Z}}(\overline{\mathcal{S}}) = \{\mathbf{A} \in \overline{\mathcal{G}}_N \subset \mathcal{G}_N : \mathbf{A} \cdot \mathbf{S}_v = \mathbf{S}_v \cdot \mathbf{A},\ v = 1, \ldots, N-k\}$$

$$= \{\mathbf{A} \in \overline{\mathcal{G}}_N : \underbrace{\beta(\mathbf{A})}_{\mathbf{a}} \odot \underbrace{\beta(\mathbf{S}_v)}_{\mathbf{s}_v} = 0,\ v = 1, \ldots, N-k\}.$$

***Proposition* 11.** *The effective centralizer $\overline{\mathcal{Z}}(\overline{\mathcal{S}})$ is a subgroup of $\overline{\mathcal{G}}_N = (\Pi^{\otimes N}, \star)$.*

Proof: This is straightforward from the fact that β is an isomorphism $\overline{\mathcal{G}}_N \backsimeq (\mathbb{F}_2^{2N}, \oplus)$ and $\Gamma_{\mathcal{R}}$ is a subgroup of $\mathbb{F}_2^{2N}$ (Proposition 8).

□

Remark 1: In the literature of stabilizer codes, the stabilizer is defined as a subgroup of $(\mathcal{G}_N, \cdot)$ instead of $(\overline{\mathcal{G}}_N, \star)$, and it is denoted by $\mathcal{S}$. The relationship between $\mathcal{S}$ and $\overline{\mathcal{S}}$ as defined in Definition 2 is

$$\mathcal{S} \triangleq \{\mathrm{A} \in \mathcal{G}_N : \mathrm{A} \equiv \mathbf{A}, \forall \mathbf{A} \in \overline{\mathcal{S}}\}.$$

Based on $\mathcal{S}$, the corresponding centralizer in $\mathcal{G}_N$ is

$$\mathcal{Z}(\mathcal{S}) \triangleq \{\mathrm{M} \in \mathcal{G}_N : \mathrm{M} \cdot \mathrm{S} = \mathrm{S} \cdot \mathrm{M}, \forall \mathrm{S} \in \mathcal{S}\}.$$

Remark 2: In the context of stabilizer codes, the concept of the normalizer set of $\mathcal{S}$, $\mathcal{N}(\mathcal{S})$, is often referenced. The normalizer of a stabilizer is defined as

$$\mathcal{N}(\mathcal{S}) = \{\mathrm{M} \in \mathcal{G}_N : \mathrm{M} \cdot \mathrm{S} \cdot \mathrm{M}^\dagger \in \mathcal{S}, \forall \mathrm{S} \in \mathcal{S}\}.$$

As it turns out, due to the product properties of Pauli operators, the normalizer and centralizer of a stabilizer group are actually the same set, $\mathcal{N}(\mathcal{S}) = \mathcal{Z}(\mathcal{S})$ [75, 131]. This can be seen by multiplying the conditional statement for the centralizer on the right by $\mathrm{M}^\dagger \in \mathcal{G}_N$ as

$$\mathrm{M} \cdot \mathrm{S} \cdot \mathrm{M}^\dagger = \mathrm{S} \cdot \mathrm{M} \cdot \mathrm{M}^\dagger \rightarrow \mathrm{M} \cdot \mathrm{S} \cdot \mathrm{M}^\dagger = \mathrm{S} \rightarrow \mathrm{M} \cdot \mathrm{S} \cdot \mathrm{M}^\dagger \in \mathcal{S}.$$

where $\mathrm{M}^{\dagger}\cdot\mathrm{M} = I^{\otimes N}$ holds because $\mathrm{M} \in \mathcal{G}_N$. Consequently, the effective normalizer over $\overline{\mathcal{G}}_N$ will satisfy $\overline{\mathcal{N}}(\overline{\mathcal{S}}) = \overline{\mathcal{Z}}(\overline{\mathcal{S}})$.

Having shown that it makes good physical sense to neglect the global phase (see section 3.1.2), and given that the commutation properties of Pauli operators with regard to the $\cdot$ product are recovered based on the isomorphism β, in order to preserve notation integrity, moving forward we will work with the concepts defined over the group $\overline{\mathcal{G}}_N = (\Pi^{\otimes N}, \star)$.

4.2.2.1 Partition $\overline{\mathcal{G}}_N/\overline{\mathcal{Z}}(\overline{\mathcal{S}})$

By applying the inverse isomorphism β^{-1} to the partition $\mathbb{F}_2^{2N}/\Gamma_{\mathcal{R}}$ in (4.10), we obtain

$$\begin{aligned} \overline{\mathcal{G}}_N &= \bigcup_{i=1}^{2^{N-k}} \mathbf{T}_i \star \overline{\mathcal{Z}}(\overline{\mathcal{S}}) \\ &= \overline{\mathcal{Z}}(\overline{\mathcal{S}}) \bigcup \left[\bigcup_{i=2}^{2^{N-k}} \mathbf{T}_i \star \overline{\mathcal{Z}}(\overline{\mathcal{S}}) \right], \end{aligned} \tag{4.14}$$

where $\mathbf{T}_i = \beta^{-1}(\mathbf{t}_i)$. Note from expression (4.9) that, except for $\mathbf{T}_1 = \beta(\mathbf{t}_1) = \beta(\mathbf{0}) = I^{\otimes N}$, which is associated to the zero syndrome vector, each $\mathbf{T}_i$ is related to a unique non-zero syndrome vector $\mathbf{w}_i$. That is, $\forall i = 2, \ldots, 2^{N-k}$,

$$\beta(\mathbf{T}_i) \odot (\beta(\mathbf{S}_1), \ldots, \beta(\mathbf{S}_{N-k})) = \mathbf{w}_i \neq \mathbf{0}. \tag{4.15}$$

4.2.2.2 Partition $\overline{\mathcal{Z}}(\overline{\mathcal{S}})/\overline{\mathcal{S}}$

Applying this same inverse isomorphism β^{-1} to the partition (4.11) provides us with the corresponding partition of the effective centralizer,

$$\overline{\mathcal{Z}}(\overline{\mathcal{S}}) = \bigcup_{j=1}^{2^{2k}} \mathbf{L}_j \star \overline{\mathcal{S}} = \overline{\mathcal{S}} \bigcup \left[\bigcup_{j=2}^{2^{2k}} \mathbf{L}_j \star \overline{\mathcal{S}} \right], \tag{4.16}$$

where $\mathbf{L}_j = \beta^{-1}(\mathbf{l}_j)$ and $\overline{\mathcal{S}} = \beta^{-1}(\mathcal{R})$.

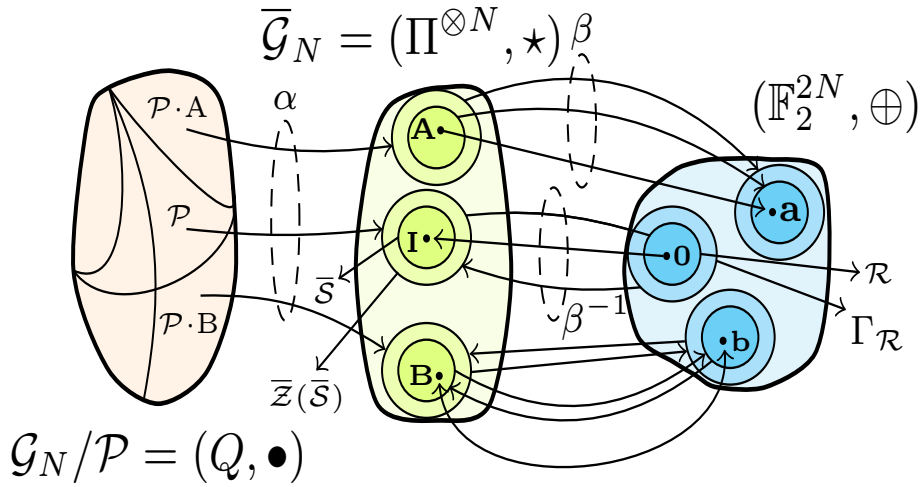


Figure 4.2: *Partitions of the effective Pauli group* $\overline{\mathcal{G}}_N = (\Pi^{\otimes N}, \star)$ *and the group of length* $2N$ *binary tuples* $(\mathbb{F}_2^{2N}, \oplus)$ *and how they are related by the isomorphism* β *and its inverse* β^{-1}*. Operator* $\mathbf{I}$ *represents the* N *qubit identity operator, i.e.,* $\mathbf{I} = I^{\otimes N}$*.*

Finally, by combining these last two partitions, we obtain the partition $\overline{\mathcal{G}}_N/\overline{\mathcal{Z}}(\overline{\mathcal{S}})/\overline{\mathcal{S}}$. In other words, we derive the partition of the entire effective Pauli group into cosets of the stabilizer $\overline{\mathcal{S}}$,

$$\overline{\mathcal{G}}_N = \bigcup_{i=1}^{2^{N-k}} \bigcup_{j=1}^{2^{2k}} (\mathbf{T}_i \star \mathbf{L}_j) \star \overline{\mathcal{S}}. \tag{4.17}$$

Figure 4.2 serves to graphically portray the relationships between the concepts that have been discussed up to this point.

From our earlier definition of the representatives $\{\mathbf{l}_j\}_{j=1}^{2^{2k}}$, we know that, except for $\mathbf{l}_1 = \mathbf{0}$ which is the representative of the set $\mathcal{R}$ itself, they belong to the set $\Gamma_{\mathcal{R}}$ but not to the set $\mathcal{R}$. This means that these same

representatives over $\overline{\mathcal{G}}_N$, $\{\mathbf{L}_j\}_{j=2}^{2^{2k}}$, belong to the set $\overline{\mathcal{Z}}(\overline{\mathcal{S}}) - \overline{\mathcal{S}}$ and satisfy

$$\beta(\mathbf{L}_j) \odot (\beta(\mathbf{S}_1), \ldots, \beta(\mathbf{S}_{N-k})) = \mathbf{0}. \tag{4.18}$$

In other words, the representatives $\{\mathbf{L}_j\}_{j=2}^{2^{2k}}$ commute with all the stabilizer elements but are not within the stabilizer.

4.2.3 CONSTRUCTION OF STABILIZER CODES

Recall that we defined the stabilizer group, $\overline{\mathcal{S}}$, as an abelian subgroup of $\overline{\mathcal{G}}_N$ that is obtained by applying the inverse isomorphism β^{-1} to the subspace $\mathcal{R} \in \mathbb{F}_2^{2N}$. The stabilizer $\overline{\mathcal{S}}$ has 2^{N-k} distinct elements, and is generated by $N-k$ independent generators $\mathbf{S}_i$, where $i = 1 \ldots N-k$. A stabilizer code can be described by a minimal set of these $N-k$ independent generators (the rest of the generators are linear combinations of the elements of the minimal set), as this provides a compact representation of the code. Based on this, we are now in the position to define a stabilizer code.

***Definition* 7.** *A stabilizer code $\mathcal{C}(\overline{\mathcal{S}})$ encodes k logical qubits into N physical qubits where its codespace, which is defined by the stabilizer $\overline{\mathcal{S}}$, can be written as*

$$\mathcal{C}(\overline{\mathcal{S}}) = \{|\psi\rangle \in \mathcal{H}_2^{\otimes N} : \mathbf{S}_i|\psi\rangle = |\psi\rangle, i = 1 \ldots N-k\},$$

where $\mathbf{S}_i|\psi\rangle \in \mathcal{H}_2^{\otimes N}$ is the evolution of state $|\psi\rangle$ under stabilizer generator $\mathbf{S}_i$.

Note that $\mathcal{C}(\overline{\mathcal{S}})$ is the subspace of $\mathcal{H}_2^{\otimes N}$ formed by the simultaneous $+1$-eigenspaces of all the operators in the stabilizer group $\overline{\mathcal{S}}$. Here $\mathcal{H}_2^{\otimes N}$ denotes the complex Hilbert space of dimension 2^N that comprises the state space of N-qubit systems.

4.2.3.1 Pure Errors, Logical Operators & Encoded Pauli operators

In the literature, stabilizer codes are generally defined over $\mathcal{G}_N$ (by using S_i in definition 7). Furthermore, the operators $\mathrm{L} \in \mathcal{G}_N$ that commute with the stabilizers elements (they map the codespace to itself) are called *logical operators* and those operators $\mathrm{T} \in \mathcal{G}_N$ that commute with the logical operators and anticommute with at least one stabilizer element are

known as *pure errors* [67, 68, 69, 70, 71, 163]. In our group theoretical interpretation, it is easy to see that the coset representatives $\{\mathbf{T}_i\}_{i=1}^{2^{N-k}}$ and $\{\mathbf{L}_j\}_{j=1}^{2^{2k}}$, of the cosets of $\overline{\mathcal{Z}}(\overline{\mathcal{S}})$ in $\overline{\mathcal{G}}_N$ (4.14) and of the cosets of $\overline{\mathcal{S}}$ in $\overline{\mathcal{Z}}(\overline{\mathcal{S}})$ (4.16), are the equivalent operators over $\overline{\mathcal{G}}_N$ of these pure errors and logical operators, respectively. This means that we can write $\{\mathrm{T} \in \mathcal{G}_N : \mathrm{T} \equiv \mathbf{T}\}$ and $\{\mathrm{L} \in \mathcal{G}_N : \mathrm{L} \equiv \mathbf{L}\}$.

Another important concept in the literature is that of the logical group $\mathcal{L} \in \mathcal{G}_N$ defined as $\mathcal{L} = \mathcal{N}(\mathcal{S})/\mathcal{S}$ [55]. Once again, we have an equivalent concept of this logical group over $\overline{\mathcal{G}}_N$ in our group theoretical framework: the partition of the effective centralizer into cosets of the stabilizer (see section 4.2.2.2). The logical group is generated by a set of operators known as encoded Pauli operators which are denoted by $\overline{Z}_q, \overline{X}_l \in \mathcal{G}_N$. Thus, we can write $\mathcal{L} = \{\overline{Z}_q, \overline{X}_l\}$ where $\{q, l\} \in \{1, \ldots, k\}$. This means that all the logical operators of a stabilizer code $\{\mathrm{L} \in \mathcal{G}_N\}$, can be obtained as linear combinations of the encoded Pauli operators. The encoded Pauli operators of an N-qubit stabilizer code are defined as those operators in $\mathcal{G}_N$ that commute with the elements of the stabilizer group and whose action on an encoded state can be understood as an X or Z operation on each of the logical qubits encoded by the stabilizer code. In this manner, $\overline{Z}_q$ represents an operator in $\mathcal{G}_N$ whose action is analogous to performing a Z operation (phase flip) on the q-th logical qubit. Thus $\overline{Z}_q \in \mathcal{G}_N$ maps to $Z_q \in \mathcal{G}_k$, where Z_q denotes the action of a Z operator on the q-th logical qubit and the action of I operators on the remaining $k-1$ logical qubits (identity operators are omitted). In consequence, each stabilizer code has $2k$ encoded Pauli operators. As previously, we define the equivalent encoded Pauli operators over the effective N-fold Pauli group as $\overline{Z} \equiv \overline{\mathbf{Z}}$ and $\overline{X} \equiv \overline{\mathbf{X}}$, where $\overline{\mathbf{Z}}, \overline{\mathbf{X}} \in \overline{\mathcal{G}}_N$.

Given the equivalence between the operators of $\mathcal{G}_N$ and $\overline{\mathcal{G}}_N$ (recall that they are equivalent in all relevant physical manners) and the commutation property preserving effect of the symplectic map, it makes sense in the context of QEC to consider that the stabilizer coding framework operates over the effective Pauli group. Thus, we adopt the nomenclature employed in the literature (which refers to concepts in $\mathcal{G}_N$) to reference the equivalent operators over $\overline{\mathcal{G}}_N$. For instance, we will refer to the coset representatives $\mathbf{T}_i$ and $\mathbf{L}_j$ as pure errors and logical operators, respectively. This terminology (the reasoning behind the names), can be better understood upon closer inspection of expressions (4.15) and (4.18).

From (4.15), we know that each $\mathbf{T}_i$ (except $\mathbf{T}_1$) is related to a unique non-zero syndrome vector $\mathbf{w}_i$. This means that each $\mathbf{T}_i$ has a unique commutation relation (with respect to the group operation $\cdot$ over $\mathcal{G}_N$) with the stabilizer generators $\{\mathbf{S}_v\}_{v=1}^{N-k}$: it anticommutes with at least one generator and commutes with the remaining ones. The term "pure" makes reference to how the singular commutation properties of each representative $\mathbf{T}_i$ are reflected solely by the corresponding syndrome $\mathbf{w}_i$.

From (4.18), we know that all the representatives $\mathbf{L}_j$ commute with all the stabilizer elements, but they do not belong to the stabilizer. This means that whenever an operator $\mathbf{L}_j$ acts on a quantum codeword, it will shift it from one stabilizer coset to another while leaving it within the same effective centralizer coset (it maps the codespace to itself). This means that the effects of each logical operator on a codeword can be interpreted as the operation of different single qubit Pauli operators on each of its logical qubits. It is for this reason that these operators are named "logical" operators. More specifically, in a scenario in which k logical qubits have been encoded into N physical qubits using a stabilizer code, logical operators are N-qubit operators whose action on the code results in an X, Y, Z or I operation being applied to each of the logical qubits. Since there are k logical qubits and 4 operators can be applied to each qubit, there are a total of $4^k = 2^{2k}$ logical operators.

The similarity between logical operators and encoded Pauli operators is worth noting. Because encoded Pauli operators define a generating set for the logical group, it is clear that they are also logical operators. Similarly, logical operators can be understood as linear combinations of encoded Pauli operators. Thus, one could simply refer to the encoded Pauli operators as the logical group generators. Although it may appear to complicate matters presently, the convention of referring to these concepts with different terms will be shown to serve an important purpose in the next chapter. For now, the primary takeaway is the fact that a stabilizer code has $2k$ encoded Pauli operators but 2^{2k} logical operators (just like how there are $N-k$ stabilizer generators but 2^{N-k} stabilizer elements).

Figure 4.3 depicts the partition of the effective N-fold Pauli group into the cosets indexed by logical operators and pure errors. Intuitively, this figure reflects how any operator $\mathbf{A} \in \overline{\mathcal{G}}_N$ can be decomposed into three distinct terms, each one associated to a pure error, a logical operator, and a stabilizer element. We will later show how this decomposition of the

operators in $\overline{\mathcal{G}}_N$ can be used to better approach and understand the task of decoding quantum stabilizer codes.

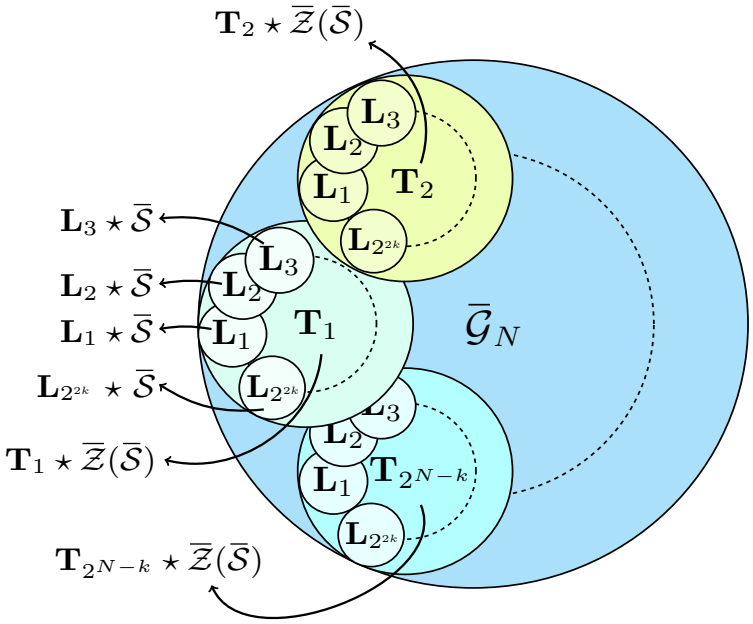


Figure 4.3: *Graphical representation of the partition of $\overline{\mathcal{G}}_N$ into cosets of the effective centralizer, $\overline{\mathcal{G}}_N = \bigcup_{i=1}^{2^{N-k}} \mathbf{T}_i \star \overline{\mathcal{Z}}(\overline{\mathcal{S}})$, and of the partition of the effective centralizer into cosets of the stabilizer group, $\overline{\mathcal{Z}}(\overline{\mathcal{S}}) = \bigcup_{j=1}^{2^{2k}} \mathbf{L}_j \star \overline{\mathcal{S}}$. Each effective centralizer coset is assigned a pure error $\mathbf{T}_i$ as its representative, and each stabilizer coset is assigned a logical operator $\mathbf{L}_j$ as its representative.*

4.2.3.2 Error detection using stabilizer codes

Let us assume we wish to transmit the information word $|\psi\rangle \in \mathcal{H}_2^{\otimes k}$ through some quantum channel ξ. First, we use a stabilizer code to encode the information word into the quantum codeword $|\overline{\psi}\rangle \in \mathcal{C}(\overline{\mathcal{S}}) \subset \mathcal{H}_2^{\otimes N}$. When the codeword $|\overline{\psi}\rangle$ is sent through the quantum channel ξ, it is exposed to the decoherence-emulating harmful effects of said channel, and a

corrupted quantum state $|\overline{\psi}\rangle_\xi$ is obtained at the channel output. To recover $|\overline{\psi}\rangle$ from $|\overline{\psi}\rangle_\xi$, which can then be used to obtain $|\psi\rangle$, the appropriate recovery operation must be applied. This requires some knowledge regarding the error induced by the channel, which must inevitably be derived from the corrupted output. However, we know that direct measurement of a quantum state needs to be avoided[2]. Instead, a methodology that avoids the measurement of quantum states while still gleaning sufficient information from the corrupted channel output to implement recovery operations is necessary. Fortunately, as we mentioned previously in section 4.2, this can be achieved by means of measuring the quantum syndrome, a strategy reminiscent of error syndrome measurements in classical coding scenarios [97, 75, 165, 164]. Let us discuss it.

Assume now that the quantum channel ξ is the generic Pauli channel ξ_P discussed in section 3.1.4.5. Any error induced by this channel will be an operator $\mathrm{E} \in \mathcal{G}_N$ that represents an N-fold tensor product of single qubit error operators E_u, where each E_u belongs to the single qubit Pauli group and $u = 1, ..., N$. This error acts on the encoded quantum state as $|\overline{\psi}\rangle_\xi = \mathrm{E}\,|\overline{\psi}\rangle$. As has been discussed previously, the global phase has no observable consequence, hence, using the notation introduced in (4.3), any $\mathrm{E} \in \mathcal{G}_N = (\widetilde{\Pi}^{\otimes N}, \cdot)$ will be taken as the operator $\mathbf{E} \equiv \mathrm{E}$ belonging to the effective Pauli group (i.e., $\mathbf{E} \in \overline{\mathcal{G}}_N = (\Pi^{\otimes N}, \star)$). Recall that although the commutation properties (in terms of the group operation $\cdot$) of the operators in $\overline{\mathcal{G}}_N$ are lost under the group operation $\star$, they can be recovered by applying the symplectic isomorphic mapping $\beta : \overline{\mathcal{G}}_N \to \mathbb{F}_2^{2N}$, together with the symplectic product $\odot$ in $\mathbb{F}_2^{2N}$ (refer to Proposition 6). Consequently, any error operator $\mathbf{E} \in \overline{\mathcal{G}}_N \subset \mathcal{G}_N$, will commute, in terms of the group operation in $\mathcal{G}_N$, with all the stabilizer generators $\mathbf{S}_v \in \overline{\mathcal{S}} \subset \overline{\mathcal{G}}_N \subset \mathcal{G}_N$, $v = 1, \ldots, N-k$, iff

$$\beta(\mathbf{E}) \odot \beta(\mathbf{S}_v) = 0.$$

On the other hand, if there is some index $v' \in \{1, \ldots, N-k\}$ where the above product takes the value 1, then $\mathbf{E}$ and $\mathbf{S}_j$ will anticommute, i.e., $\beta(\mathbf{E}) \odot \beta(\mathbf{S}'_v) = -\beta(\mathbf{S}_j) \odot \beta(\mathbf{E})$. Thus, the commutation properties of any $\mathbf{E} \in \overline{\mathcal{G}}_N$ with regard to the stabilizer generators $\{\mathbf{S}_v\}_{v=1}^{N-k}$, will be completely characterized by the syndrome vector $\mathbf{w} \in \mathbb{F}_2^{N-k}$ defined in (4.8). That is to say,

[2]Recall that measurement of a quantum state forces its superposition state to collapse, which in this case would result in the loss of the information regarding the initial state $|\psi\rangle$.

$$\mathbf{w} = \beta(\mathbf{E}) \odot (\beta(\mathbf{S}_1), \ldots, \beta(\mathbf{S}_{N-k})) = \mathbf{e} \odot (\mathbf{s}_1, \ldots, \mathbf{s}_{N-k}). \tag{4.19}$$

We can also write this as

$$\mathrm{E} \cdot \mathrm{S}_v = (-1)^{\mathrm{w}_v} \mathrm{S}_v \cdot \mathrm{E}, \tag{4.20}$$

or equivalently

$$\mathbf{e} \odot \mathbf{s}_v = (-1)^{\mathrm{w}_v} \mathbf{s}_v \odot \mathbf{e}, \tag{4.21}$$

where w_v denotes the v-th component of the syndrome $\mathbf{w} = [\mathrm{w}_1, \ldots, \mathrm{w}_{N-k}]$. Based on these expressions, and considering that $|\overline{\psi}\rangle_{\xi_P} = \mathrm{E}\,|\overline{\psi}\rangle$, we can write

$$\begin{aligned} \mathrm{S}_v\,|\overline{\psi}\rangle_{\xi_P} = \mathrm{S}_v \cdot \mathrm{E}\,|\overline{\psi}\rangle &= (-1)^{\mathrm{w}_v} \mathrm{E} \cdot \mathrm{S}_v\,|\overline{\psi}\rangle \\ &= (-1)^{\mathrm{w}_v} \mathrm{E}\,|\overline{\psi}\rangle = (-1)^{\mathrm{w}_v}\,|\overline{\psi}\rangle_{\xi_P}\,, \end{aligned} \tag{4.22}$$

where we recover the notation over $\mathcal{G}_N$ to preserve the commutation relations. From (4.22), we can tell that $|\overline{\psi}\rangle_{\xi_P}$ is an eigenstate of each stabilizer generator associated to the ± 1 eigenvalues. We can also tell from this expression that the components of the syndrome $\mathbf{w}$ will determine the value of each particular eigenvalue. Therefore, we know that it is possible to determine the commutation relations between the channel error and the stabilizer generators, i.e, measure the syndrome $\mathbf{w}$, by measuring the eigenvalues of the stabilizer generators of the code [128]. This can be achieved by using a circuit like the one shown in Figure 4.4.

4.2.3.3 Quantum Parity Check Matrix of a Stabilizer Code

In section 3.2.1 we defined the PCM of a linear block code as the matrix whose rows represent the linear constraints (the parity check equations) that make up the code. We can adapt this notion to stabilizer codes by building a matrix whose rows are the stabilizer generators. We define the

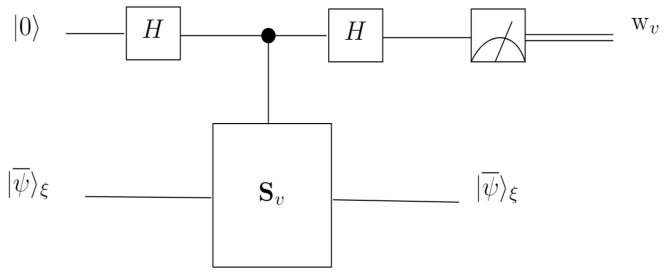


Figure 4.4: *Quantum circuit that measures the syndrome associated to each of the stabilizer generators. The H block represents the Hadamard gate defined in section 3.1.4.1 and the $\mathbf{S}_v$ block represents a controlled unitary gate where the stabilizer generator $\mathbf{S}_v$ is the unitary that is applied.*

Quantum Parity Check Matrix (QPCM) of a stabilizer code as the $N-k\times 2N$ matrix in $\mathbb{F}_2^{N-k\times 2N}$

$$\mathbf{H}_{\overline{\mathcal{S}}} = (\mathrm{H}_x|\mathrm{H}_z) \triangleq \begin{pmatrix} (\mathbf{s}_{1,x}|\mathbf{s}_{1,z}) \\ \vdots \\ (\mathbf{s}_{N-k,x}|\mathbf{s}_{N-k,z}) \end{pmatrix}.$$

Using this QPCM representation, we can rewrite the expression in (4.19) as $\mathbf{w} = \mathbf{e}\odot\mathbf{H}_{\overline{\mathcal{S}}} = \mathbf{H}_{\overline{\mathcal{S}}}\odot\mathbf{e}$. Recall that expression (4.7) relates the symplectic product $\odot$ with the mod 2 operations of $(\mathbb{F}_2^N, \oplus)$. Therefore, based on this relationship we can write

$$\mathbf{w} = \mathbf{H}_{\overline{\mathcal{S}}} \odot \mathbf{e} = \mathbf{e}_z\mathrm{H}_x^\top \oplus \mathbf{e}_x\mathrm{H}_z^\top. \tag{4.23}$$

Observe also that by construction $\beta(\mathbf{S}_i)\odot\beta(\mathbf{S}_j) = 0$, $i,j = 1,\dots N-k$ (i.e., all the stabilizer generators $\{\mathbf{S}_v\}_{v=1}^{N-k}$ commute with respect to the group operation $\cdot$ in $\mathcal{G}_N$). Therefore, matrix $\mathbf{H}_{\overline{\mathcal{S}}} = (\mathrm{H}_x|\mathrm{H}_z)$ should verify the following constraint, which is commonly known as the *symplectic criterion*,

$$\mathrm{H}_x\mathrm{H}_z^\top \oplus \mathrm{H}_z\mathrm{H}_x^\top = 0. \tag{4.24}$$

Constraint (4.24) is significant, because it specifies which existing classical codes can be used to design stabilizer codes. For instance, given two parity check matrices of any two classical codes of the same length and dimension, H_1 and H_2, the parity check matrix obtained as $\mathbf{H} = (\mathrm{H}_1|\mathrm{H}_2)$ will only define a stabilizer code if it satisfies (4.24), i.e., $\mathrm{H}_1\mathrm{H}_2^\top \oplus \mathrm{H}_2\mathrm{H}_1^\top = 0$.

4.2.4 DECODING QUANTUM STABILIZER CODES

From what we have seen thus far, we know that direct measurement of quantum information states is avoided by measuring a binary vector known as the quantum syndrome. Additionally, we know that the quantum syndrome can be written using QPCM notation as shown in (4.23) and that it can be obtained by measuring the eigenvalues of the generators of the stabilizer code. Physically, this task is performed by a circuit like the one shown in Figure 4.4. In what follows we discuss how stabilizer codes can be decoded based on the quantum syndrome, and we draw parallels to the classical decoding problem presented in section 3.2.2.2. It is worth mentioning that his has become a prominent research topic in the field of QEC [16, 18, 20, 54, 55, 56, 75, 131, 165].

For the sake of simplicity, we will assume operation in a framework that employs a stabilizer code to encode k logical qubits into N physical qubits and that transmits the encoded states over a generic Pauli channel. Furthermore, we will restrict the discussion to the sum-product based decoding of stabilizer codes, which (mostly) concerns QLDPC and QTC codes [46, 110, 111, 112]. Certain families of stabilizer codes, such as topological codes [72, 73, 74], can be decoded differently and so this discussion may not apply to them.

Based only on our knowledge of the decoding process of classical linear block codes (see section 3.2.2.2), we would approach the decoding of stabilizer codes by using a factor graph to represent (4.23) and then running the SPA algorithm to find the most likely error pattern (its symplectic representation to be more precise) associated to the measured syndrome $\mathbf{w}$. This can be written as

$$\hat{\mathbf{e}} = \arg\max_{\mathbf{e}} P(\mathbf{e}|\mathbf{w}). \tag{4.25}$$

However, in contrast to what happens in the classical domain, the solution given by (4.25) does not necessarily lead to optimal decoding performance for a stabilizer code. Let us discuss this counter-intuitive result.

4.2.4.1 Optimal decoding of quantum stabilizer codes

We showed earlier (see section 4.2.3.1) that any error operator $\mathbf{E} \in \overline{\mathcal{G}}_N$, can be decomposed into three different terms: a pure error, $\mathbf{T}$, which represents the effective centralizer coset that contains $\mathbf{E}$, a logical operator, $\mathbf{L}$, which represents the stabilizer coset that contains $\mathbf{E}$, and a stabilizer component, $\mathbf{S}$, which represents the specific operator in the stabilizer associated with $\mathbf{E}$. That is, any arbitrary N qubit effective Pauli error operator $\mathbf{E} \in \overline{\mathcal{G}}_N$ can be decomposed as

$$\mathbf{E} = \mathbf{T} \star \mathbf{L} \star \mathbf{S}, \tag{4.26}$$

where $\mathbf{T} \in \{\mathbf{T}_i\}_{i=1}^{2^{N-k}}$, $\mathbf{L} \in \{\mathbf{L}_j\}_{j=1}^{2^{2k}}$, and $\mathbf{S} \in \overline{\mathcal{S}}$, denote the pure error, the logical operator, and the stabilizer element involved in the decomposition of the error operator $\mathbf{E}$, respectively.

We know from (4.15) that the pure errors $\{\mathbf{T}_i = \beta^{-1}(\mathbf{t}_i)\}_{i=2}^{2^{N-k}}$ will anticommute, in terms of the group operation in $\mathcal{G}_N$, with at least one stabilizer generator and commute with the rest of them. This means that each $\mathbf{T}_i$ will have a unique combination of commutation relations with the stabilizer generators. However, we know from (4.18) that the logical operators $\{\mathbf{L}_j = \beta^{-1}(\mathbf{t}_j)\}_{j=1}^{2^{2k}}$ will commute (with regard to the group operation over $\mathcal{G}_N$) with the stabilizer generators, and we also know that all the stabilizer generators commute with each other. This means that, among the operators $\mathbf{T}$, $\mathbf{L}$, and $\mathbf{S}$, only the pure error component can possibly anti-commute (in terms of the group operation in $\mathcal{G}_N$) with the stabilizer generators. Therefore, while syndrome measurements serve to reveal the pure error component of the error, the logical and stabilizer components remain unknown.

This limitation in quantum syndrome decoding is similar to what happens when syndrome decoding is applied in the classical domain. In the classical scenario, the syndrome only identifies the coset of the code and not the specific error pattern. In order to optimize performance, classical syndrome decoders generally choose the coset leader (the most likely error pattern in the coset) as the error operator that has taken place. How-

ever, the quantum version of this problem is significantly more nuanced, as choices from various different cosets can be made: the identified centralizer coset is composed of 2^{2k} cosets of the stabilizer, each one of these represented by a logical operator $\mathbf{L}_j,\ j = 1 \dots 2^{2k}$. Fortunately, the stabilizer component of the decomposition, $\mathbf{S}$, is irrelevant in the error correction procedure and can be ignored in the decoding process[3].

Assuming that the quantum decoder finds the appropriate pure error component from the set $\{\mathbf{T}_i\}_{i=1}^{2^{N-k}}$ using the syndrome, the next step is to find the correct logical component, $\mathbf{L} \in \{\mathbf{L}_j\}_{j=1}^{2^{2k}}$, of the error $\mathbf{E}$. Therefore, optimal decoding for quantum stabilizer codes can be understood as the task of locating the most likely logical operator in $\{\mathbf{L}_j\}_{j=1}^{2^{2k}}$ involved in the decomposition of the error operator, $\mathbf{E}$, given the measured quantum syndrome $\mathbf{w}$. That is,

$$\hat{\mathbf{L}} = \underset{\mathbf{L} \in \{\mathbf{L}_j\}_{j=1}^{2^{2k}}}{\arg\max}\ P(\mathbf{L}|\mathbf{w}). \tag{4.27}$$

This decoding task can be seen as obtaining the coset $\mathbf{L}_j \star \overline{\mathcal{S}}$ that has the highest probability amidst those that have the same pure error component, $\mathbf{T}$. This means that the decoder has to locate the stabilizer coset indexed by the representative $\mathbf{L}$ that has the highest probability among all the stabilizer cosets that belong to the effective centralizer coset with representative $\mathbf{T}$, i.e., it must find the most likely coset $\mathbf{T} \star \mathbf{L} \star \overline{\mathcal{S}}$. Notice how the complexity increases in comparison to the classical syndrome decoding problem (4.25): In the classical domain, the most likely error pattern associated with a given syndrome is obvious (for a BSC, the error pattern with minimum weight in the coset), while in quantum decoding the solution to (4.27) is not obvious[4]. Therefore, the results obtained by a decoder following the classical decoding rule in (4.25) may differ from those obtained following the optimal quantum decoding rule given in (4.27). If a classical decoder is employed to decode a stabilizer code, the estimated error will be the Pauli operator of minimum weight among those belonging to the effective centralizer coset $\mathbf{T} \star \overline{\mathcal{Z}}(\overline{\mathcal{S}})$ associated to the syndrome. This is shown in [70], where by weight we refer to the same concept as that of the Hamming weight (see section 3.2.1);

[3]The reason for this is that any error operator contained in the same stabilizer coset as $\mathbf{S}$ will have the same effect on a codeword. This is discussed further in the next section.

[4]Notice that this solution is independent of the error operator introduced by the channel and only depends on the channel parameters and the code structure. Thus, it could be calculated for each one of the pure errors prior to the decoding process.

the number of non-identity elements of the error pattern. In [70], the strategy of decoding quantum stabilizer codes based on the classical rule given in (4.25) is called non-degenerate decoding or Quantum Maximum Likelihood Decoding (QMLD), while the optimal rule given in (4.27) is called degenerate decoding or Degenerate QMLD (DQMLD). Essentially, a decoder based on the QMLD strategy is "blind" to the partition of the coset $\mathbf{T} \star \overline{\mathcal{Z}}(\overline{\mathcal{S}})$ into cosets of the stabilizer. In contrast, the optimal quantum decoder is capable of finding the stabilizer coset $(\mathbf{T} \star \mathbf{L}) \star \overline{\mathcal{S}}$ which has the highest probability and will simply choose any operator contained in it. To do so, it must compute the probability of each stabilizer coset by adding the probabilities of all the operators that make up the coset, which is significantly harder than choosing the minimum-weight operator [55, 56, 70].

4.2.4.2 Optimal vs SPA decoding of quantum stabilizer codes

In order to discuss the differences between decoding based on (4.25) and (4.27), we redefine the coset partition of [53]. This is shown graphically in Figure 4.5. To comprehend this figure, recall the division of the effective N-fold Pauli group shown previously in Figure 4.3, where $\overline{\mathcal{G}}_N$ is partitioned into effective centralizer cosets $\mathbf{T}_i \star \overline{\mathcal{Z}}(\overline{\mathcal{S}})$, which themselves are partitioned into stabilizer cosets $\mathbf{L}_j \star \overline{\mathcal{S}}$, where $i = 1, \ldots, 2^{N-k}$ and $j = 1, \ldots, 2^{2k}$. In Figure 4.5, $\overline{\mathcal{G}}_N$ is represented by the outermost red bordered rectangle that contains the smaller shaded rectangles. These shaded rectangles portray the coset partition that divides $\overline{\mathcal{G}}_N$ into 2^{N-k} cosets of the effective centralizer. Each of these cosets is assigned a different representative from the set $\{\mathbf{T}_j\}_{j=1}^{2^{N-k}}$, where $\mathbf{T}_1 = \beta^{-1}(\mathbf{t}_1) = \beta^{-1}(\mathbf{0}) = I^{\otimes N}$. In addition, each of the shaded rectangles is itself divided into smaller rectangles. This subdivision represents the coset partition of each effective centralizer coset into cosets of the stabilizer. These stabilizer cosets are each indexed by a different logical operator $\{\mathbf{L}_j\}_{j=1}^{2^{2k}}$, where $\mathbf{L}_1 = \beta^{-1}(\mathbf{l}_1) = \beta^{-1}(\mathbf{0}) = I^{\otimes N}$. Within each stabilizer coset, effective Pauli operators that differ in terms of their stabilizer component can be found. However, knowing that all stabilizer elements have the same effect on a codeword, errors within each stabilizer coset need not be distinguished (this is studied in the next section).

Based on this graphical representation, the generic classical decoder defined by the rule in (4.25) will obtain the most likely error operator (i.e., the error operator with the lowest weight) for the received syndrome. Thus, it will identify the correct coset $\mathbf{T} \star \overline{\mathcal{Z}}(\overline{\mathcal{S}})$ and the corresponding pure error

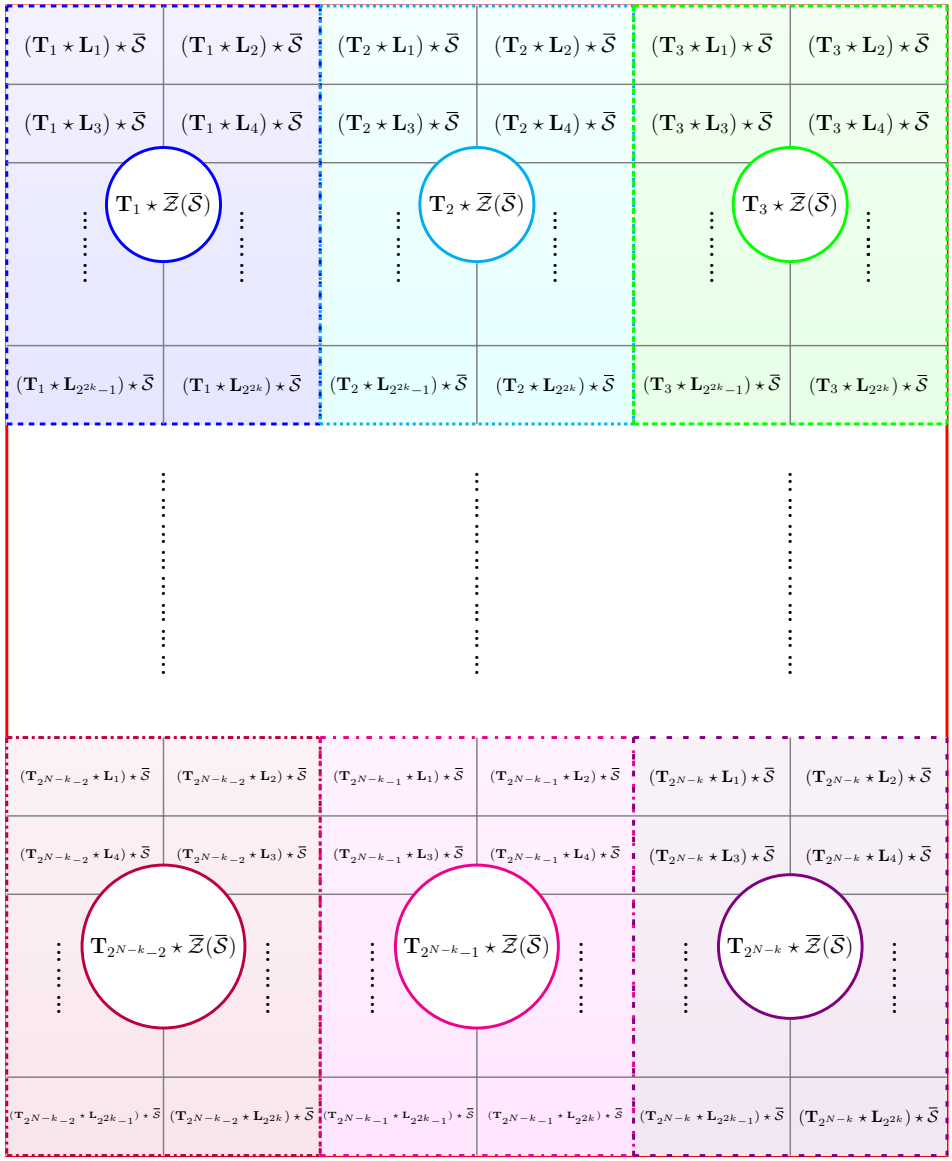


Figure 4.5: *Cosets of $\overline{\mathcal{S}}$ and $\overline{\mathcal{Z}}(\overline{\mathcal{S}})$ in $\overline{\mathcal{G}}_N$. The first coset partition is represented by the differently colored shaded rectangles and given in Eq. 4.14. The second coset partition is represented by the rectangles found within each of the effective centralizer cosets and given in (4.16).*

$\mathbf{T}$. However, as explained before, Figure 4.5 illustrates that this solution does not necessarily correspond to the one obtained by a quantum decoder that follows (4.27). To solve (4.27), the logical operator component of the error, $\mathbf{L}$, must be appropriately estimated. This can be understood graphically in Figure 4.5 as finding the appropriate stabilizer coset within the effective centralizer coset corresponding to the received syndrome. The classical decoder of (4.25) treats the entire effective centralizer coset as a whole, without considering its partition into stabilizer cosets. Graphically, this entails being limited to choosing the error with the highest symbol-wise probability (the error with the lowest weight) within the effective centralizer coset, which in some cases will not belong to the most likely coset. This issue can be resolved by using a decoder that operates based on the rule shown in (4.27), which is capable of considering the partition of the effective centralizer cosets into cosets of the stabilizer and finding the most likely stabilizer coset. Alas, no decoder for QLDPC codes that efficiently implements the Degenerate QMLD decoding rule of (4.27) [56, 70] exists at the time of writing.

It is important to mention that the differences between (4.25) and (4.27) are only relevant when the elements of the stabilizer group have low weight, i.e., when they have a small amount of non-identity operators [131]. Because low weight stabilizer elements are quite common in QLDPC codes, developing implementations of the optimal decoder is germane to the field. However, understanding why the differences between (4.25) and (4.27) are negligible when there are no low weight stabilizer elements also merits discussion.

Assume that we have a stabilizer code whose stabilizer elements have sufficiently large weight. Then, there will be an error operator in each effective centralizer coset (colored shaded rectangles in Figure 4.5), specifically one belonging to the stabilizer coset indexed by $\mathbf{L}_1 = I^{\otimes N}$, with a much smaller weight than the rest of the error operators contained in the coset. This will occur regardless of the partition of the effective centralizer coset into stabilizer cosets [131]. This low weight error will dominate the probability of the entire effective centralizer coset, and will force the probability of the stabilizer coset $\mathbf{T}_i \star \mathbf{L}_1 \star \overline{\mathcal{S}}$ to be higher than that of any other stabilizer coset. Therefore, decoding based on the QMLD classical rule (4.25) will lead to the lowest weight error operator $\hat{\mathbf{E}} \in \mathbf{T}_i \star \mathbf{L}_1 \star \overline{\mathcal{S}}$, while decoding based on (4.27) will produce a solution from the coset $\mathbf{T}_i \star \mathbf{L}_1 \star \overline{\mathcal{S}}$. As is shown in the next section, due to degeneracy any element of the coset

$\mathbf{T}_i \star \mathbf{L}_1 \star \overline{\mathcal{S}}$ will perform the same recovery operation, leading to the same result. Thus, in this case the performance of a decoder based on the classical rule (QMLD) will be identical to that of optimal quantum decoding (DQMLD). On the other hand, when the code has low weight stabilizer elements, there are many low weight error operators that may contribute to the probability of the different stabilizer cosets in the effective centralizer coset. Thus, the most likely error operator obtained by the classical rule (4.25) may not belong to the most likely stabilizer coset, which is the solution obtained by the optimal quantum decoder (4.27), and this would mean that the performance of a decoder based on the classical rule would be worse than that of optimal quantum decoding. We illustrate this with an example in section 4.3.3.

4.3 DEGENERACY AND WHY IT ARISES

In earlier sections of this chapter, we mentioned that due to degeneracy the stabilizer component of the error can be ignored in the decoding process, i.e, errors that differ only by a stabilizer element need not be distinguished by the decoder. Let us study why this happens.

For this purpose we introduce a similar example to the one given in [55]. Consider a scenario where the quantum state $|\psi\rangle$ has been encoded using a stabilizer code into the quantum codeword $|\overline{\psi}\rangle$ and transmitted through a quantum channel ξ. At the channel output, the noisy quantum state $|\overline{\psi}\rangle_\xi = \mathbf{E}' |\overline{\psi}\rangle$ is obtained[5], where $\mathbf{E}' = \mathbf{E} \star \mathbf{S}_v$, i.e., the error operator $\mathbf{E}'$ that has corrupted the quantum codeword can be expressed as the operation of another error operator $\mathbf{E}$ and a stabilizer element $\mathbf{S}_i$. Since by definition[6] of a stabilizer code $\mathbf{E}' |\overline{\psi}\rangle = \mathbf{E} \star \mathbf{S}_v |\overline{\psi}\rangle = \mathbf{E} |\overline{\psi}\rangle$, the measurable effects of error operator $\mathbf{E}'$ on the codeword will be the same as those of $\mathbf{E}$. In consequence, if we were to decompose $\mathbf{E}'$ and $\mathbf{E}$ based on (4.26), we would see that they would differ only in terms of their stabilizer component $\mathbf{S}_v$, i.e., their logical and pure error components would be identical.

From a graphical perspective, this scenario can be understood as $\mathbf{E}$ and $\mathbf{E}'$ belonging to the same stabilizer coset (gray bordered rectangles within

[5]Recall that the notation $\mathbf{E}' |\overline{\psi}\rangle \in \mathcal{H}_2^{\otimes N}$ denotes the evolution of state $|\overline{\psi}\rangle$ under the action of the error operator $\mathbf{E}'$.

[6]All elements of the stabilizer commute amongst themselves and have the same effect on any given quantum codeword [75, 131], i.e, they leave the state unchanged (see section 4.2).

colored shaded rectangles) in Figure 4.5. Based on this depiction, in order to accurately distinguish both errors, a decoder capable of differentiating operators contained within the same stabilizer coset would be required. Although the task of finding the correct stabilizer component of an error remains out of the reach of current quantum decoders, when focusing on the recovery of the transmitted quantum codeword, devising a decoder that can solve this task is unnecessary: the same reason that leads to the inability of current decoders to estimate the stabilizer component of the errors also makes the endeavor itself pointless. To see this, consider a recovery operator $\mathbf{R}$ that nullifies the effects of $\mathbf{E}$ on the transmitted quantum codeword, i.e., $\mathbf{R} \star \mathbf{E} \left|\overline{\psi}\right\rangle = \left|\overline{\psi}\right\rangle$. Then, this same recovery operator reverts the impact of $\mathbf{E}'$, since

$$\mathbf{R} \star \mathbf{E}' \left|\overline{\psi}\right\rangle = \mathbf{R} \star \mathbf{E} \star \mathbf{S}_i \left|\overline{\psi}\right\rangle = \mathbf{R} \star \mathbf{E} \left|\overline{\psi}\right\rangle = \left|\overline{\psi}\right\rangle . \tag{4.28}$$

Essentially, this shows that errors that differ only in the stabilizer component (errors that belong to the same stabilizer coset) can all be corrected using the same recovery operator. Hence, the decoder does not need to distinguish these errors to begin with. Similarly, all operators belonging to a specific stabilizer coset would recover the same quantum state, and any of these operators would be able to correct any error belonging to the coset. As a result, degeneracy should theoretically improve the performance of quantum codes, as it enables many errors to be corrected based on the same recovery operation. In particular instances, the positive effects that degeneracy can have on performance have already been observed [53, 132, 133].

4.3.1 DEGENERACY IN SPARSE QUANTUM CODES

Previously we mentioned that the performance improvements provided by the optimal quantum decoder of (4.27) over the classical decoder of (4.25) may become more notable when a large amount of stabilizer elements have low weight. When this happens, it is possible for many low weight operators to be spread out across the stabilizer cosets of the code. In such circumstances, properly summing the probabilities over these cosets is paramount to perform optimal quantum decoding following (4.27). Such is the case for many quantum LDPC codes, whose sparse nature typically results in them having a large amount of low weight stabilizer generators [55, 56, 70, 98, 131, 134]. The number of low weight stabilizer operators is higher for sparser codes, and given that the sparsity of these codes is their

primary appeal[7], QLDPC codes will generally have a large number of low weight stabilizer elements.

Should we be equipped with an optimal QLDPC decoder, we would be able to exploit the high degree of degeneracy of sparse quantum codes to improve performance, as typical (low weight) errors are likely to correspond to degenerate error operators [55]. In fact, it is likely that such a decoding strategy would result in a significant leap forward in terms of the performance of QLDPC codes. Unfortunately, implementation of a DQMLD decoder for sparse quantum codes remains an open research problem, so most QLDPC decoding schemes are based on the classical SPA decoder of (4.25). Given that in some cases the presence of low-weight stabilizer elements may result in the most likely error pattern not being contained in the most likely coset, decoding QLDPC codes based on the QMLD classical rule (4.25), rather than the DQMLD rule (4.27), can worsen their performance. This idea can be better understood by introducing the concept of end-to-end errors.

4.3.2 END-TO-END ERRORS

As was shown in (4.26), any error operator can be understood as the combination of a pure error, a logical operator and a stabilizer element, i.e, $\mathbf{E} = \mathbf{T} \star \mathbf{L} \star \mathbf{S}$. After decoding is performed, the estimated error operator, $\hat{\mathbf{E}} = \hat{\mathbf{T}} \star \hat{\mathbf{L}} \star \hat{\mathbf{S}}$, may be the same as the error operator introduced by the Pauli channel or it may be different. In the former case, the transmitted codeword will be recovered perfectly. In the latter however, i.e, when $\hat{\mathbf{E}} \neq \mathbf{E}$, we can have different situations:

1. **End-to-end degenerate errors:** These events take place when the estimated error pattern and the channel error both belong to the same stabilizer coset but they do not match, i.e., $\mathbf{T}_i = \hat{\mathbf{T}}_i$ and $\mathbf{L}_j = \hat{\mathbf{L}}_j$ but $\hat{\mathbf{E}} \neq \mathbf{E}$. However, because stabilizer elements have no effect on the transmitted quantum codeword, this difference is irrelevant and the encoded state will be recovered perfectly. In Figure 4.3, this situation occurs when $\mathbf{E}$ and $\hat{\mathbf{E}}$ belong to the same stabilizer coset $(\mathbf{T}_i \star \mathbf{L}_j) \star \overline{\mathcal{S}}$, where $i = 1, \ldots, 2^{N-k}$ and $j = 1, \ldots, 2^{2k}$.

[7] Typically, sparser codes require less quantum gates. This is beneficial because quantum gates are faulty elements that can introduce additional errors in the recovery process.

2. **End-to-end identical syndrome errors:** These events take place when the estimated error sequence and the channel error both belong to the same centralizer coset but each of them belongs to a different stabilizer coset, i.e., $\mathbf{T}_i = \hat{\mathbf{T}}_i$ and $\mathbf{L}_j \neq \hat{\mathbf{L}}_j$ (the channel logical operator and estimated logical operator do not match). This means that $\hat{\mathbf{E}}$ and $\mathbf{E}$ exhibit identical commutation relations with the stabilizer generators, and so they have the same syndrome $\mathbf{w}$. However, because they differ in their logical operator components, they will each act on the transmitted codeword in a distinct non-trivial manner. In consequence, the recovered quantum codeword will be different from the transmitted one. In Figure 4.3, this situation occurs when $\hat{\mathbf{E}}$ and $\mathbf{E}$ belong to the effective centralizer coset $\mathbf{T}_i \star \overline{\mathcal{Z}}(\overline{\mathcal{S}})$ but not to the same stabilizer coset. The impact of these errors, which are impossible to avoid in noisy channels, may be alleviated by employing the as of yet nonexistent optimal quantum decoder given in (4.27) rather than the classical decoder in (4.25). The classical counterpart of this situation would be the case in which the error introduced by the classical channel belongs to the codespace and so the transmitted codeword is mapped onto another codeword, which results in the information being corrupted in an undetectable manner. It should be noted that some literature refers to these end-to-end identical syndrome error events as logical errors.

3. **End-to-end errors with different syndromes:** These events occur when the error sequence estimated by the decoder and the real error sequence belong to different centralizer cosets, i.e., $\mathbf{T}_i \neq \hat{\mathbf{T}}_i$. Thus, they take place because the syndrome of $\hat{\mathbf{E}}$ is different from the syndrome of $\mathbf{E}$. Equipped with a perfect decoder, such scenarios would not exist, as the syndrome of the estimated error pattern $\hat{\mathbf{E}}$ would always match the measured syndrome associated to the channel error $\mathbf{E}$. However, because the decoding algorithm for sparse quantum codes is not ideal for the task, these errors can take place with varying probability [55, 98, 135].

The existence of end-to-end errors with different syndromes merits further discussion. As mentioned previously in section 3.2.2.1, the SPA decoder assumes that the individual marginal probabilities it computes are exact. However, this only holds whenever the factor graph is a tree, which, given the well-documented notoriety of LDPC and QLDPC codes for having short cycles, results in this requirement almost never being met

[20, 136, 137, 138]. In the classical paradigm, as a result of the negative impact of short-cycles on SPA-based decoding methods, a large body of research exists on how to minimize the presence of short-cycles in LDPC factor graphs [139, 140, 141, 142, 143]. This work becomes even more relevant for QEC, since the presence of short cycles of length-4 is essentially unavoidable in the construction of QLDPC codes. This occurs because of the commutativity constraint of stabilizer codes (recall that all the elements of the stabilizer must commute), which results in the QPCM of the code having an even number of row overlaps and in the appearance of length-4 cycles in the corresponding factor graph [20]. The negative effects of these cycles can be mitigated by using specific QLDPC construction strategies like the bicycle codes of [18], LDGM based CSS and non-CSS codes [21, 22, 23], or the quasi-cyclic constructions of [144, 145, 146, 147]. Recently, some results have shown that there is some merit in preserving a number of these cycles [148], as they can help with spreading information throughout the factor graph during the decoding process. This last result is further reaffirmed by the work of [55], where the methods employed to correct the so-called *symmetric degeneracy errors* benefit from the presence of short cycles.

Symmetric degeneracy errors [20, 55, 134, 149] are a particular type of end-to-end error with different syndrome that are caused by specific flaws in the structure of the factor graphs of sparse quantum codes (short cycles for instance). The term was coined in the work of Poulin et al. [55], which introduces a 2-qubit example where the SPA decoder produces a "symmetric" estimate of the channel error sequence, even though neither the true channel error nor the operators in its equivalence class exhibit this symmetry (hence a decoding mistake is made). The reader is referred to [134, 150] for a rigourous discussion on the symmetric degeneracy error phenomenon.

All in all, in order for QLDPC codes to realize their full potential, means of minimizing the frequency with which end-to-end errors with identical syndromes and end-to-end errors with different syndromes occur are necessary. To do so, we must first have the capacity to distinguish between different types of end-to-end errors, which, as will be shown in the next chapter, is a far from trivial task.

4.3.3 A USEFUL EXAMPLE

We close this chapter by providing a practical example to illustrate and facilitate the understanding of many of the concepts that have been presented thus far. The exercise is designed with the goal of helping the reader comprehend the phenomenon of degeneracy and the coset partitions of the effective N-fold Pauli group.

For the purpose of simplicity, we will be working with a very reduced number of qubits. Consider a scenario in which we wish to encode a single logical qubit into three physical qubits ($k = 1$ and $N = 3$). To build this system, we define a 3-qubit stabilizer code with generators

$$\begin{aligned} \mathbf{S}_1 &= XYZ \\ \mathbf{S}_2 &= YXI, \end{aligned} \tag{4.29}$$

whose encoding circuit, which requires the use of two ancilla qubits[8], is shown in Figure 4.6. The Phase (P), Hadamard (H), and CNOT gates that make up this encoding circuit are defined in Chapter 3 (see section 3.1.4.1).

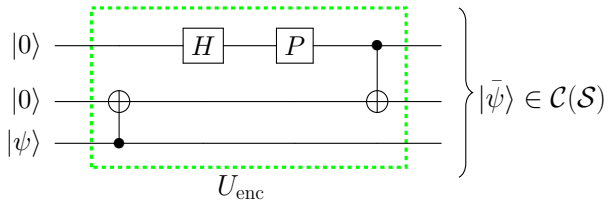


Figure 4.6: *Encoding circuit for the stabilizer code with the generators shown in (4.29). U_{enc} represents the unitary that performs the encoding operation, which, with the aid of the ancilla qubits $|0\rangle^{\otimes 2}$ takes the logical qubit $|\psi\rangle \in \mathcal{H}_2^{\otimes 1}$ to the codespace $|\overline{\psi}\rangle \in \mathcal{C}(\overline{\mathcal{S}}) \subset \mathcal{H}_2^{\otimes 3}$.*

If the encoded quantum state, codeword $|\overline{\psi}\rangle$, is transmitted through an arbitrary quantum channel ξ, a corrupted quantum state $|\overline{\psi}\rangle_\xi$ will be

[8] In the context of encoding operations, ancilla qubits are the extra qubits that are used to encode logical qubits into physical qubits.

obtained at the channel output. The error syndrome associated to this corrupted quantum state will be an $N-k$ binary vector that represents the commutation status of the stabilizer generators and the Pauli error introduced by the channel[9][97]. Figure 4.7 shows the coset partition of $\overline{\mathcal{G}}_N$ into cosets of the effective centralizer of the code $\overline{\mathcal{Z}}(\overline{\mathcal{S}})$. Since for this example we have $k=1$ and $N=3$, the effective 3-qubit Pauli space $\overline{\mathcal{G}}_3$ will be divided into $2^{(3-1)}$ cosets of $\overline{\mathcal{Z}}(\overline{\mathcal{S}})$. In Figure 4.7, the effective centralizer cosets are depicted in different colours. Moreover, each coset is headlined by its representative $\mathbf{T}_i$ as well as being associated to its respective syndrome $\mathbf{w}_i$, where $i=1,2,3,4$.

Let us now consider the centralizer coset $\mathbf{T}_1 \star \overline{\mathcal{Z}}(\overline{\mathcal{S}})$. Since $\mathbf{T}_1 = \beta^{-1}(\mathbf{t}_1)$ and $\mathbf{t}_1 = \mathbf{0}$, it is easy to see that the coset $\mathbf{T}_1 \star \overline{\mathcal{Z}}(\overline{\mathcal{S}})$ is actually the effective centralizer itself, since $\mathbf{T}_1 = \beta^{-1}(\mathbf{0}) = I^{\otimes 3} = III$. The effective centralizer of a stabilizer code is comprised of 2^{2k} stabilizer cosets. In our example, given that $k=1$, each effective centralizer coset is divided into 4 stabilizer cosets. This is represented in Figure 4.7 by the subdivision of the distinctly coloured rectangles (which represent the cosets of $\overline{\mathcal{Z}}(\overline{\mathcal{S}})$) into even smaller rectangles (which represent the stabilizer cosets). Each of these rectangles is indexed by the appropriate representative $\mathbf{L}_j$, where $j=1,2,3,4$. These representatives embody non-trivial operations on the logical qubit. More explicitly, $\mathbf{L}_1$ represents the action of the I Pauli gate on the logical qubit, $\mathbf{L}_2$ represents the action of the X Pauli gate on the logical qubit, i.e., $U_{\text{enc}}(|0\rangle^{\otimes 2} X|\psi\rangle) = \mathbf{L}_2|\bar{\psi}\rangle$, $\mathbf{L}_3$ represents the action of the Y Pauli gate on the logical qubit, and $\mathbf{L}_4$ represents the action of the Z Pauli gate on the logical qubit. We can also relate these logical operators to the concept of encoded Pauli operators. For this purpose, recall that the $2k$ encoded Pauli operators form a generating set for the logical group, i.e, all the operators $\{\mathbf{L}_j\}_{j=1}^4$ can be generated from $\overline{\mathbf{X}}_1$ and $\overline{\mathbf{Z}}_1$. Knowing that the Pauli matrices are related as $Y = iXZ$ (see section 3.1.4.1), then it is clear that in this example[10], $\overline{\mathbf{X}}_1 = \mathbf{L}_2$ and $\overline{\mathbf{Z}}_1 = \mathbf{L}_4$.

All the error operators of a specific stabilizer coset $\mathbf{T}_i \star \mathbf{L}_j \star \overline{\mathcal{S}}$ (the degenerate Pauli operators associated to the logical operator $\mathbf{L}_j$ and pure error $\mathbf{T}_i$) have the same effect on the transmitted codeword and they will be

[9]Recall that, up to an overall phase, assuming that errors induced by a Pauli channel belong to $\overline{\mathcal{G}}_N$ is identical to assuming that they belong to $\mathcal{G}_N$.

[10]In more complex scenarios the encoded Pauli operators of a stabilizer code are found by computing the standard form of the QPCM of the code. This is discussed in the next chapter.

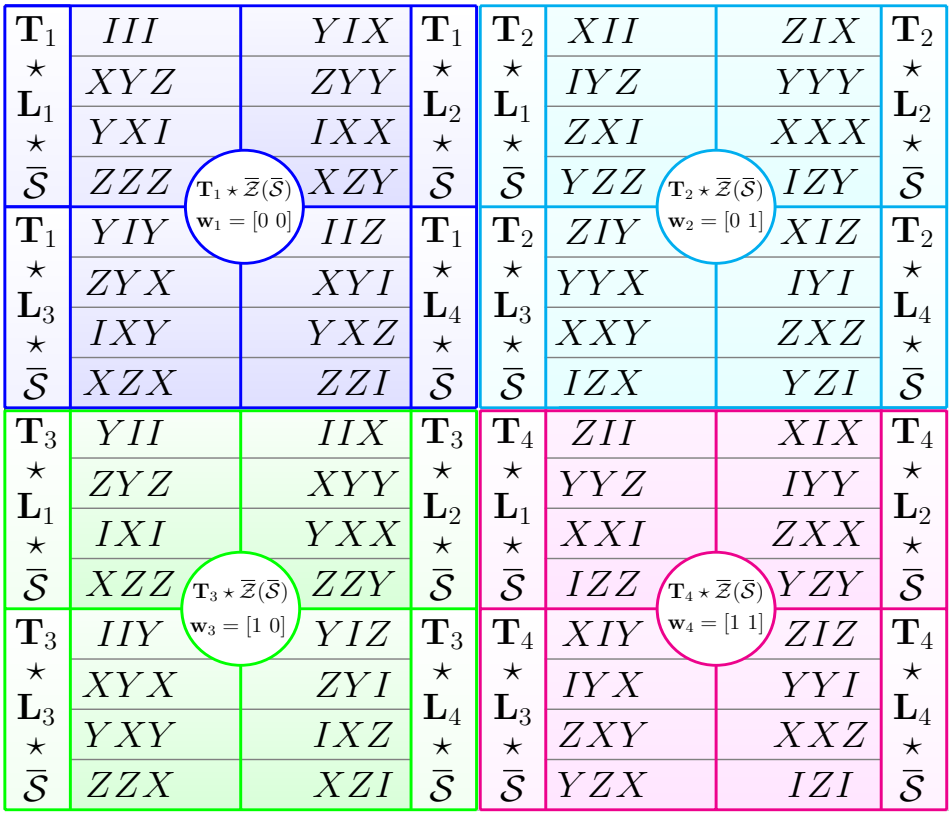


Figure 4.7: *Partition of $\overline{\mathcal{G}}_3$ into cosets when the 3-qubit stabilizer code with generators $\{XYZ, YXI\}$ is used. The first coset partition divides $\overline{\mathcal{G}}_3$ into $2^{N-k} = 4$ cosets of the effective centralizer, which are distinguished by different colours, indexed by their corresponding pure error representative $\mathbf{T}_i$ and denoted as $\mathbf{T}_i \star \overline{\mathcal{Z}}(\overline{\mathcal{S}})$, where $i = 1 \ldots 4$. Each of these cosets is also associated to its corresponding syndrome $\mathbf{w}_i$. The second coset partition is represented by the subdivision of each coset $\mathbf{T}_i \star \overline{\mathcal{Z}}(\overline{\mathcal{S}})$ into the $2^{2k} = 4$ cosets associated to their corresponding logical operator representatives $\mathbf{L}_j$, where $j = 1 \ldots 4$. This partition divides every effective centralizer coset into cosets of the stabilizer. Within each stabilizer coset, the degenerate errors associated to each logical operator can be found.*

reversible via the same recovery operator. Notice that the coset associated to $\mathbf{L}_1$ within $\overline{\mathcal{Z}}(\overline{\mathcal{S}})$ is actually the stabilizer of the code. As shown in Figure 4.7, the stabilizer is made up of the 3-qubit identity operator III, the generators XYZ and YXI, and the product of the generators ZZZ.

Table 4.1 shows the logical operators $\{\mathbf{T}_1 \star \mathbf{L}_j = \mathbf{L}_j\}_{j=1}^{4}$ that serve as the coset representatives of the stabilizer cosets found within the first centralizer coset, $\mathbf{T}_1 \star \overline{\mathcal{Z}}(\overline{\mathcal{S}}) = \overline{\mathcal{Z}}(\overline{\mathcal{S}})$. It also includes the pure errors $\{\mathbf{T}_i\}_{i=1}^{4}$ that serve as the coset representatives of the cosets of $\overline{\mathcal{Z}}(\overline{\mathcal{S}})$. Based on the information provided by this table, along with the stabilizer generators, we can generate the entire partition $\overline{\mathcal{G}}_3/\overline{\mathcal{Z}}(\overline{\mathcal{S}})/\overline{\mathcal{S}}$. First, we obtain the set $\{\mathbf{S}_v\}_{v=0}^{3}$ of elements that make up $\overline{\mathcal{S}}$ from the stabilizer generators $\mathbf{S}_1 = XYZ$ and $\mathbf{S}_2 = YXI$, where $\mathbf{S}_0$ is the trivial generator III. Following this, we can obtain the set of operators that define the effective centralizer $\overline{\mathcal{Z}}(\overline{\mathcal{S}})$ by computing the products $\mathbf{L}_j \star \mathbf{S}_v$, where $j = 1, \ldots 4$. Lastly, we can obtain the rest of the 3-qubit effective Pauli space $\overline{\mathcal{G}}_3$ by computing the products $(\mathbf{T}_i \star \mathbf{L}_j) \star \mathbf{S}_v$, where $i = 1, \ldots, 4$, $j = 1, \ldots, 4$.

Table 4.1: *Coset representatives* $\{\mathbf{T}_i\}_{i=1}^{4}$ *and* $\{\mathbf{L}_j\}_{j=1}^{4}$.

$\mathbf{L}_1$	III	$\mathbf{T}_1$	III
$\mathbf{L}_2$	YIX	$\mathbf{T}_2$	XII
$\mathbf{L}_3$	YIY	$\mathbf{T}_3$	YII
$\mathbf{L}_4$	IIZ	$\mathbf{T}_4$	ZII

Table 4.1 also serves to provide practical examples of the decomposition given in (4.26) for operators in $\overline{\mathcal{G}}_N$. For instance, consider the operator $\mathbf{A}_1 = XYX$ which is the second element of the coset $(\mathbf{T}_3 \star \mathbf{L}_3) \star \overline{\mathcal{S}}$ (left-most bottom green rectangle in Figure 4.7). Based on the aforementioned decomposition, we should be able to write this operator as the $\star$ product of its pure, logical and stabilizer components. This can be easily obtained as

$$\mathbf{A}_1 = XYX = YII \star YIY \star XYZ \ ,$$

where $\mathbf{T}(\mathbf{A}_1) = YII$, $\mathbf{L}(\mathbf{A}_1) = YIY$, and $\mathbf{S}(\mathbf{A}_1) = XYZ$, denote the pure error, logical and stabilizer components, respectively.

Most importantly, Figure 4.7 can be used to illustrate the effects of degeneracy and the impact of employing suboptimal methods to decode QLDPC codes. Let us assume that we use our 3-qubit stabilizer code to

transmit quantum codeword $|\overline{\psi}\rangle$ over a depolarizing channel with depolarizing probability $p = 0.05$, and that we obtain the noisy quantum state $|\overline{\psi}\rangle_{\xi_P}$ at the channel output. Based on the probability distribution of the depolarizing channel shown in section 3.1.4.5, we can compute the probability of every possible 3-qubit error operator shown in Figure 4.7 that may have acted on the transmitted quantum codeword. These probabilities are shown in Figure 4.8a, which also shows the operators that the traditional SPA decoder of (4.25) would estimate out of each of the effective centralizer cosets. For instance, if the syndrome extracted from $|\overline{\psi}\rangle_{\xi_P}$ is $\mathbf{w}_1 = [0\ 0]$, decoding based on (4.25) would yield $\hat{\mathbf{E}} = III$, which is the most likely error operator in the coset $\mathbf{T}_1 \star \overline{\mathcal{Z}}(\overline{\mathcal{S}})$. In this case, having a decoder with the ability to consider the partition into stabilizer cosets would not yield any gain, because the probability of the coset indexed by $\mathbf{T}_1 \star \mathbf{L}_1$ is vastly greater than the probability of the remaining stabilizer cosets. This is shown in Figure 4.8b, where the probabilities of the stabilizer cosets (the sum over each of the stabilizer equivalence classes) are depicted. Thus, for this scenario, a decoder based on the classical SPA algorithm and a degenerate decoder would attain the same results.

Let us now consider the unlikely[11] outcome that performing measurements on $|\overline{\psi}\rangle_{\xi_P}$ yields the syndrome $\mathbf{w}_4 = [1\ 1]$. Successful decoding based on the traditional SPA decoder would yield the lowest weight operator from the coset $\mathbf{T}_4 \star \overline{\mathcal{Z}}(\overline{\mathcal{S}})$ which is either the error operator ZII belonging to coset $\mathbf{T}_4 \star \mathbf{L}_1 \star \overline{\mathcal{S}}$ or the error operator IZI belonging to coset $\mathbf{T}_4 \star \mathbf{L}_4 \star \overline{\mathcal{S}}$. Notice, as shown in Figure 4.8b, that the probability of the coset $\mathbf{T}_4 \star \mathbf{L}_1 \star \overline{\mathcal{S}}$ is identical to that of the coset $\mathbf{T}_4 \star \mathbf{L}_4 \star \overline{\mathcal{S}}$. As in the previous case, a degenerate decoder capable of considering the partition of the effective centralizer cosets into stabilizer cosets would not outperform the traditional SPA decoder in this scenario. Note that, for both of the previous examples, this happens because the error sequence with highest probability is in the stabilizer coset with highest probability. As shown in (4.28), regardless of whether the estimated error sequence and the channel error match, the traditional SPA decoder will correctly solve the decoding problem if the estimated error sequence is in the same stabilizer coset as the channel error.

To provide an example in which the degenerate decoder clearly outperforms the traditional SPA decoder, let us now consider transmission of encoded quantum states over a Pauli channel with $p_x = 0.26$, $p_y = 0.28$,

[11] The likelihood of such an outcome will increase as the depolarizing probability of the channel increases.

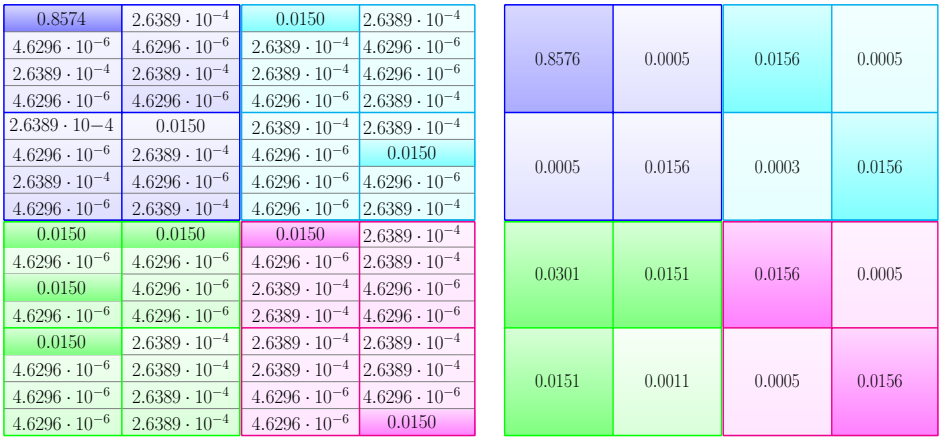


Figure 4.8: ***(a)*** *Probability of the Pauli operators shown in Figure 4.7 when a depolarizing channel with parameter $p = 0.05$ is considered. The darker shaded rectangles denote the operators from each centralizer coset that could be obtained by the classical SPA decoder of (4.25).* ***(b)*** *Probability of each stabilizer coset considered in the partition $\overline{\mathcal{G}}_3$ shown in Figure 4.7 when a depolarizing channel with parameter $p = 0.05$ is considered. The darker shaded rectangles denote the stabilizer cosets from which the classical SPA decoder of (4.25) can produce solutions. Notice that for syndrome $\mathbf{w}_1$, the result produced by the classical SPA decoder belongs to the stabilizer coset that has the highest probability, and thus classical decoding produces the same result as optimal quantum (degenerate) decoding.*

and $p_z = 0.15$. Figures 4.9a and 4.9b portray the individual probabilities of each of the operators shown in Figure 4.7 and the probabilities of each stabilizer coset, respectively, after transmitting over the Pauli channel with $p_x = 0.26$, $p_y = 0.28$, and $p_z = 0.15$. In this case, there are multiple scenarios in which the optimal (degenerate) QLDPC decoder would outperform the traditional SPA decoder. Let us discuss some of them.

Assume, as previously, that we have transmitted a quantum codeword over this channel and that an error $\mathbf{E}_2 \in \mathbf{T}_1 \star \overline{\mathcal{Z}}(\overline{\mathcal{S}})$ has been introduced by the Pauli channel, which results in the syndrome measurement yielding $\mathbf{w}_1 = [0\ 0]$. A traditional SPA decoder would be capable of determining that $\mathbf{T}(\mathbf{E}_2) = \mathbf{T}_1 = III$, and would choose the error operator contained in the effective centralizer coset $\mathbf{T}_1 \star \overline{\mathcal{Z}}(\overline{\mathcal{S}})$ that has the highest probability of occurring, i.e., $\hat{\mathbf{E}}_2^{\text{SPA}} = III$ (see Figure 4.9a). However, Figure 4.9b shows that the probability of the stabilizer coset $(\mathbf{T}_1 \star \mathbf{L}_1) \star \overline{\mathcal{S}}$, which contains the most likely operator, $\hat{\mathbf{E}}_2^{\text{SPA}} = III$, is lower than the probability of the stabilizer coset $(\mathbf{T}_1 \star \mathbf{L}_3) \star \overline{\mathcal{S}}$. This means that producing an estimate $\hat{\mathbf{E}}_2 \in (\mathbf{T}_1 \star \mathbf{L}_3) \star \overline{\mathcal{S}}$ will result in better performance than choosing the most likely error operator in $\mathbf{T}_1 \star \overline{\mathcal{Z}}(\overline{\mathcal{S}})$. Such an estimate can only be produced by the decoder given in (4.27), which is capable of considering the partition of the effective centralizer into cosets of the stabilizer. Therefore, in this instance, the degenerate decoder would outperform the traditional SPA decoder. This would also occur if an error $\mathbf{E}_3 \in \mathbf{T}_2 \star \overline{\mathcal{Z}}(\overline{\mathcal{S}})$ were to take place. By comparing Figures 4.9a and 4.9b, we can see how, in this scenario, the most likely error operator does not belong to the most likely stabilizer coset.

In the case that $\mathbf{E} \in \mathbf{T}_3 \star \overline{\mathcal{Z}}(\overline{\mathcal{S}})$ or $\mathbf{E} \in \mathbf{T}_4 \star \overline{\mathcal{Z}}(\overline{\mathcal{S}})$, there are multiple error operators that have the highest probability to occur, but only one of them belongs to the most likely stabilizer coset. This implies that, although performance of the traditional SPA decoder would be better than in the two previous cases, it would sometimes produce suboptimal estimates that would not belong to the most likely stabilizer coset. Therefore, in instances such as this one, the optimal degenerate decoder would also outperform traditional SPA decoding methods.

Lastly, it is important to mention that when a traditional SPA-based decoder is used to estimate operators from $\overline{\mathcal{G}}_N$ and it fails to find the correct logical component of the error, this goes unnoticed. Recall that although the effect of logical operators on the codeword is non-trivial, their actions

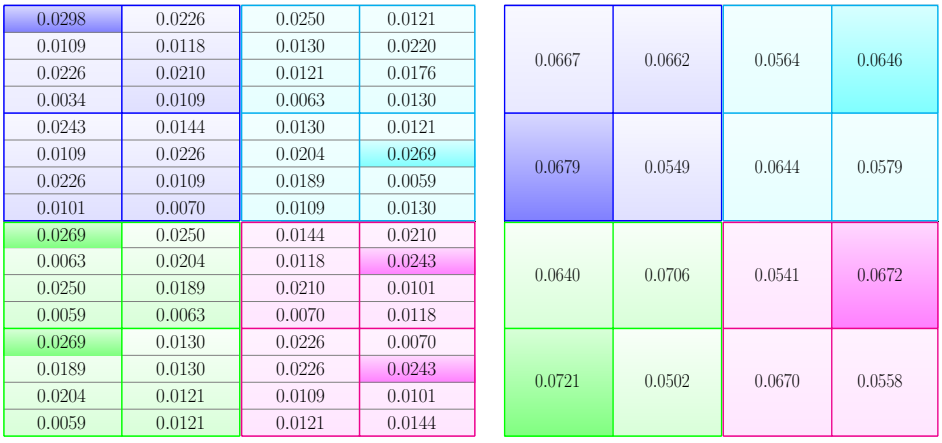


Figure 4.9: ***(a)*** *(a) Probability of the Pauli operators that are shown in Figure 4.7 when a Pauli channel with parameters $p_x = 0.26$, $p_y = 0.28$, and $p_z = 0.15$ is considered. The darker shaded rectangles denote the error operators from each centralizer coset that could be obtained by the classical SPA decoder of (4.25).* ***(b)*** *Probability of each stabilizer coset considered in the partition $\bar{\mathcal{G}}_3$ shown in Figure 4.7 when a Pauli channel with parameters $p_x = 0.26$, $p_y = 0.28$, and $p_z = 0.15$ is considered. The darker shaded rectangles denote the stabilizer cosets with the highest probability within each effective centralizer coset. Notice that for syndrome $\mathbf{w}_1$, the result produced by the classical SPA decoder does not belong to the stabilizer coset that has the highest probability, and thus classical decoding underperforms when compared to optimal quantum (degenerate) decoding.*

preserve the codespace. Based on this fact, it is easy to see how these undetected decoding errors can quickly become a significant nuisance when using stabilizer codes with a large number of stabilizer elements. As mentioned earlier, this is a common occurrence when operating with QLDPC codes that utilize a large number of qubits, as they may sometimes have many low weight operators spread out across their stabilizer cosets. In such scenarios, hefty performance prices may potentially be paid in consequence of employing a suboptimal decoder based on traditional SPA decoding.

4.4 CHAPTER SUMMARY

The quantum phenomenon known as degeneracy should theoretically improve the performance of quantum codes. Unfortunately, degeneracy can end up sabotaging the performance of sparse quantum codes because its existence is neglected in the decoding process. In this chapter we have provided a broad overview of the role this phenomenon plays in the realm of quantum error correction and, more specifically, in the field of QLDPC codes. We began by introducing a group theoretical explanation of the most relevant concepts in the field of quantum error correction. Following this, we examined the differences between the classical decoding problem and its quantum counterpart. Despite the intricate similarities between classical and quantum decoders, the coset partition of the N-fold Pauli space in the context of stabilizer codes reveals the higher complexity of the quantum decoding task. Based on this discussion, we were later able to show why applying the classical decoding algorithm for sparse codes to the quantum problem is suboptimal. Then, we provided a detailed explanation regarding the origin of degeneracy and how it may have detrimental effects on traditional SPA-based decoders. We finished with an example that serves to facilitate the comprehension of the topics discussed throughout the chapter.

CHAPTER 5

Detecting Degeneracy and Improved Decoding Strategies

"Our need will be the real creator"

Plato.

In the previous chapter we analyzed the phenomenon of degeneracy through the lens of group theory. We introduced the coset structure of stabilizer codes and showed how degeneracy is responsible for the existence of error equivalence classes (stabilizer cosets) whose individual elements act identically on the transmitted information (encoded quantum states). Following this, we presented how, when using decoding based on the QMLD rule, different decoding scenarios can be encountered. We referred to these scenarios as end-to-end errors, and showed how they represent particular instances that must be distinguished in order to both assess and improve the performance of stabilizer codes. For example, it is necessary to differentiate between end-to-end degenerate errors and end-to-end identical syndrome errors, since the former type of end-to-end error has no impact on encoded quantum states but the latter one does (it acts non-trivially on the codespace). This is especially relevant when studying sparse quantum codes, whose degenerate nature will more than likely increase the frequency with which degenerate errors occur [16, 55, 56, 67, 68, 69, 70, 71, 98].

In this chapter, we employ the group theoretic framework presented in Chapter 4 to derive a strategy capable of accurately assessing the impact of degeneracy on the performance of sparse quantum codes. We also explain another method that has been used in [151, 152, 153, 154] to do so. Then, we use our methodology to show how sparse quantum codes should always be assessed using a metric known as the logical error rate. We back our claims up by using our strategy to analyze the frequency with which different types of errors occur when using sparse quantum codes. Finally, we provide insight on how the design and decoding of these codes can be improved, and we discuss the most relevant advancements that have appeared in the literature [18, 20, 55, 135, 149, 151, 152, 153, 155].

5.1 PERFORMANCE METRICS

Despite its theoretical performance benefits, limited research exists on how to quantify the true impact that degeneracy has on QLDPC codes. This has resulted in the performance of sparse quantum codes being assessed differently throughout the literature; while some research considers the effects of degeneracy by computing a metric known as the *logical error rate* [55, 135, 151, 152, 153, 154, 156], other works employ the classical strategy of computing the *physical error rate* [20, 21, 22, 23, 149, 157], a metric which provides an upper bound on the performance of these codes. Let us analyze the differences between the two.

5.1.1 PHYSICAL AND LOGICAL ERROR RATE

The manifestation of the degeneracy phenomenon in the realm of quantum error correction makes it impossible to accurately predict the performance of QEC codes with methods that disregard its presence. The more degenerate the code family that is being studied, the larger the discrepancy between results computed based on degeneracy-ignoring metrics and the true performance of these codes will be. Thus, in the paradigm of sparse quantum codes appropriate performance assessment is of paramount importance. As was mentioned in the previous chapter, this requires a method that can distinguish between the different types of end-to-end errors. For reference, the characteristics of these end-to-end errors are summarized in table 5.1 (see section 4.3.2 for a more complete discussion on end-to-end errors). Fortunately, a metric capable of telling different end-to-end errors apart already exists. It is known as the logical error rate and it has been

widely employed to assess the performance of various other families of QEC codes such as QTCs and quantum topological codes [68, 29, 72, 73, 74]. Generally, the decoders used in QTC or topological schemes are capable of producing estimates of the stabilizer coset representatives (the logical operators) of the channel errors, hence these decoders "inherently" compute the logical error rate. In contrast, the classical decoding algorithms [48, 47] used to decode QLDPC codes by solving the QMLD rule (4.25) are unable to produce estimates of stabilizer coset representatives, and so their logical error rate must be computed differently.

Table 5.1: *Characteristics of the different types of end-to-end errors that can arise when using stabilizer codes. The hat notation is used to represent estimations made by the decoder, i.e.,* $\hat{\mathbf{w}}$*,* $\hat{\mathbf{E}}$*,* $\hat{\mathbf{T}}_i$*, and* $\hat{\mathbf{L}}_j$ *represent the estimated syndrome, estimated error sequence, centralizer coset representative of the estimated error sequence and stabilizer coset representative of the estimated error sequence, respectively.*

Type of error	Defining Characteristics	Outcome (Physical)	Outcome (Logical)
End-to-end error with different syndrome	$\hat{\mathbf{w}} \neq \mathbf{w}$ $\hat{\mathbf{T}}_i \neq \mathbf{T}_i$ $\hat{\mathbf{E}} \neq \mathbf{E}$	Failure	Failure
End-to-end identical syndrome error	$\hat{\mathbf{w}} = \mathbf{w}$ $\hat{\mathbf{T}}_i = \mathbf{T}_i$ and $\hat{\mathbf{L}}_j \neq \mathbf{L}_j$ $\hat{\mathbf{E}} \neq \mathbf{E}$	Failure	Failure
End-to-end degenerate error	$\hat{\mathbf{w}} = \mathbf{w}$ $\hat{\mathbf{T}}_i = \mathbf{T}_i$ and $\hat{\mathbf{L}}_j = \mathbf{L}_j$ $\hat{\mathbf{E}} \neq \mathbf{E}$	Failure	Success

With our knowledge regarding the coset structure of sparse quantum codes, intuition would point towards calculating the logical error rate by finding and comparing the stabilizer cosets of the estimated error sequences and the stabilizer cosets of the channel errors. Unfortunately, the task of computing stabilizer cosets has been shown to be computationally hard [56, 70, 149], which is the reason why the performance of some sparse

quantum codes [21, 22, 20, 23, 157] and some improved decoding strategies [149] has been assessed based on the *physical error rate*. This metric is computed by comparing the error sequence estimated by the decoder, $\hat{\mathbf{E}} \in \overline{\mathcal{G}}_N$, to the channel error, $\mathbf{E} \in \overline{\mathcal{G}}_N$. Essentially, if the estimation matches the channel error, the decoder has been successful, and if not, a decoding failure has occurred. Note, however, that because this metric ignores the degenerate nature of stabilizer codes, the physical error rate overestimates the number of decoding failures and actually represents an upper bound on the performance of stabilizer codes.

Despite the use of the physical error rate in some works, other literature has successfully computed the logical error rate of specific sparse quantum codes [55, 135, 151, 152, 153, 154, 156]. These works succeed in computing the logical error rate because they do not approach the issue from the perspective of stabilizer cosets. Instead, in most of these works [55, 151, 152, 153, 154], they use Gaussian elimination to obtain the parity check matrix of the code in what is known as *standard form* [16, 158, 159], and then use it to extract a basis for the encoded Pauli operators of the corresponding codes. Then, they employ this basis to differentiate between different types of end-to-end errors. Additionally, other research [135] employs a different, albeit much more computationally demanding, method to compute the logical error rate. These strategies and concepts are explained in what follows.

Undetected classical errors

Before studying how the logical error rate can be computed, it should be mentioned that the concept of *undetected* errors is not exclusive to the quantum paradigm. In fact, even though degeneracy does not exist in the classical coding realm, *undetected* or *logical* errors in classical LDPC codes and classical turbo codes have previously been studied. This idea was introduced by the early work of MacKay et al. [112], where a classical undetected error is defined as a decoding estimate that is not equal to the original error sequence and that is produced when the decoder exits before the maximum number of decoding iterations (it produces a valid syndrome). More explicitly, a classical undetected error occurs when the estimate of the channel error is not equal to the real error but the estimated syndrome and the measured syndrome match. In their analysis of classical LDPC codes, MacKay et al. showed that all of the decoding mistakes they encountered were detected errors (classical undetected errors were only observed in turbo codes). This was also shown in [153] for a slightly differ-

ent decoding algorithm. Similar outcomes were observed in the quantum paradigm for the failed recoveries of the modified decoding strategies of [55, 149]. The techniques of [55, 149] serve to improve standard SPA decoding of quantum codes by post-processing the initial error estimates and producing new estimates of the channel error. If these new estimates do not revert the channel error, then they are referred to as failed recoveries or failed error corrections. In [55], all of the failed error corrections were shown to be end-to-end errors with different syndromes, whereas in [149] a small percentage of failed estimates were shown to be end-to-end identical syndrome errors and end-to-end degenerate errors[1].

Aside from these failed recovery analyses, the literature is limited when it comes to assessing the percentage of decoding failures that is caused by each type of end-to-end error. It is reasonable to believe that QLDPC codes, given their large number of degenerate operators [55, 56, 70, 98, 131, 134], will experience a large number of end-to-end degenerate errors. We confirm this intuition in the final section of this chapter, where we show how up to 30% of the end-to-end errors that take place when using the QLDPC codes of [21, 22, 23] are degenerate.

5.1.2 DISCRIMINATING BETWEEN DIFFERENT TYPES OF END-TO-END ERRORS

It is clear from the defining characteristics of each specific type of end-to-end error (see table 5.1) that end-to-end errors with different syndromes are the easiest type of error to identify. In fact, doing so is trivial, as all that is required is a comparison of the syndrome estimate, $\hat{\mathbf{w}} \in \mathbb{F}_2^{N-k}$, and the measured syndrome, $\mathbf{w} \in \mathbb{F}_2^{N-k}$, where k and N represent the number of logical qubits and physical qubits (block length) of the quantum code, respectively. Similarly, knowing that either an end-to-end identical syndrome error or an end-to-end degenerate error has occurred is simple. This can be done by comparing the estimate of the error sequence $\hat{\mathbf{E}}$ to the channel error $\mathbf{E}$ whenever $\hat{\mathbf{w}} = \mathbf{w}$, i.e., if $\hat{\mathbf{E}} \neq \mathbf{E}$ either an end-to-end degenerate error or an identical syndrome error has occurred, and if $\hat{\mathbf{E}} = \mathbf{E}$, no error has taken place. The issue arises when trying to distinguish between these two families of end-to-end errors. Notice that the comparison $\hat{\mathbf{E}} \neq \mathbf{E}$ does not reveal if the error estimate belongs to the same stabilizer

[1]The authors of this work do not distinguish between end-to-end identical syndrome errors and end-to-end degenerate errors.

coset as the channel error. Hence, we have no way of discriminating between end-to-end degenerate errors and end-to-end identical syndrome errors.

The first strategy that comes to mind to resolve this problem is to compute the stabilizer of the code in question, calculate the $\star$ product of the stabilizer with the channel error $\mathbf{E}$ to extract the specific stabilizer coset of the channel error, and then check if $\hat{\mathbf{E}}$ belongs to this coset. This works because the coset representative choice for the coset $(\mathbf{T}_i \star \mathbf{L}_j) \star \overline{\mathcal{S}}$ is irrelevant (any operator belonging to the coset serves as a valid representative). Thus, computing $\mathbf{E} \star \overline{\mathcal{S}}$ will yield the stabilizer coset of the channel error, i.e., $\mathbf{E} \star \overline{\mathcal{S}} = (\mathbf{T}_i \star \mathbf{L}_j) \star \overline{\mathcal{S}}$. Therefore, whenever $\hat{\mathbf{w}} = \mathbf{w}$ and $\hat{\mathbf{E}} \neq \mathbf{E}$, we will know that an end-to-end degenerate error has occurred if the estimated error sequence $\hat{\mathbf{E}}$ is in the coset $\mathbf{E} \star \overline{\mathcal{S}}$. If this does not occur, then an end-to-end identical syndrome error will have taken place.

Unfortunately, since extracting the stabilizer of a quantum code becomes increasingly complex as its block length increases, this strategy will only be applicable to short quantum codes. We know from Chapter 4 that the number of elements in the stabilizer of a quantum code with block length N and rate R_Q is given by $2^{N-k} = 2^{N(1-R_Q)}$, which grows exponentially with N and can rapidly become intractable on a classical machine as this parameter increases. In light of this, it is apparent that more practical methods to differentiate between end-to-end degenerate errors and end-to-end identical syndrome errors are necessary. Both the strategy we propose herein and that employed in [55, 153, 151, 152, 154], which are explained in the next two sections, resolve this issue.

5.2 AN ALGEBRAIC PERSPECTIVE ON END-TO-END DEGENERATE ERRORS

The problem of differentiating between end-to-end identical syndrome errors and end-to-end degenerate errors can also be formulated as a set of linear equations. We know from section 4.2.3.3 that the QPCM of a stabilizer code that encodes k logical qubits into N physical qubits (a rate $R_Q = \frac{k}{N}$ code with block length N) can be written as

$$\mathbf{H}_{\overline{\mathcal{S}}} = \begin{pmatrix} \mathbf{h}_1 \\ \mathbf{h}_2 \\ \vdots \\ \mathbf{h}_{N-k} \end{pmatrix}, \tag{5.1}$$

where $\mathbf{h}_v = \mathbf{s}_v$ denotes the symplectic representation of the generators $\{\mathbf{S}_v\}_{v=1}^{N-k} \in \overline{\mathcal{G}}_N$ that define the stabilizer group. Each of the elements of $\overline{\mathcal{S}}$ is a linear combination of the $N-k$ generators, hence, if $\mathbf{S}$ is an element of the stabilizer and $\mathbf{s}$ is the symplectic representation of this stabilizer element, then

$$\mathbf{s} = \left(\sum_{v=1}^{N-k} \mathrm{a}_v \mathbf{h}_v \right) \mod 2, \tag{5.2}$$

where $\mathbf{a} = (\mathrm{a}_1, \ldots, \mathrm{a}_{N-k})$ is a unique binary vector.

Whenever a channel error $\mathbf{E}$ takes place, the decoder will compute an estimate of this error and produce an estimate of the syndrome associated to it. As discussed previously, this syndrome only determines which specific effective centralizer coset the channel error belongs to. In other words, the syndrome diagnoses the effective centralizer coset representative $\mathbf{T}_i$ of the channel error. Assuming that an end-to-end error with different syndrome does not occur, the estimated syndrome will match the measured syndrome, hence the centralizer coset representative of the estimated error sequence and the centralizer coset representative of the channel error will also be the same, i.e., $\hat{\mathbf{T}}_i = \mathbf{T}_i$. Thus, if we compute the $\star$ operation of the channel error $\mathbf{E}$ and the estimated error $\hat{\mathbf{E}}$, which can also be understood as the mod2 sum of their symplectic representations over $\mathbb{F}_2^{2N}$: $\beta(\mathbf{E}) \oplus \beta(\hat{\mathbf{E}}) = \mathbf{e} \oplus \hat{\mathbf{e}}$, where β denotes the symplectic map, the sequence $\mathbf{E}$ will be shifted to the effective centralizer $\overline{\mathcal{Z}}(\overline{\mathcal{S}})$. Note that we can also write this using the symplectic map as $\mathbf{E} \star \hat{\mathbf{E}} \in \overline{\mathcal{Z}}(\overline{\mathcal{S}}) \rightarrow \beta(\mathbf{E}) \oplus \beta(\hat{\mathbf{E}}) \in \beta(\overline{\mathcal{Z}}(\overline{\mathcal{S}})) = \mathbf{e} \oplus \hat{\mathbf{e}} \in \Gamma_R$. Based on this, the issue of determining whether an end-to-end error is degenerate can be resolved by answering the following question:

$$\exists\, \mathbf{a} : \mathbf{e} \oplus \hat{\mathbf{e}} = \left(\sum_{i=1}^{N-k} \mathrm{a}_i \mathbf{h}_i \right) \mod 2 \ \ ? \tag{5.3}$$

Essentially, if a set of coefficients $\mathbf{a}$ exists such that the above equation holds, i.e., if $\mathbf{e} \oplus \hat{\mathbf{e}} \in \mathbb{F}_2^{2N}$ is a linear combination of the symplectic representation of the stabilizer generators, then $\mathbf{E} \star \hat{\mathbf{E}}$ belongs to the stabilizer and an end-to-end degenerate error will have occurred. If such a set of coefficients does not exist, then and end-to-end identical syndrome error has taken place.

The expression shown in (5.3) defines a linear system of Diophantine equations over the binary field. An answer to this question can be found by

writing the augmented matrix $[\mathbf{H}_{\mathcal{S}}^{\top}|(\mathbf{e}\oplus\hat{\mathbf{e}})^{\top}]$ in its row-echelon form. This means that, based on this procedure, it is possible to determine the type of end-to-end errors that occur and subsequently compute the logical error rate. This is done in [135]. However, although the procedure is conceptually simple, rewriting the augmented matrix in such a manner becomes increasingly computationally complex as matrices grow in size. Unfortunately, for sparse quantum codes to be good, the block length must be large, which implies that the PCMs of these codes will also be large[2]. Furthermore, the row-echelon form of the matrix $[\mathbf{H}_{\mathcal{S}}^{\top}|(\mathbf{e}\oplus\hat{\mathbf{e}})^{\top}]$ must be computed during every simulation iteration (whenever the estimated syndrome and the measured syndrome match) to determine what type of end-to-end error has taken place, which may significantly increase simulation time. For these reasons, calculating the logical error rate based on this procedure can become a cumbersome and lengthy endeavour. Hence, the task at hand is to find a more practical and less computationally demanding way to determine if the congruence equation system given in (5.3) has a solution, as this suffices to determine if the end-to-end error is degenerate (we do not actually need to solve the system itself).

5.2.1 CLASSICAL CODING INSPIRED STRATEGY

It is possible to find an answer to (5.3) by casting the problem in the framework of classical linear block codes. In classical coding theory, the encoding matrix or generator matrix $\mathbf{G}_c$ of a binary linear block code and its corresponding parity check matrix $\mathbf{H}_c$ fulfil $(\mathbf{G}_c\mathbf{H}_c^{\top}) \mod 2 = (\mathbf{H}_c\mathbf{G}_c^{\top}) \mod 2 = 0$. This means that the parity check matrix defines a basis for the null space of the generator matrix and viceversa. In the classical paradigm, having a basis for the null space of a code enables us to determine whether the decoding outcome $\mathbf{x}$ belongs to the code by simply computing its product with the parity check matrix of the code, i.e., if $(\mathbf{H}_c\mathbf{x}^{\top}) \mod 2 = 0$ then $\mathbf{x}$ is a codeword. Essentially, whenever $(\mathbf{H}_c\mathbf{x}^{\top}) \mod 2 = 0$, the decoding outcome is a linear combination of the rows of the generator matrix $\mathbf{G}_c$ and it belongs to the code, and whenever $(\mathbf{H}_c\mathbf{x}^{\top}) \mod 2 \neq 0$, $\mathbf{x}$ does not belong to the code.

[2]The PCMs of QEC codes are of size $N-k\times 2N$. A common size in the literature of QLDPC codes is 10000 qubits, thus the PCM associated to such a code would be of size $(10000-k)\times 20000$.

Notice that, based on this formulation, the quandary posed in (5.3) is reminiscent of the classical decoding scenario. The main difference is that instead of determining if the decoding outcome belongs to the code, we must discover if the sum of the symplectic representations of the channel error and the estimated error belong to the stabilizer. This parallelism between the classical and quantum problems allows us to apply the classical resolution strategy to the quantum paradigm with only a slight caveat: answering (5.3) requires an inverse approach to the classical method. Since the generators of the stabilizer code are given by the rows of the parity check matrix $\mathbf{H}_{\overline{\mathcal{S}}}$, the corresponding kernel generator matrix[3] $\mathbf{G}_{\overline{\mathcal{S}}}$ (instead of the parity check matrix like in the classical paradigm) must be used to discover if $\mathbf{e} \oplus \hat{\mathbf{e}}$ can be written as a linear combination of the stabilizer generators. The matrix $\mathbf{G}_{\overline{\mathcal{S}}}$ defines a basis for the nullspace of the stabilizer code, hence it will suffice to compute $[\mathbf{G}_{\overline{\mathcal{S}}}^{\top}(\mathbf{e} \oplus \hat{\mathbf{e}})^{\top}] \mod 2$ to find the answer to (5.3). If $[\mathbf{G}_{\overline{\mathcal{S}}}^{\top}(\mathbf{e} \oplus \hat{\mathbf{e}})^{\top}] \mod 2 = 0 \rightarrow \mathbf{E} \star \hat{\mathbf{E}} \in \overline{\mathcal{S}}$ which means that an end-to-end degenerate error has occurred, and if $[\mathbf{G}_{\overline{\mathcal{S}}}^{\top}(\mathbf{e} \oplus \hat{\mathbf{e}})^{\top}] \mod 2 \neq 0 \rightarrow \mathbf{E} \star \hat{\mathbf{E}} \notin \overline{\mathcal{S}}$, meaning that an end-to-end identical syndrome error will have taken place.

This strategy provides us with a simple and computationally efficient method to determine the type of end-to-end error that has taken place. The only requirement is obtaining the matrix $\mathbf{G}_{\overline{\mathcal{S}}}$, which can be computed once (by finding a basis for the nullspace of its parity check matrix $\mathbf{H}_{\overline{\mathcal{S}}}$) and can then be stored offline for any stabilizer code. In this manner, we have designed a simple method to solve (5.3) that does not require the computation of the stabilizer and so avoids the complexity issues that this entails.

5.2.2 DETECTING END-TO-END DEGENERATE ERRORS USING ENCODED PAULI OPERATORS

There is another manner of distinguishing between end-to-end identical syndrome errors and end-to-end degenerate errors. It involves obtaining the encoded Pauli operators (see section 4.2.3.1) of a code following a method derived by Gottesman in his seminal work [16], and then using these operators to determine whether the error estimate produced by the decoder is

[3] We refer to the matrix $\mathbf{G}_{\overline{\mathcal{S}}}$ as the kernel generator matrix to avoid the term stabilizer generator matrix, as this latter term implies that the the matrix can be used for encoding purposes (which may not be true in the present case).

in the stabilizer coset of the channel error. This strategy was first applied to QLDPC codes in [55], and has since been used in [151, 152, 153, 154].

Based on the definition of the encoded Pauli operators given in the previous chapter, we know that an encoded Pauli operator $\overline{\mathbf{Z}}_q$ commutes with all the elements of $\overline{\mathcal{S}}$ as well as with all other encoded Pauli operators except for the operator $\overline{\mathbf{X}}_l$ when $q = l$. This means that, if the encoded Pauli operators of a stabilizer code are known, we can determine if an operator $\mathbf{A} \in \overline{\mathcal{Z}}(\overline{\mathcal{S}}) \subset \overline{\mathcal{G}}_N$ belongs to $\overline{\mathcal{S}}$ by checking the commutation relations of $\mathbf{A}$ with the encoded Pauli operators. If $\mathbf{A}$ commutes with all the encoded Pauli operators it is within the stabilizer and if it does not (it anti commutes with one encoded Pauli operator) it is not in the stabilizer. Against this backdrop, it is easy to see how this strategy can be applied to solve the issue of discriminating between end-to-end identical syndrome errors and end-to-end degenerate errors. After successful decoding (the decoder produces a matching estimate of the syndrome), we know that $\mathbf{E} \star \hat{\mathbf{E}} \in \overline{\mathcal{Z}}(\overline{\mathcal{S}})$. Now, we determine if $\mathbf{E} \star \hat{\mathbf{E}}$ is in $\overline{\mathcal{S}}$ by checking its commutation status with $\{\overline{\mathbf{Z}}_q\}_{q=1}^{k}$ and $\{\overline{\mathbf{X}}_l\}_{l=1}^{k}$. If $\mathbf{E} \star \hat{\mathbf{E}}$ commutes with all the encoded Pauli operators, $\mathbf{E} \star \hat{\mathbf{E}} \in \overline{\mathcal{S}}$ and an end-to-end degenerate error has occurred. If not, an end-to-end identical syndrome error has occurred. It is based on these comparisons that the logical error rate was successfully computed in [151, 152, 153, 154].

Naturally, to be able to apply this method, one must first have knowledge of the encoded Pauli operators of the code. This is similar to the classical coding-based strategy we proposed earlier, which requires the computation of the kernel generator matrix $\mathbf{G}_{\overline{\mathcal{S}}}$. The encoded Pauli operators of a stabilizer code can be found based on the concept of the *standard form* (see Chapter 4 of [16]). In this work, Gottesman showed how, by applying row operations (i.e. Gaussian elimination) together with the necessary qubit permutations (i.e. column permutations) on the parity check matrix of a stabilizer code, one can obtain a special row reduced echelon form of the parity check matrix: the standard form. Once the standard form is known, the encoded Pauli operators $\{\overline{\mathbf{Z}}_q\}_{q=1}^{k}$ and $\{\overline{\mathbf{X}}_l\}_{l=1}^{k}$ can be directly obtained from it using matrix algebra [16, 158]. As with the kernel generator matrix $\mathbf{G}_{\overline{\mathcal{S}}}$, the encoded Pauli operators of a specific code need only be computed once and can then be stored offline.

5.3 FREQUENCY OF EACH TYPE OF END-TO-END ERROR

We now employ our method to study the frequency with which the different types of end-to-end error occur when using QLDPC codes. For this purpose, we simulate the CSS QLDGM codes of [21, 22, 23] with different rates and block lengths over the depolarizing channel. These codes are presented in detail in the next chapter. If desired, the reader can skip ahead for a rigorous discussion on their design. However, this is not necessary to understand the remainder of this chapter. For reference, the particular characteristics of the simulated codes are detailed in table 5.2.

The results of our simulations are shown in Figure 5.1, where each subfigure groups the results by block length, i.e, each of the subfigures portrays the results for all the codes with the same value of N. The graphs plot the ratio of a specific type of end-to-end error against the depolarizing probability. The aforementioned ratio is computed as $\frac{\mathrm{E}_i}{\mathrm{E}_T}$, where E_i denotes the total number of end-to-end errors of a specific type ($i = 1, 2, 3.$):

- $\mathrm{E}_1 \rightarrow$ end-to-end errors with different syndromes,
- $\mathrm{E}_2 \rightarrow$ end-to-end errors with identical syndromes,
- $\mathrm{E}_3 \rightarrow$ end-to-end degenerate errors,

and E_T represents the total number of end-to-end errors. To ensure that the simulation results are precise, the ratios have been computed after a total of 1000 decoding mistakes have been made (following the Monte Carlo simulation rule of thumb provided in [162]), i.e., $\mathrm{E}_T = 1000$. For a complete discussion on how these simulations are conducted the reader is referred to Appendix C.

The outcomes portrayed in Figure 5.1 confirm our initial intuition that sparse quantum codes are degenerate. It is easy to see that for all of the simulated block lengths and rates (except for $R_Q = 0.5$), the percentage of end-to-end errors that are not of the E_1 type is not negligible, i.e., $\frac{\mathrm{E}_2}{\mathrm{E}_T} + \frac{\mathrm{E}_3}{\mathrm{E}_T} \neq 0$. Furthermore, these results speak towards the higher precision of the logical error rate compared to the physical error rate when assessing the performance of these codes. For instance, at a noise level of $p = 0.005$, $\frac{\mathrm{E}_3}{\mathrm{E}_T} = 0.198$ for the $N = 500$ $R_Q = 0.1$ code. This means that 19.8% of the end-to-end errors are degenerate and should not be counted as decoding

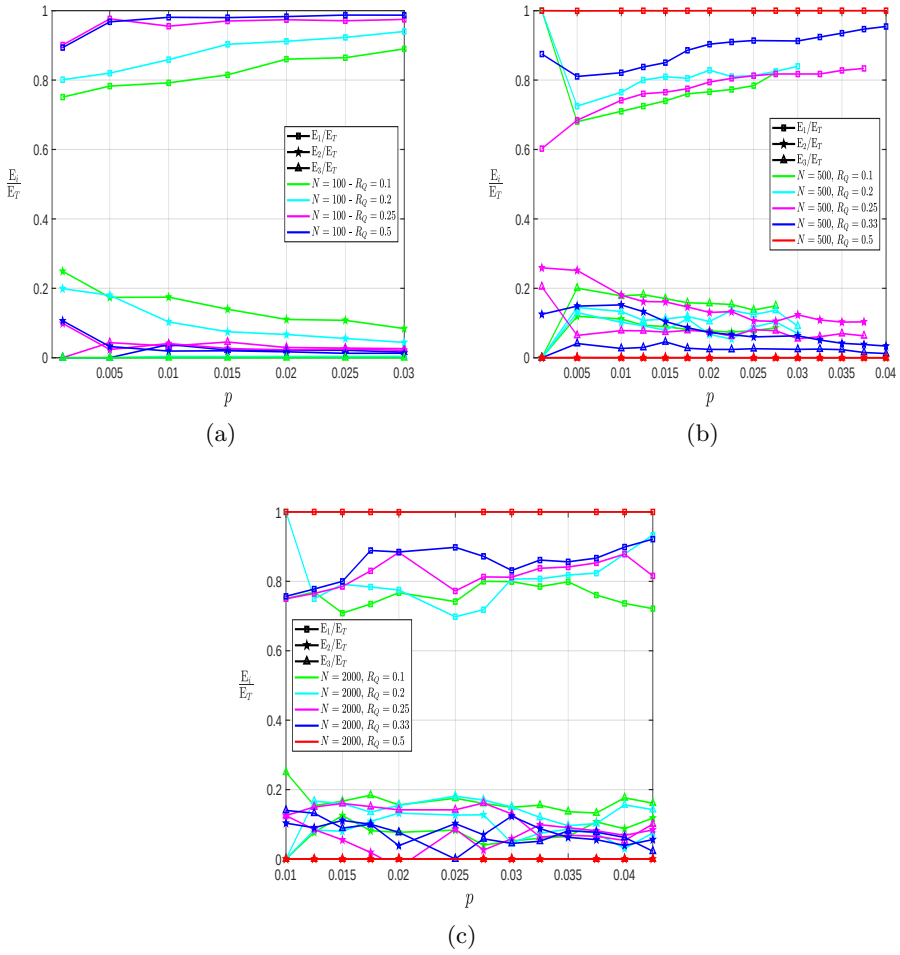
Table 5.2: *Parameter values and configurations of simulated codes.*

N	R_Q	Classical LDGM	$[m, p, x, y]$
100	0.1	P(3, 3)	[45, 24, 6, 3]
100	0.2	P(3, 3)	[40, 18, 6, 3]
100	0.25	P(3, 3)	[38, 15, 6, 3]
100	0.5	P(3, 3)	[25, 6, 7.57, 3]
500	0.1	P(5, 5)	[225, 170, 11, 3]
500	0.2	P(5, 5)	[200, 144, 11, 3]
500	0.25	P(5, 5)	[188, 130, 11, 3]
500	0.33	P(5, 5)	[163, 102, 11, 3]
500	0.5	P(5, 5)	[125, 60, 11, 3]
2000	0.1	P(9, 9)	[900, 691, 11, 3]
2000	0.2	P(9, 9)	[800, 581, 11, 3]
2000	0.25	P(9, 9)	[750, 526, 11, 3]
2000	0.33	P(9, 9)	[670, 438, 11, 3]
2000	0.5	P(9, 9)	[500, 251, 11, 3]

failures. Thus, in this scenario, the physical error rate overestimates the number of decoding failures and does not provide an accurate representation of the performance of the code. In fact, regardless of the noise level of the channel, the rate of the code (except for $R_Q = 0.5$), and the block length of the code, end-to-end degenerate errors take place, and so the physical error rate will always provide an inaccurate representation of the performance of these sparse quantum codes. Therefore, as is stated in [154], it is clear that performance results assessed based on the physical error rate ($\hat{\mathbf{E}} = \mathbf{E}$ as the decoding success criterion) [20, 21, 22, 23, 149, 157] are inaccurate (they report an upper bound).

Furthermore, the results shown in Figure 5.1 also reveal how the frequency with which each type of end-to-end error takes place varies as a function of different parameters:

- End-to-end errors with different syndromes represent a large percentage of the total number of end-to-end errors when the rate of the code is high. This percentage decreases as the rate of the codes goes from



(a) (b)

(c)

Figure 5.1: *Ratios of end-to-end different syndrome errors, identical syndrome errors, and degenerate errors for codes of various rates and block lengths:* ***(a)*** $N = 100$ ***(b)*** $N = 500$ ***(c)*** $N = 2000$. .

$R_Q = 0.5$ to $R_Q = 0.1$ (see ratio $\frac{\mathrm{E}_1}{\mathrm{E}_T}$ in Figure 5.1). This trend becomes further exacerbated as the block length of the simulated codes increases, i.e., for low rate large block length codes the ratio $\frac{\mathrm{E}_1}{\mathrm{E}_T}$ will be significantly smaller than for low rate short codes.

- End-to-end identical syndrome errors represent the smallest percentage of the total number of end-to-end errors in most of the simulated cases. This is reflected by the fact that $\frac{\mathrm{E}_2}{\mathrm{E}_T} < \frac{\mathrm{E}_1}{\mathrm{E}_T}, \frac{\mathrm{E}_3}{\mathrm{E}_T}$ in all of our simulation outcomes.
- As the noise level of the channel grows, the ratio $\frac{\mathrm{E}_1}{\mathrm{E}_T}$ becomes larger and the ratio $\frac{\mathrm{E}_2}{\mathrm{E}_T}$ diminishes. The ratio of end-to-end degenerate errors $\frac{\mathrm{E}_3}{\mathrm{E}_T}$ stays relatively constant.

The relationships between these parameters and different types of end-to-end errors serve to draw important conclusions. For instance, the large values of $\frac{\mathrm{E}_1}{\mathrm{E}_T}$ in many of the simulated instances can be understood as a sign that performance gains may be attained by improving the decoding algorithm. This means that applying modified decoding strategies, such as those of [135, 149, 151, 152, 153], will aid in reducing the presence of end-to-end errors with different syndromes and improve performance. A matter that should also be considered is how often these strategies produce failed error corrections in the form of end-to-end identical syndrome errors. For the methodology of [149], such events were shown to be rare, hence we expect these strategies to be a good approach to improve the performance of QLDPC codes. We discuss modified decoding strategies for QLDPC codes in the next section of this chapter.

In terms of other end-to-end error ratios, the large values of $\frac{\mathrm{E}_3}{\mathrm{E}_T}$ when compared to $\frac{\mathrm{E}_2}{\mathrm{E}_T}$, especially at higher block lengths, show that end-to-end identical syndrome errors are the least frequent of all the end-to-end error types. Despite the relatively small percentage that end-to-end identical syndrome errors represent, it is possible that their relevance will grow when the amount of end-to-end errors with different syndromes is reduced (using modified decoding strategies) or when the degenerate content of the code is increased (through design). At this point, it may be that further improvements in performance will only be possible by designing an optimal degenerate decoder with the capability to correct end-to-end identical syndrome errors. Finally, given that the method proposed in 5.2.1 is valid to detect end-to-end degenerate errors, it could be interesting in future

work to employ this methodology to specifically design codes to be degenerate. This might lead to code constructions whose probability of suffering end-to-end degenerate errors is maximized, which would allow the positive effects of degeneracy (improved error correction capabilities without a decoding complexity increase) to be completely exploited for quantum error correction purposes.

5.4 IMPROVED DECODING STRATEGIES FOR QLDPC CODES

We close this chapter by providing a succinct summary of the most relevant improved decoding strategies that have been proposed in the literature. A more thorough look into each of these methods is provided in Appendix B.

In the previous section we discussed how $\frac{\mathrm{E}_1}{\mathrm{E}_T}$, the ratio of end-to-end errors with different syndromes, accounts for a large portion of the decoding mistakes of the simulated QLDGM codes. Fortunately, although they cause decoding mistakes, these errors are always detected (the estimated syndrome is not equal to the measured syndrome). This means that the decoder will know when these errors occur, an outcome that can be used to apply certain modifications to the original QMLD strategy to try to revert the errors. Eliminating the presence of these end-to-end errors may lead to drastic performance improvements, and although it does not address the issue of constructing a degenerate decoder (end-to-end identical syndrome errors will still occur), it represents an important frontier for QLDPC code design.

The earliest mention of improved decoding strategies for QLDPC codes dates back to the work of Mackay et al. [18], who proposed a strategy to adapt the conventional SPA decoder to the depolarizing channel. Later on, this technique was applied by Lou et al in [22], where a modified decoder was built for the depolarizing channel and shown to surpass traditional BP decoders. Following this, a set of heuristic methods that successfully improved the ability of the classical qubit-wise BP decoder to handle detected harmful errors was built in [55]. In this work, the authors also discussed the concept of symmetric degeneracy errors and proposed a host of techniques (freezing, collision, and random perturbation) capable of improving the performance of QLDPC codes under BP decoding. Simultaneously, an enhanced-feedback decoder specifically designed for the depolarizing chan-

nel was derived in [149]. Later on, in [20], a modified decoder for dual-containing CSS QLDPC codes which also led to performance improvements was introduced. Recently, in [135, 151, 152, 153, 154], a number of new and improved decoding strategies have been proposed. A timeline of the most significant contributions to the field of modifed QLDPC decoding is provided in table 1.

Except for the decoders proposed in [22, 20, 153], all of these heuristic methods operate according to the same working principle: prior to applying any changes to the decoder, the original qubit-wise BP decoder must fail. Generally, these modified decoding processes begin by running a standard unaltered version of the qubit-wise BP decoder. If the standard decoder finds an error estimate whose corresponding syndrome is equal to the real one, i.e., $\hat{\mathbf{w}} = \mathbf{w}$, then the procedure halts and assumes that the error estimate is correct. However, if the standard decoder fails to find a matching estimate of the syndrome after the allotted number of iterations, changes are made to the decoder (according to specific techniques) and decoding is reattempted in order to fix this decoding impasse.

These decoding alterations combine a range of strategies to revert the impact of end-to-end errors with different syndromes. Some techniques, like those of [18, 22, 55, 135, 149], utilize the incorrect error sequence and the syndrome estimates of the original decoder to compute new a priori probabilities that, when used by the decoder, may result in a matching syndrome estimate. Others, like the augmented decoder of [135] and the supernode decoder of [20], rely on changing the structure of the original factor graph. Also in [135], the authors apply the two proposed approaches in unison, so that when one fails the other is applied and viceversa, a technique called the combined decoder. Finally, the most recent strategies of [151, 152, 153, 154] focus more on post-processing and direct modifications to the decoding algorithm.

Table 5.3: *Most significant modified decoding strategies for QLDPC codes.*

2004	Initial Proposal for a Modified SPA-based decoding Strategy [18].
2005	Correlation Exploiting Decoder [22].
2008	Freezing, Collision & Random Perturbation Decoders [55].
2012	Enhanced Feedback Decoder [149].
2015	Supernode Decoder [20].
2019	Adjusted & Augmented Decoders [135].
2019/2020	Ordered Statistics Decoder [151, 152].
2020	Refined Belief Propagation Decoding [153].
2021	Degeneracy Exploiting Decoder [154].

Regardless of the specific modified decoding method that is employed, the impact of its modifications spreads quickly during the new decoding phase, mostly due to the presence of short cycles in QLDPC codes. This means that, while short cycles hinder the traditional BP decoder, they benefit these modified decoding strategies [55, 135, 149]. The downside of these heuristic schemes comes in the form of an increased decoding complexity. Because most of these techniques rely on the failure of a standard decoder and a re-execution of the modified process anew, the chosen method

sometimes requires numerous decoding attempts before it finds the correct syndrome estimate. With regard to performance, the degree to which it can be improved varies depending on the chosen strategy; the techniques of [55, 135] can yield performance improvements of about 30% in terms of the word error rate over traditional decoding, while the enhanced feedback decoding technique of [149] has been shown to be approximately 10 times better than the traditional SPA decoder, even though it only requires a 25-35% increase in the number of decoding iterations. In [151, 152] the Ordered Statistics Decoder (OSD) is shown to outperform all of the aforementioned modified decoding techniques.

In any case, aside from a potential increase in complexity, these heuristic methods are relatively simple tools that can be easily included within conventional qubit-wise BP decoders to improve the performance of QLDPC codes under BP/SPA decoding.

5.5 CHAPTER SUMMARY

In this chapter we have presented a method to detect degenerate errors in sparse quantum codes in a computationally efficient manner. Based on this method, we have shown how the physical error rate provides an inaccurate representation of the performance of sparse quantum codes and we have shown that logical error rate should be used instead. The discrepancy between the logical error rate and the physical error rate is especially relevant to the field of sparse quantum codes because of their degenerate nature. This is reflected by the results we have obtained for a specific family of QLDPC codes, whose performance can be up to 20% better than would be expected based on previous results in the literature. In addition, these simulation outcomes serve to show how performance may be improved by constructing degenerate quantum codes, and they also speak toward the positive impact that modified decoding strategies can have on the performance of sparse quantum codes. The last portion of this chapter surveys the literature on these modified decoding strategies, most of which are thoroughly discussed in the Appendix.

CHAPTER 6

Non-CSS QLDPC codes

"The most important step a man can take. It's not the first one, is it? It's the next one. Always the next step..."

Brandon Sanderson.

Up to this point in the dissertation our focus has been (for the most part) theoretical. Aside from studying the phenomenon of degeneracy, what we have seen thus far has served to lay out the necessary groundwork to understand how QEC works and why it is important. This context should be useful to understand the remaining chapters of the dissertation, where our attention will shift to the design and performance of a specific family of QLDPC codes. Before getting started with Chapter 6, in the following discussion we provide some insight into why we chose to work with sparse quantum codes.

WHY BUILD QEC CODES USING LDPC CODES?

In the realm of classical communications, turbo codes [46, 166] and LDPC codes [110, 111, 112, 167, 168, 169] are known to exhibit capacity approaching performance at a reasonable decoding computational complexity. In particular, turbo codes offer great flexibility when it comes

to choosing their block length and rate, whereas the sparsity of LDPC codes makes them easy to decode. Given their appealing characteristics, the scientific community has searched diligently for the quantum equivalents of classical turbo and LDPC codes. Quantum codes based on turbo codes first appeared in [28, 29], and have since been modified and improved [30, 31, 170, 171, 172]. In turn, QLDPC codes were first analyzed in [18], were their sparse nature was shown to be especially advantageous in the quantum paradigm because it ensures that a smaller number of quantum interactions per qubit is necessary during the error correction procedure. Because quantum gates are faulty, having less quantum interactions per qubit avoids additional quantum operations, which minimizes errors and facilitates fault-tolerant computing [88, 173, 174]. Thus, QLDPC codes can be said to be particularly well suited for quantum error correction.

Out of the existing types of LDPC codes, LDGM codes [175] provide a seamless manner for code design in the quantum domain. LDGM codes are a specific subset of LDPC codes whose generator matrices are also sparse, and thus their encoding complexity is similar to that of turbo codes, and much lesser than for standard LDPC codes. Given that LDGM codes form a special subclass of the LDPC code family, they can be decoded in the same manner and with the same complexity as any other LDPC code. LDGM codes have been extensively studied [155, 175, 176] and used in classical communications [177, 178]. In [112], regular LDGM codes were studied and shown to be asymptotically bad, displaying error floors that do not decrease with the block length. In [155] and [176], a concatenated LDGM scheme was shown to achieve irregular LDPC code-like performance at a very low encoding/decoding complexity.

Based on what was shown in Chapter 4, we know that quantum codes can be built from classical codes by casting them in the framework of stabilizer codes [16]. In [20], the authors document the design of QLDPC codes from their classical counterparts based on the stabilizer formalism by detailing numerous construction and decoding techniques along with their flaws and merits. Among the discussed methods, the construction of QLDPC codes based on LDGM codes is shown to yield performance and code construction improvements, albeit at an increase in decoding complexity. This method was originally proposed in [21] and [22], where CSS quantum codes based on regular LDGM classical codes were shown to surpass the best quantum coding schemes of the time. Later, performance was significantly

improved in [155] and [176] by utilizing a parallel concatenation of two regular LDGM codes.

As with most QLDPC designs, QLDGM-based quantum code implementations are based on CSS constructions. CSS codes, simultaneously proposed by Calderbank, Shor, and Steane in [57], [58], are a particular subset of the stabilizer code family. They provide a straightforward method to design quantum codes via existing classical codes. However, due to the specific nature of CSS constructions, their performance is limited by an unsurpassable bound, referred to as the *CSS lower bound* [62]. This inspires the search for non-CSS constructions, as they should theoretically be able to outperform CSS codes if designed optimally. Non-CSS LDPC-based codes were proposed in [63] and [64]. However, despite showing promise, they fail to outperform existing CSS QLDPC codes for comparable block lengths.

In this chapter, we introduce a non-CSS scheme based on LDGM codes and compare its performance to existing CSS QLDGM codes. We begin by explaining how CSS codes are constructed and how they can be modified to create non-CSS quantum codes. Following this, we provide insight regarding how our construction is optimized, and we show how the performance of our non-CSS codes is similar to that of the CSS schemes they are derived from, despite the fact that their quantum rate is higher. When their rate is the same, the non-CSS scheme outperforms the original CSS design over the depolarizing channel. Finally, we compare the non-CSS structures proposed here with other existing QLDPC codes in the literature, illustrating that our method surpasses such error correction schemes.

The chapter is structured as follows. We commence with a brief presentation of important preliminary topics, such as CSS codes or LDGM codes, that did not appear in Chapter 3. We proceed by presenting CSS LDGM based codes in section 6.2 and our non-CSS LDGM-based strategy in section 6.3. In section 6.4, we compare the performance of the proposed scheme to existing CSS quantum codes.

6.1 PRELIMINARIES

In this section we provide an overview of some preliminary topics in the field of QEC that were not presented in Chapter 3.

6.1.1 PAULI CHANNELS

We saw in Chapter 3 that the most popular quantum channel model is the generic Pauli channel ξ_p. To recap, the effect of the Pauli channel ξ_p upon an arbitrary quantum state described by its density matrix ρ can be written as

$$\xi_p(\rho) = (1 - p_x - p_y - p_z)\rho + p_x X\rho X + p_y Y\rho Y + p_z Z\rho Z.$$

This expression can be interpreted as the quantum state experiencing a bit-flip (X Pauli operator) with probability p_x, a phase-flip (Z Pauli operator) with probability p_z, or a combination of both (Y Pauli operator) with probability p_y.

6.1.1.1 Depolarizing channel

We also mentioned previously that most of the QEC literature considers the independent depolarizing channel model. This model is a specific instance of the Pauli channel in which the depolarizing probabilities are all equal, i.e, $p_x = p_z = p_y = p$, and the channel is characterized by the depolarizing probability p. When quantum states of N qubits are considered, the errors that take place belong to the N-fold Pauli[1] group $\mathcal{G}_N$. These N-qubit error operators are made up of single qubit Pauli operators that act independently on each qubit causing an X, Z, or Y error with probability $p/3$ and leaving it unchanged with probability $(1 - p)$.

6.1.1.2 i.i.d. X/Z Channel

A simpler quantum channel model, known as the *standard flipping channel* or *i.i.d. X/Z channel*, was introduced in [18], where Z and X errors are modelled as independent events identically distributed (i.i.d.) according to the flip probability f_m. This quantum channel model is analogous to two independent Binary Symmetric Channels (BSCs) with marginal bit flip probability $f_m = 2p/3$, where the separate BSCs can be seen as Z and X error channels, respectively. Given that Y errors occur when both a phase and a bit-flip happen to the same qubit, the simplified notion of the i.i.d.

[1]Recall that the global phase has no observable consequence, so we can consider the channel errors to be elements of the effective N-fold Pauli group.

X/Z channel ignores any correlation that exists between X and Z errors in the depolarizing channel.

6.1.2 THE HASHING BOUND

In the introduction to section 3.2 we mentioned how the classical channel capacity establishes the maximum information transfer rate at which reliable communications can take place over a classical channel. Unsurprisingly, we can also derive the quantum counterpart of this concept and use it to study the theoretical limits of quantum channels.

The capacity of a quantum channel is defined as the highest possible achievable rate at which quantum information can be asymptotically transmitted in an error-less manner. Unfortunately, a closed formula for the capacity of the Pauli channel is not known [97, 179, 180], which means that a closed formula for the capacity of the depolarizing channel (the most prominent channel model for decoherence in the literature) is also unknown. Instead, we work with a quantity known as the Hashing bound, which defines a lower bound on the capacity of the depolarizing channel and is computed as $C_{\text{Hash}}(p) = 1 - H_2(p) - p\log_2 3$, where $H_2(p)$ is the binary entropy function and p represents the depolarizing probability [101]. This means that for a given value of p, $C_{\text{Hash}}(p)$ represents a lower bound on the highest possible coding rate at which asymptotically error-free quantum communication is possible. In theory, it is possible for quantum codes to surpass the Hashing bound due to the phenomenon of degeneracy (which is also the reason why a closed formula for Pauli channels is difficult to obtain) [54]. Alternatively, for a specific quantum rate R_Q, where we have $R_Q = C_{\text{Hash}}(p^*)$, p^* represents a bound on the channel's depolarizing probability [165]. As in the classical domain, we can refer to p^* as the *noise limit*.

6.1.2.1 Distance to the Hashing bound

Ideally, quantum codes that are properly designed should ensure error-free communications close to the noise limit p^*. Thus, we can efficiently characterize the quality of quantum codes of a specific quantum rate R_Q built for the depolarizing channel by assessing how far away they are from the Hashing bound. The distance from the Hashing bound can be computed based on the expression

$$\delta = 10\log_{10}\left(\frac{p^*}{p}\right), \tag{6.1}$$

where we use δ to represent the distance to the Hashing bound in decibels (dB), p^* is the noise limit of the depolarizing channel for a specific quantum coding rate R_Q, and p is the highest depolarizing probability at which the code in question can operate in an error-free manner.

6.1.3 CSS CODES

Based on our previous discussion on stabilizer codes (see section 4.2.3.3), we know that two binary classical LDPC codes can only be used to construct a quantum stabilizer code if they satisfy the symplectic criterion (4.24). With this knowledge, the first design strategy one could devise to construct quantum stabilizer codes would be to randomly select pairs of classical LDPC codes and combine them into a QPCM. However, finding two LDPC codes of reasonable block size that satisfy (4.24) is highly unlikely. Calderbank-Shor-Steane (CSS) codes [57], [58], provide a more efficient design strategy than random selection of classical codes. The quantum parity check matrix of these codes is written as

$$\mathbf{H}_Q = (\mathrm{H}_x|\mathrm{H}_z) = \begin{pmatrix} \mathrm{H}_x^{'} & 0 \\ 0 & \mathrm{H}_z^{'} \end{pmatrix}, \tag{6.2}$$

where $\mathrm{H}_x = \begin{pmatrix} \mathrm{H}_x^{'} \\ 0 \end{pmatrix}$ and $\mathrm{H}_z = \begin{pmatrix} 0 \\ \mathrm{H}_z^{'} \end{pmatrix}$.

In this construction, H_x' and H_z' are the parity check matrices of two binary classical LDPC codes C_1 and C_2, respectively, where each matrix is used to correct either bit-flips or phase-flips. The classical codes are chosen so that $C_2^{\perp} \subseteq C_1$, where $C_2^{\perp}$ is the dual of the classical LDPC code C_2. This design constraint, generally referred to as the *CSS condition*, reduces (4.24) to $(\mathrm{H}_x^{'}\mathrm{H}_z^{'\top}) = 0$.

6.1.4 SYSTEMATIC CLASSICAL LDGM CODES

Systematic LDGM codes are useful, both in classical and quantum environments, because of the particular structure of their generator and parity

check matrices. Let C be a binary systematic LDGM code. Then, its generator matrix $\tilde{\mathrm{G}}$ and its parity check matrix $\tilde{\mathrm{H}}$ can be written as

$$\begin{aligned} \tilde{\mathrm{G}} &= (\mathrm{I}\ \mathrm{P}) \\ \tilde{\mathrm{H}} &= (\mathrm{P}^{\top}\ \mathrm{I}), \end{aligned} \tag{6.3}$$

where I denotes the identity matrix, and P is a sparse matrix.

Because LDGM codes belong to the family of linear block codes, these matrices will satisfy $\tilde{\mathrm{G}}\tilde{\mathrm{H}}^{\top} = \tilde{\mathrm{H}}\tilde{\mathrm{G}}^{\top} = 0$. Those systematic LDGM codes whose variable and parity check nodes have degrees[2] x and y, respectively, will be denoted as (x, y) regular LDGM codes. Regular LDGM codes are known to be asymptotically bad [112], displaying error floors that do not decrease with the block length. However, in [181], codes built via the parallel concatenation of two regular LDGM codes[3] were shown to yield significant reduction in these error floors. The parallel concatenation of two regular LDGM codes C_1 and C_2 with generator matrices $\mathrm{G}_1 = [\mathrm{I}\ \mathrm{P}_1]$ and $\mathrm{G}_2 = [\mathrm{I}\ \mathrm{P}_2]$, where P_1 and P_2 have degree distributions (y_1, y_1) and (y_2, z_2), is the irregular LDGM code with generator matrix $\mathrm{G} = [\mathrm{I}\ \mathrm{P}_1\mathrm{P}_2]$. Generally, this concatenation is accomplished by using a high rate code C_2 that is able to reduce the error floor of C_1, while also causing negligible degradation of the original convergence threshold.

With regard to decoding, because LDGM codes are a specific subset of LDPC codes, they are decoded in exactly the same manner as generic LDPC codes. In Chapter 3 we showed that classical LDPC syndrome-based decoding is performed by solving the equation $\mathbf{z} = \mathbf{He}$, where $\mathbf{z}$ represents the received classical syndrome, $\mathbf{H}$ is the PCM of the code, and $\mathbf{e}$ is the error pattern that has corrupted our information (a rigorous description on the details of this procedure is provided in Appendix A). It should be mentioned that it is also possible to decode systematic LDGM codes by solving equation $\mathbf{c} = \mathrm{P}\mathbf{u}$, where $\mathbf{c}$ is the vector of parity bits generated at the encoder, P is the constituent sparse matrix of the LDGM generator matrix (see equation (6.3)), and $\mathbf{u}$ is the information message we want to

[2]The degree of the variable nodes is the number of nonzero entries per column of the PCM. The degree of the parity check nodes is given by the number of nonzero entries per row of the PCM. An LDGM code is said to be regular when all the rows of its PCM have the same number of nonzero entries, x, and so do its columns, y.

[3]The parallel concatenation of regular LDGM codes is equivalent to an LDGM code with an irregular degree distribution.

obtain. This means that the decoding algorithm for LDGM codes can also be implemented by applying the SPA [48, 47] over the graph associated to the equation $\mathbf{c} = \mathrm{P}\mathbf{u}$.

6.2 CSS LDGM-BASED CODES

Based on what we have seen thus far, our first intuition to derive the QPCM of a QLDGM CSS code would be to select any classical LDGM code with parity check and generator matrices $\tilde{\mathrm{H}}$ and $\tilde{\mathrm{G}}$, and set $\mathrm{H}'_z = \tilde{\mathrm{H}}$ and $\mathrm{H}'_x = \tilde{\mathrm{G}}$ in (6.2), since the property $\tilde{\mathrm{G}}\tilde{\mathrm{H}}^\top = \tilde{\mathrm{H}}\tilde{\mathrm{G}}^\top = 0$ would ensure the fulfilment of the symplectic criterion. However, this results in a QPCM $\mathbf{H}_Q$ of size $N \times 2N$, which cannot be used for encoding purposes. This is easy to see based on the following discussion.

Consider a stabilizer code that has $(N-k)$ stabilizer generators and whose QPCM is of of size $(N-k) \times 2N$. Such a quantum code encodes k logical qubits into N physical qubits, which implies that the code has a quantum rate $R_Q = \frac{k}{N}$. This means that the quantum rate of a code with QPCM $\mathbf{H}_Q$ of size $N \times 2N$ is $R_Q = 0$. Therefore, to build a valid quantum code from classical LDGM codes, we must reduce the number of rows in $\mathbf{H}_Q$ while ensuring that the *CSS condition* is fulfilled. In [21], the authors successfully achieve this via linear row operations. They do so by applying the following theorem.

Theorem 3. *Given the generator and parity check matrices of a systematic LDGM code (6.3), define* $\mathrm{H}_{m_1 \times N} = [\mathrm{M}_1]_{m_1 \times n_1}[\tilde{\mathrm{H}}]_{n_1 \times N}$ *and* $\mathrm{G}_{m_2 \times N} = [\mathrm{M}_2]_{m_2 \times n_2}[\tilde{\mathrm{G}}]_{n_2 \times N}$*, where* $n_1 + n_2 = N$ *and* M_1 *and* M_2 *are low-density full-rank binary matrices whose number of rows satisfy* $m_1 < n_1$ *and* $m_2 < n_2$*, respectively. Then, the quantum PCM shown in (6.4), obtained by setting* $\mathrm{H}'_x = \mathrm{H}$ *and* $\mathrm{H}'_z = \mathrm{G}$ *in (6.2), is the quantum PCM of an LDGM-based CSS code with rate* $R_Q = \frac{N - m_1 - m_2}{N}$*.*

$$\mathbf{H}_Q = (\mathrm{H}_x|\mathrm{H}_z) = \begin{pmatrix} \mathrm{H} & 0 \\ 0 & \mathrm{G} \end{pmatrix} = \begin{pmatrix} \mathrm{M}_1\tilde{\mathrm{H}} & 0 \\ 0 & \mathrm{M}_2\tilde{\mathrm{G}} \end{pmatrix}. \tag{6.4}$$

We showed in Chapter 4 that sparse quantum codes are decoded by running the SPA algorithm over the factor graph associated to the expression

$$\mathbf{w} = \mathbf{H}_{\overline{\mathcal{S}}} \odot \mathbf{e} = \mathbf{e}_x\mathrm{H}_z^\top \oplus \mathbf{e}_z\mathrm{H}_x^\top,$$

where $\mathbf{H}_{\bar{\mathcal{S}}}$ denotes the QPCM of the stabilizer code in question.

Thus, CSS LDGM codes will be decoded by applying the SPA over the factor graph defined by $\mathbf{w} = \mathbf{H}_Q \odot \mathbf{e}$, where $\mathbf{H}_Q$ is given in (6.4). We can easily derive this factor graph based on the unique structure of a CSS QPCM and the properties of the symplectic representation, which allows us to split $\mathbf{e} = (\mathbf{e}_x|\mathbf{e}_z)$. Notice how, given the diagonal structure of $\mathbf{H}_Q$, the terms $\mathbf{e}_x\mathrm{H}_z^\top$ and $\mathbf{e}_z\mathrm{H}_x^\top$ have no non-zero overlaps. This means that the syndrome is made up of two separate parts, $\mathbf{w} = [\mathbf{w}_x, \mathbf{w}_z]$, where $\mathbf{w}_x = \mathbf{e}_x\mathrm{H}_z^\top$ and $\mathbf{w}_z = \mathbf{e}_z\mathrm{H}_x^\top$. Thus, we can actually decode CSS codes separately, i.e, by generating two separate subgraphs associated to $\mathbf{w}_x$ and $\mathbf{w}_z$. In the following, we illustrate this derivation for $\mathbf{w}_x = \mathbf{e}_x\mathrm{H}_z^\top$. The procedure for $\mathbf{w}_z = \mathbf{e}_z\mathrm{H}_x^\top$ is identical but using G instead of H in (6.5). We can write the first half of the syndrome of a CSS code, $\mathbf{w}_x^\top$ (in column form), as

$$\mathbf{w}_x^\top = (\mathbf{e}_x\mathrm{H}_z^\top)^\top = \mathrm{H}_z\mathbf{e}_x^\top = \mathrm{M}_1\tilde{\mathrm{H}}\mathbf{e}_x^\top = \mathrm{M}_1[\mathrm{P}^\top\ \mathrm{I}]\mathbf{e}_x^\top. \tag{6.5}$$

If we now split $\mathbf{e}_x^\top = [\mathbf{e}_{x_1}\ \mathbf{e}_{x_2}]^\top$, we can write

$$\begin{aligned} \mathrm{d}_x &= [\mathrm{P}^\top\ \mathrm{I}]\mathbf{e}_x^\top = [\mathrm{P}^\top\ \mathrm{I}]_{n_1\times N}\begin{pmatrix}\mathbf{e}_{x_1}\\ \mathbf{e}_{x_2}\end{pmatrix}_{N\times 1} \\ &= \mathrm{P}^\top_{n_1\times n_2}[\mathbf{e}_{x_1}^\top]_{n_2\times 1} + [\mathbf{e}_{x_2}^\top]_{n_1\times 1}. \end{aligned} \tag{6.6}$$

We then relate d_x to $\mathbf{w}_x^\top$ as

$$[\mathbf{w}_x^\top]_{m_1\times 1} = \mathrm{M}_{1_{m_1\times n_1}}\mathrm{d}_{x_{n_1\times 1}}. \tag{6.7}$$

The factor graph shown in Figure 6.1 is obtained based on expressions (6.6) and (6.7), as well as their equivalents when using e_z and G in (6.5).

As mentioned previously, and upon closer examination of the QPCM shown in in (6.4), it is easy to see that decoding for the H and G matrices can be done separately. Notice how this is visible in Figure 6.1, where the leftmost subgraph is associated to the matrix H, which is used to decode $\mathbf{e}_x$, and the rightmost subgraph is associated to the matrix G, which is used to decode $\mathbf{e}_z$. Separate decoding of these matrices is made possible by the nature of CSS constructions, which results in syndrome nodes containing

information only of either X or Z operators, hence why we can write $\mathbf{w} = [\mathbf{w}_x, \mathbf{w}_z]$. This is reflected on the factor graph by the fact that a specific s node connects to either a d_x or a d_z node. Note that, for notation purposes, we refer to the syndrome nodes of the factor graph (top level diamonds in Figure 6.1) as s nodes.

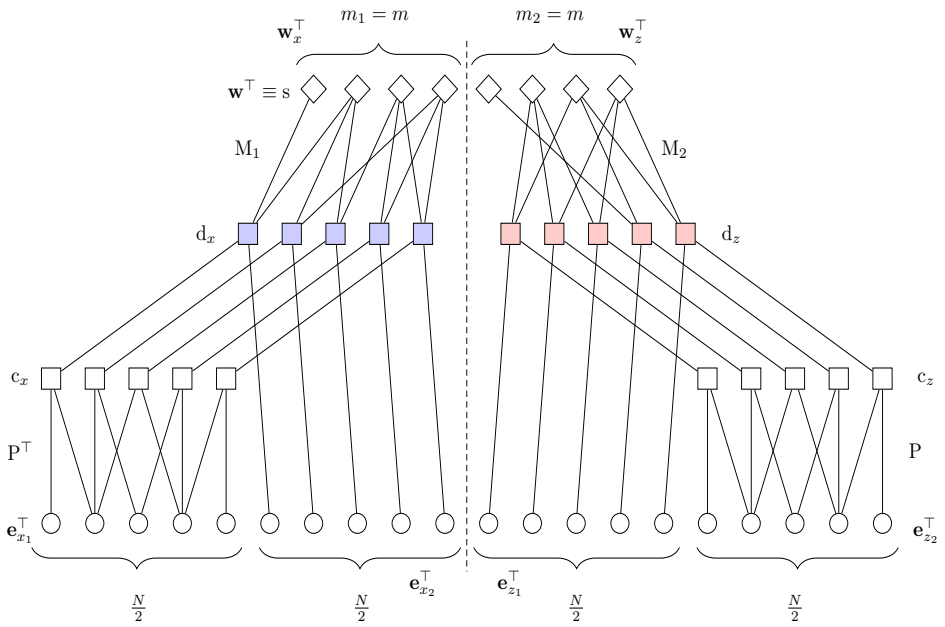


Figure 6.1: *Decoding graph for a QLDGM CSS scheme. The dotted line is included to emphasize the separation of the two constituent subgraphs. The top layer nodes (diamonds) represent the syndrome nodes, which we will denote as* s *nodes. The leftmost subgraph decodes the X operators while the one on the right decodes the Z operators. We have assumed that $m_1 = m_2$, $n_1 = n_2 = \frac{N}{2}$, and $m = m_1 + m_2$.*

The matrix multiplications used to perform the linear row operations on $\tilde{\mathrm{H}}$ and $\tilde{\mathrm{G}}$ generate a middle layer, represented by the c and d nodes, in both decoding subgraphs of Figure 6.1. This new layer hampers the

decoding algorithm, especially during the initial decoding iterations, since *a priori* information regarding the aforementioned middle layer nodes is not be available. This can be seen in Figure 6.2, where a generic quantum communication system is shown. The LDGM decoder block of this figure, which runs the SPA over the graph shown in Figure 6.1, has the syndrome $\mathbf{w}$ and the *a priori* probability of the error pattern P_{ch} as its inputs. However, it receives no information pertaining to the c and d nodes.

In [21], the authors circumvent this lack of information by using the so-called *doping* technique of [182]. This method introduces degree-1 syndrome nodes into the decoding graph. These degree-1 nodes, which we will refer to as s_A nodes, transmit correct information to the d nodes they are connected to, ultimately pushing the decoding process in the right direction. They are embodied within the M_1 and M_2 matrices as rows with a single non-zero entry, which corresponds to the edge that connects a given s_A node to a d node. The other rows of matrices M_1 and M_2, which correspond to the rest of the s nodes, have as many non-zero entries as required to guarantee the regularity[4] of the d nodes and the necessary number of s_A nodes. This results in matrices M_1 and M_2 having a special degree distribution which is described by means of the notation $(y; 1, x)$ and the parameter t, where y represents the degree of the d nodes, t is the number of syndrome nodes that are forced to have degree 1 (they become s_A nodes), and x represents the degree of the remaining syndrome nodes, referred to as s_B nodes.

Given the particular structure of the M_1 and M_2 matrices and the number of different types of nodes that are present in the factor graph shown in Figure 6.1, the sum-product decoding of these quantum LDGM CSS codes becomes relatively nuanced. The reason for deriving a complex graph like the one shown in Figure 6.1 is that decoding over it is different to decoding over the graph associated to the matrix $\mathbf{H}_Q$ that results from the product shown in (6.4). This occurs because the matrix products $\mathrm{M}_1\tilde{\mathrm{H}}$ and $\mathrm{M}_2\tilde{\mathrm{G}}$ eliminate some edges and introduces more cycles, which negatively impacts the performance of the SPA [183]. This is is similar to what happens for serial concatenated LDGM schemes in classical error correction [184]. Therefore, although decoding over the graph associated to the set of matrix products shown in Figure 6.1 is more complex than doing so over the factor graph associated to the final matrix $\mathbf{H}_Q$, it is worth doing so for

[4]Regularity in this context implies that all the d nodes have the same degree, i.e, that they are all connected to the same number of s nodes.

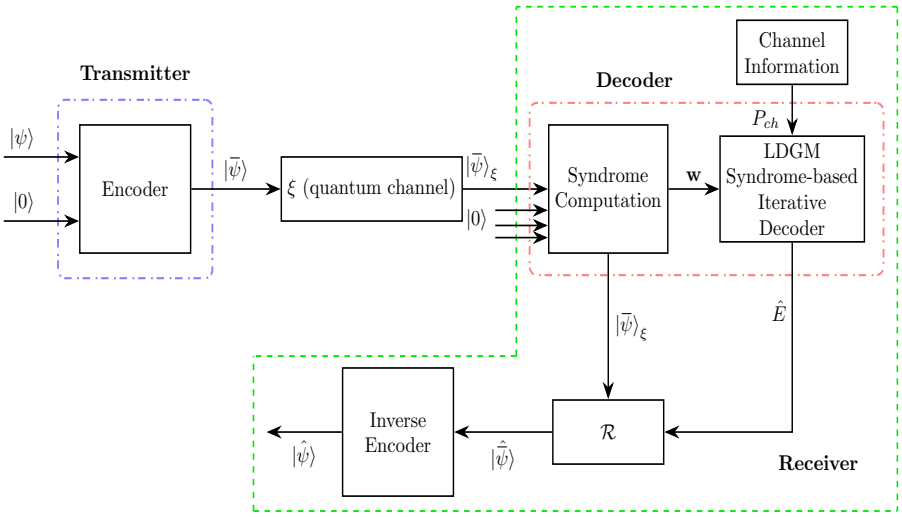


Figure 6.2: *Schematic of a quantum communication system using QLDGM codes. At the transmitter, a QLDGM code with QPCM* $\mathbf{H}_Q$ *maps the quantum state (logical qubits)* $|\psi\rangle \in \mathcal{H}_2^{\otimes k}$ *onto the codeword (physical qubits)* $|\bar{\psi}\rangle \in \mathcal{H}_2^{\otimes N}$ *by using* $(N-k)$ *ancilla qubits (denoted in the figure by* $|0\rangle$*). At the receiver, the noisy codeword* $|\bar{\psi}\rangle_\xi = E\,|\bar{\psi}\rangle$ *is received, where* $E \in \mathcal{G}_N$ *is the error inflicted by the quantum channel. The quantum state* $|\bar{\psi}\rangle_\xi$ *is processed by the syndrome computation block at the decoder to compute its error syndrome* $\mathbf{w}$*. The ancilla qubits at the input of the syndrome computation block are necessary to physically implement the syndrome calculation. Together with the a priori channel information* P_{ch}*,* $\mathbf{w}$ *is provided to the syndrome based LDGM decoder, which yields an estimate of the original error pattern* $\hat{E}$*. The recovery operator* $\mathcal{R}$ *uses* $\hat{E}$ *to correct the noisy codeword. Finally, the inverse encoder yields an estimate of the logical qubits* $|\hat{\psi}\rangle$ *from the corrected codeword* $|\hat{\bar{\psi}}\rangle$*.*

the sake of performance. In [176], a technique known as Discretized Density Evolution (DDE) [114] is applied to optimize the design of quantum LDGM CSS codes and a complete description of how the decoding process unfolds over the graph shown in Figure 6.1 is provided.

6.3 DESIGN OF NON-CSS LDGM-BASED CODES

We know from the previous section that LDGM-based CSS codes can be decoded over two separate (sub)graphs like the ones shown in Figure 6.1. This is made possible by the specific nature of the quantum PCMs of CSS codes (6.2), and is visible on a CSS decoding graph by the fact that any given s node can only be connected to either d_x or d_z nodes: a subset of s nodes is used to decode the X operators (the X component of the symplectic representation of the error sequence $\mathbf{e}_x$) and another subset is used to decode the Z operators (the Z component of the symplectic representation of the error sequence $\mathbf{e}_z$).

The main appeal of non-CSS codes is their ability to exploit redundancy more efficiently than CSS schemes. In our proposed non-CSS construction, we achieve this by allowing edges from a given s node to go to both d_x and d_z nodes. The first method that comes to mind to implement this idea is to randomly distribute the edges of the upper layer of the graph in a manner that ensures that s nodes are connected to both d_x and d_z nodes. However, attempting to decode the X and Z parts over a decoding graph with s nodes whose edges have been haphazardly assigned to both d_x and d_z nodes will cause numerous decoding problems. For instance, not defining a specific distribution for these edges inadvertently causes a reduction in the number of s_A nodes, and not limiting their total number causes a reduction in the values of the log-likelihood messages exchanged in the decoding process, which severely degrades the decoding performance. Therefore, it is important to optimally design the upper layer of the decoding graph when constructing a non-CSS QLDGM based code. Devising a proper way of distributing the connections among s, d_x, and d_z nodes in the decoding graph is paramount to construct good non-CSS LDGM-based quantum codes.

6.3.1 PROPOSED PROCEDURE FOR THE CONSTRUCTION OF NON-CSS QLDGM CODES

We begin the non-CSS design process by using a CSS quantum code based on classical LDGM codes [21, 22, 155, 176] as the starting point. For the sake of simplicity and comparison continuity, we maintain the requirements enforced in [155, 176]: the matrices used to perform linear row operations are equal to each other $\mathrm{M}_1 = \mathrm{M}_2 = \mathrm{M}$, and the degree distribution of $\mathrm{P}^\top$ and P is the same.

The CSS QLDGM code used as a starting point will be associated to two separate decoding subgraphs, one for H and the other for G. The upper layers of these subgraphs (the number and degree distribution of the d, s_A, and s_B nodes) will be defined by two identical matrices M of size $m \times \frac{N}{2}$ described by $(y; 1, x)$ which have t rows with a single non-zero entry.

We can build our non-CSS scheme using two different strategies. Given that both of them involve very similar procedures, we begin by explaining the simplest construction method. Then, we will present the second proposed design technique.

6.3.1.1 Method 1: Syndrome node combination

Non-CSS codes based on the first strategy are constructed as follows:

1. First, generate a new matrix, M_d, as

$$\mathrm{M}_d = \begin{pmatrix} \mathrm{M}_{m\times\frac{N}{2}} & 0_{m\times\frac{N}{2}} \\ 0_{m\times\frac{N}{2}} & \mathrm{M}_{m\times\frac{N}{2}} \end{pmatrix}_{2m\times N}. \tag{6.8}$$

2. Select q nodes out of the $2t$ s_A nodes of matrix M_d[5], which we will refer to as s_C nodes, and add an edge from these nodes to the d nodes on the side of the decoding graph they are not connected to. We apply a criterion to ensure these new connections are not made randomly: *the edges added to the q selected s_A nodes can only be made to a* d *node (*d_x *or* d_z*) that is a* d_A *node.* We define d_A nodes as any d nodes that are connected to an s_A node. Of the q s_C nodes, half

[5]Note that M_d, as defined in (6.8), is the matrix representation of the upper layer of the graph in Figure 6.1.

of them proceed from s_A nodes in the CSS subgraph used to decode the X operators, while the other half come from s_A nodes in the CSS subgraph used to decode the Z operators. Figure 6.3 illustrates how an s_C node is generated.

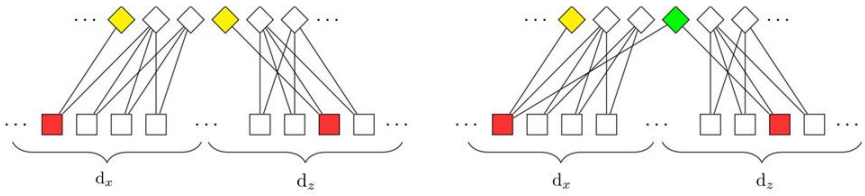


Figure 6.3: *Generation of an s_C node. The upper nodes represent the syndrome nodes while the bottom nodes represent the* d *nodes (d_x and d_z denote the* d *nodes associated to each of the separate CSS decoding subgraphs). The s_A nodes are represented in yellow, the d_A nodes are shown in red, and the s_C node is pictured in green.*

The reasoning behind adding edges that traverse the X and Z sides of the graph only to s_A nodes is based on the following considerations: First, transforming an s_A node into an s_C node implies that the new node no longer provides perfect syndrome information, given that it is now connected to two d nodes. However, the fact that an s_C node only has two edges implies that its syndrome information, although not transferred exactly, still has high impact when computing messages for associated nodes. At the same time, the edge that traverses from the s_C node to the other side of the factor graph (either d_x or d_z) reaches a d_A node. Considering that messages from d_A nodes are more likely to be correct (they are connected to an s_A node), coupled with the fact that s_C node syndrome information still plays an important role in the messages that the node computes, it is reasonable to assume that s_C nodes relay accurate information and that they behave in a similar manner to s_A nodes. Therefore, s_C nodes provide a way in which reliable messages can be exchanged between both sides of the factor graph, which should have a positive impact on decoding and improve performance.

It is important to note that if we were to add a cross-graph edge to an s_B node, because of its high degree, the messages received over this new edge would play a limited role in the computations made by the node. By association, the cross-graph messages transmitted over the new edge would also have a very limited effect on the computation of messages exchanged on the other side of the graph and little performance improvement, if any, would be obtained.

Notice that at this stage we have transformed M_d into a new matrix M'_d, modifying the upper layer of the original CSS decoding graph of Figure 6.1 in the following manner:

- There are q s_C nodes that are connected to both sides of the graph.
- Some d nodes are connected to both s_A and s_C nodes.

These modifications force the s and d nodes of the non-CSS decoding graph to have a somewhat irregular edge distribution. Indeed, the "regularity" of the d nodes has been violated in order to connect the separate CSS decoding subgraphs, resulting in $\frac{q}{2}$ d_x nodes and $\frac{q}{2}$ d_z nodes having an additional edge. Furthermore, q s nodes now have two edges, one of them directed towards a d_x node and the other towards a d_z node.

It is intuitive to think that the performance of this novel non-CSS structure should at least be as good, if not better, than that of the CSS scheme utilized as a starting point, provided that the parameter q is chosen properly. If we select $q << m$, the decrease in the number of s nodes providing perfect information will be small and should have negligible impact in the decoding process[6]. On the contrary, the degree-2 s_C nodes allow the exchange of information between both sides of the graph as the iterative decoding process progresses, potentially improving the decoding performance. Therefore, we expect this scheme to present its best performance for a specific range of small values of q, with deterioration occurring when q is increased beyond this range.

[6]A total of q s_A nodes get converted into s_C nodes, which do not provide perfect information.

6.3.1.2 Method 2: Syndrome node combination + removal of s_A nodes

The second design technique removes q syndrome nodes from the decoding graph generated by the first method, specifically the q s_A nodes connected to a d_A node that is linked to an s_C node. The process is shown in Figure 6.4. Notice that with this construction the regularity of the d nodes is maintained and the data rate of the code is increased[7]. Moreover, as long as the number of removed s_A nodes is not too large, the impact on decoding should be minimal: although s_C nodes do not provide "perfect" information, much of the reliability of the messages they transmit to the corresponding d (previously d_A) nodes will be kept, as they only have two edges. This should mitigate any impact on performance. Because method 2 requires the removal of some s_A nodes, compared with the CSS code used as a starting point, a larger amount of s_A nodes may be necessary in the non-CSS structure to ensure acceptable performance as the value of q grows. This hints towards an increased importance of applying *doping* (increasing the value of t) to the constituent matrices M.

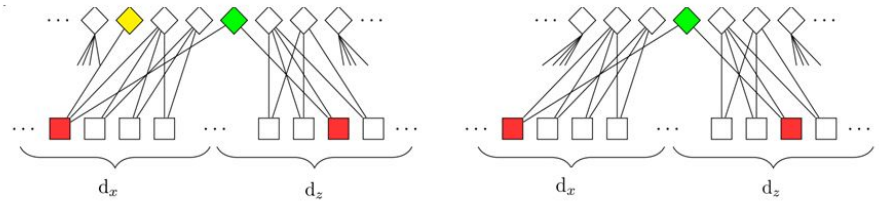


Figure 6.4: *Removal of an s_A (yellow) node that was previously used to generate an s_C (green) node.*

Once the specific s_A nodes are removed, we obtain a new matrix M'_d of size $(2m - q) \times N$, which has the following characteristics:

- It has $2(t - q)$ rows with a single nonzero entry. In terms of the decoding graph, this means that there are $2(t - q)$ s_A nodes.

[7] By eliminating syndrome nodes we are decoding the same number of qubits using less syndrome information.

- There are q rows with two nonzero entries. The first entry must be placed in any of the first $\frac{N}{2}$ columns of the matrix while the second one must be placed in any of the last $\frac{N}{2}$ columns. This ensures that s_C nodes connect both sides of the decoding graph.
- The other $2(m-t)$ rows have x nonzero entries. In the decoding graph, these rows correspond to the s_B nodes, which remain the same as in the CSS structure used as a starting point.

6.3.1.3 Non-CSS QPCM

The quantum PCM of the proposed non-CSS code obtained using either of the aforementioned methods is calculated as

$$\begin{aligned}\mathbf{H}_{Q_{\mathrm{nonCSS}}} &= \mathrm{M}'_d\mathbf{H}_{\mathrm{CSS}} = \mathrm{M}'_d(\mathrm{H}_x|\mathrm{H}_z)\\ &= \mathrm{M}'_d\begin{pmatrix}\tilde{\mathrm{H}} & 0\\ 0 & \tilde{\mathrm{G}}\end{pmatrix} = (\mathrm{H}''_x|\mathrm{H}''_z),\end{aligned} \tag{6.9}$$

where $\mathbf{H}_{\mathrm{CSS}}$ is defined as in (6.2), $\tilde{\mathrm{H}}$ and $\tilde{\mathrm{G}}$ are the parity check and generator matrices of a classical LDGM code, and M'_d is obtained using any of the construction methods.

As shown below, the construction in (6.9) satisfies the symplectic criterion given in (4.24). Assume that M'_d is obtained based on the second construction method presented above. With the goal of simplifying the proof, we write $[\mathrm{M}'_d]_{m_r\times N}$ as the concatenation of two sub-matrices, i.e., $[\mathrm{M}'_d]_{m_r\times N} = [\mathrm{M}'_{\alpha_{m_r\times\frac{N}{2}}}\mathrm{M}'_{\gamma_{m_r\times\frac{N}{2}}}]$. Substituting this expression into (6.9), we obtain

$$\begin{aligned}\mathbf{H}_{Q_{\mathrm{nonCSS}}} &= (\mathrm{H}''_x|\mathrm{H}''_z)\\ &= [\mathrm{M}'_d]_{m_r\times N}\mathbf{H}_{\mathrm{CSS}_{N\times 2N}}\\ &= [\mathrm{M}'_{\alpha_{m_r\times\frac{N}{2}}}\mathrm{M}'_{\gamma_{m_r\times\frac{N}{2}}}]\begin{pmatrix}\tilde{\mathrm{H}}_{\frac{N}{2}\times N} & 0_{\frac{N}{2}\times N}\\ 0_{\frac{N}{2}\times N} & \tilde{\mathrm{G}}_{\frac{N}{2}\times N}\end{pmatrix}\\ &= \left([\mathrm{M}'_{\alpha_{m_r\times\frac{N}{2}}}\tilde{\mathrm{H}}_{\frac{N}{2}\times N}]_{m_r\times N} \;\middle|\; [\mathrm{M}'_{\gamma_{m_r\times\frac{N}{2}}}\tilde{\mathrm{G}}_{\frac{N}{2}\times N}]_{m_r\times N}\right),\end{aligned}$$

where $m_r = 2m - q$ is the total number of rows of M'_d.

Since for an LDGM code $\tilde{\mathrm{G}}\tilde{\mathrm{H}}^T = \tilde{\mathrm{H}}\tilde{\mathrm{G}}^T = 0$, when checking the symplectic criterion, we obtain

$$\begin{aligned}
\mathrm{H}''_x\mathrm{H}''^{\top}_z \oplus \mathrm{H}''_z\mathrm{H}''^{\top}_x = & \\
& \Big([\mathrm{M}'_{\alpha_{m_r\times\frac{N}{2}}} \tilde{\mathrm{H}}_{\frac{N}{2}\times N}][\mathrm{M}'_{\gamma_{m_r\times\frac{N}{2}}} \tilde{\mathrm{G}}_{\frac{N}{2}\times N}]^{\top} \\
& \oplus [\mathrm{M}'_{\gamma_{m_r\times\frac{N}{2}}} \tilde{\mathrm{G}}_{\frac{N}{2}\times N}][\mathrm{M}'_{\alpha_{m_r\times\frac{N}{2}}} \tilde{\mathrm{H}}_{\frac{N}{2}\times N}]^{\top} \Big) \\
& = \Big(\mathrm{M}'_{\alpha_{m_r\times\frac{N}{2}}} \underbrace{\tilde{\mathrm{H}}_{\frac{N}{2}\times N}\tilde{\mathrm{G}}^{\top}_{N\times\frac{N}{2}}}_{0_{\frac{N}{2}\times\frac{N}{2}}} \mathrm{M}'^{\top}_{\gamma_{\frac{N}{2}\times m_r}} \\
& \oplus \mathrm{M}'_{\gamma_{m_r\times\frac{N}{2}}} \underbrace{\tilde{\mathrm{G}}_{\frac{N}{2}\times N}\tilde{\mathrm{H}}^{T}_{N\times\frac{N}{2}}}_{0_{\frac{N}{2}\times\frac{N}{2}}} \mathrm{M}'^{\top}_{\alpha_{\frac{N}{2}\times m_r}} \Big) \\
& = 0_{m_r\times m_r}
\end{aligned}$$

proving that $\mathbf{H}_{\mathcal{Q}_{\mathrm{nonCSS}}}$ satisfies the symplectic criterion.

6.3.1.4 Mixture of both methods

Another possibility to design non-CSS codes is to remove only a fraction of the s_A nodes that are used to generate s_C nodes in the first construction method. This procedure is identical to the second technique, with the sole exception that instead of removing the entire subset of q s_A syndrome nodes involved in the generation of the s_C nodes, only $l < q$ s_A nodes are removed. In the following section, we focus on codes obtained using the first two methods. Optimizing the performance of codes derived using this third approach may be of interest in future work.

6.3.2 DECODING NON-CSS QLDGM CODES

Independently of the design method, decoding of the novel non-CSS quantum codes is performed utilizing the sum-product algorithm over the factor graph defined by the product $\mathrm{M}'_d \times \mathbf{H}_{\mathrm{CSS}}$. Message passing will differ from that performed at a CSS QLDGM decoder, in the sense that both sides of the graph interact during the decoding process (messages are

exchanged between nodes on the left and right subsections of the graph). This occurs because of the modified upper layer in the decoding graph of the non-CSS code, as shown in Figure 6.5 for a non-CSS code derived using the first design method.

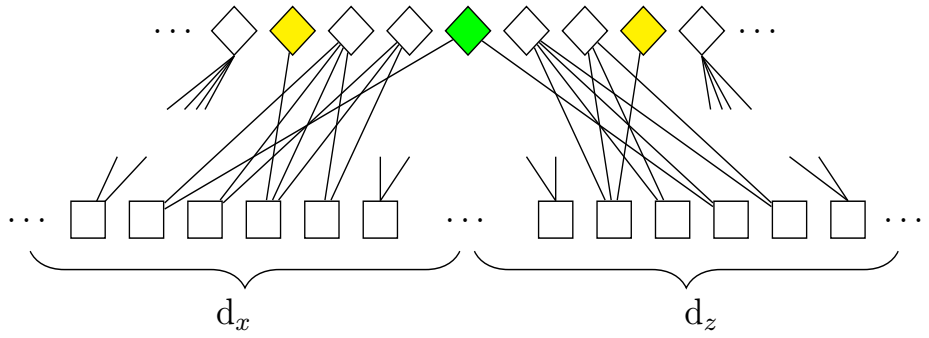


Figure 6.5: *Upper layer of the decoding graph associated to a non-CSS code obtained using method 1. The upper nodes represent the syndrome nodes, while the bottom nodes represent the* d *nodes (*d_x *and* d_z *denote the* d *nodes associated to each of the original separated CSS decoding subgraphs). The* s_A *nodes are represented in yellow and the* s_C *nodes are pictured in green.*

6.3.3 RATE CONSIDERATIONS

Both design methods allow a high degree of flexibility in terms of selecting the rate of the non-CSS quantum code. In fact, different non-CSS codes of the same rate can be obtained depending on the selected design method. Consider an arbitrary non-CSS code of quantum rate R_Q obtained based on the first design method taking as a starting point a CSS code of the same rate. A different non-CSS code of rate R_Q can be obtained by using the second design method, taking as a starting point a CSS code of lower rate. In fact, the second construction technique allows us to build multiple matrices M'_d of equal size (which will lead to codes of equal quantum rate) by starting from different matrices M (and thus from CSS codes of different rates) and varying the parameter q, which represents number of s_A nodes that are removed. For instance, we could design a matrix M'_{d_1}

of size $(2m_1 - q_1) \times N$ using two matrices M_1 of size $m_1 \times \frac{N}{2}$, and construct a second matrix M'_{d_2} of size $(2m_2 - q_2) \times N$ using two matrices M_2 of size $m_2 \times \frac{N}{2}$. If $2m_1 - q_1 = 2m_2 - q_2$, then both codes will have the same rate. Therefore, when the second method is utilized, code optimization depends on the choice of matrix M_d, as well as parameters m and q.

Notice that for a fixed $m \times N$ matrix M, by using the second design method and varying the value of q, we obtain different matrices M'_d of size $(2m-q) \times 2N$. This will result in non-CSS codes that encode $N-(2m-q)$ logical qubits into N physical qubits, and thus their rate will be

$$R_{Q,\text{non-CSS}} = \frac{N-(2m-q)}{N}. \tag{6.10}$$

Since $q > 0$, $R_{Q,\text{non-CSS}}$ is always higher than the rate of the CSS QLDGM code used as a starting point, which is given by

$$R_{Q,\text{CSS}} = \frac{N-2m}{N}. \tag{6.11}$$

Notice that if the non-CSS code obtained from method 2 maintains the same performance as the original CSS code, this will be achieved with a higher rate. We introduce the parameter R_I, defined as

$$R_I = R_{Q,\text{non-CSS}} - R_{Q,\text{CSS}} = \frac{q}{N-2m}, \tag{6.12}$$

to quantify the rate increase provided by the non-CSS scheme derived using the second construction method when compared to the original CSS scheme. This rate increase is determined by the value of q, which for the second design method represents the number of s_C nodes in the non-CSS decoding graph, as well as the number of s_A nodes removed from the decoding graph of the CSS code used as a starting point. The value of q will influence the performance of the resulting non-CSS code: intuitively, large increases in its value should lead to worsened performance, as the doping effect is reduced, but $q > 0$ allows for information exchange between the left and the right sides of the decoding graph, which should have a positive effect on performance. The impact of q on the proposed schemes is studied in the following section.

6.4 SIMULATION RESULTS

In this section we compare the performance of the proposed non-CSS codes to that of the CSS codes of [155] and [176] when they are used over

the i.i.d. X/Z channel and the depolarizing channel. The CSS codes in [155] and [176] have rate $R_Q \approx \frac{1}{4}$ and block length $N = 19014$, encoding $k = 4752$ qubits into N qubits. Matrix P, of size 9507×9507, has the same degree distribution as its transpose $\mathrm{P}^\top$, and corresponds to a rate $\frac{1}{2}$ classical LDGM code. Hence, both $\tilde{\mathrm{G}}$ and $\tilde{\mathrm{H}}$ have size 9507×19014. Matrix M, which is used to transform $\tilde{\mathrm{G}}$ and $\tilde{\mathrm{H}}$ into G and H, is full-rank, low-density, and has size 7131×9507. Results are depicted in terms of either the Qubit Error Rate (QBER) or the Word Error Rate (WER). QBER represents the fraction of qubits that experience an error, while WER is the fraction of transmitted blocks that have at least one qubit error. We use the QBER metric for some of our simulations because it can be estimated with high confidence faster than the WER, which is helpful in shortening the required simulation time of some of our codes. Readers should refer to Appendix C for a more complete explanation on how these simulations have been conducted.

First, the codes are simulated over the i.i.d. X/Z channel model of [18], where Z and X errors are modeled as independent events identically distributed according to the flip probability f_m. We begin by using as a starting point the family of CSS codes of the first proposed structure in [155] and [176], which are individual regular LDGM codes. The simplicity of the channel model and of the code structure allow us to assess, in a rapid and efficient manner, the values of q that optimize the performance of the non-CSS construction. Using these values of q, we will repeat the simulations when the CSS codes used as the starting point consist of the parallel concatenation of regular LDGM codes, as described in [155, 176]. Both design methods are utilized to obtain the resulting non-CSS codes. Finally, we repeat the same process for the depolarizing channel.

6.4.1 I.I.D. X/Z CHANNEL - NON-CSS CODES BASED ON INDIVIDUAL REGULAR LDGM CODES

For these simulations, the CSS code utilized as a starting point is an individual regular LDGM code. Matrix P is generated pseudorandomly and corresponds to a regular (X, X) LDGM code. M is characterized by the parameter values[8] $\mathrm{M}(3; 1, 8.72)$ and $t = 4361$. The degrees[9] of P (and

[8]The fractional number 8.72 represents the fact that 72% of the s_B nodes will have degree 9 while 28% of them will have degree 8.

[9]Given that the degree distribution of $\mathrm{P}^\top$ and P is the same, we refer to them indistinctly throughout this paper.

$\mathrm{P}^{\top}$) are varied between $(9, 9)$ and $(13, 13)$. In the figures that follow, f_m is the probability of error in each separate X and Z error channels and the analytical error floors of the LDGM codes have been obtained as shown in [183].

6.4.1.1 Non-CSS codes derived using method 1

As explained before, non-CSS codes based on this method are obtained by transforming q s_A nodes of the CSS decoding graph into degree-2 s_C nodes. All the non-CSS codes obtained in this manner have the same quantum rate ($R_Q = \frac{1}{4}$). Matrix M_d is built from M as shown in (6.8). The rest of the underlying components of the non-CSS configuration are identical to those of the CSS designs[10]. We test three different configurations, $q = [100, 750, 1500]$, for different degrees of the P matrices. The simulation results, which are shown in Figure 6.6 reflect how for $q = 100$ and $q = 750$, regardless of the degree of the pseudorandom matrix P, the performance of the proposed non-CSS codes is slightly better than that of the CSS schemes used as a starting point. This performance improvement is most notable for the case of $\mathrm{P}(13, 13)$, where, as f_m increases, the non-CSS schemes operate closer than the original CSS code to the analytical error-floor. However, for $q = 1500$, the performance of the non-CSS code is worse than that of the CSS scheme. The best results are observed when the non-CSS code is designed using $q = 750$.

6.4.1.2 Non-CSS codes derived using method 2

Earlier we saw how non-CSS codes derived using method 2 are obtained by removing q syndrome nodes from the decoding graph generated by the first method, specifically the q s_A nodes connected to a d_A node that is linked to an s_C node. Note that removing syndrome nodes will result in codes with different quantum rates. As previously, we test the values $q = [100, 750, 1500]$ for different degrees of the P matrices. The results are shown in Figure 6.7.

The results displayed in Figure 6.7 are once again consistent regardless of the degree of P. They show how for the two smaller q values, 100 and 750, the non-CSS codes yield the same performance as the CSS schemes of

[10]The only difference between the CSS and non-CSS codes lies in matrix M'_d. The rate $\frac{1}{2}$ classical LDGM code and, therefore, matrices P, $\tilde{\mathrm{G}}$, and $\tilde{\mathrm{H}}$, are identical.

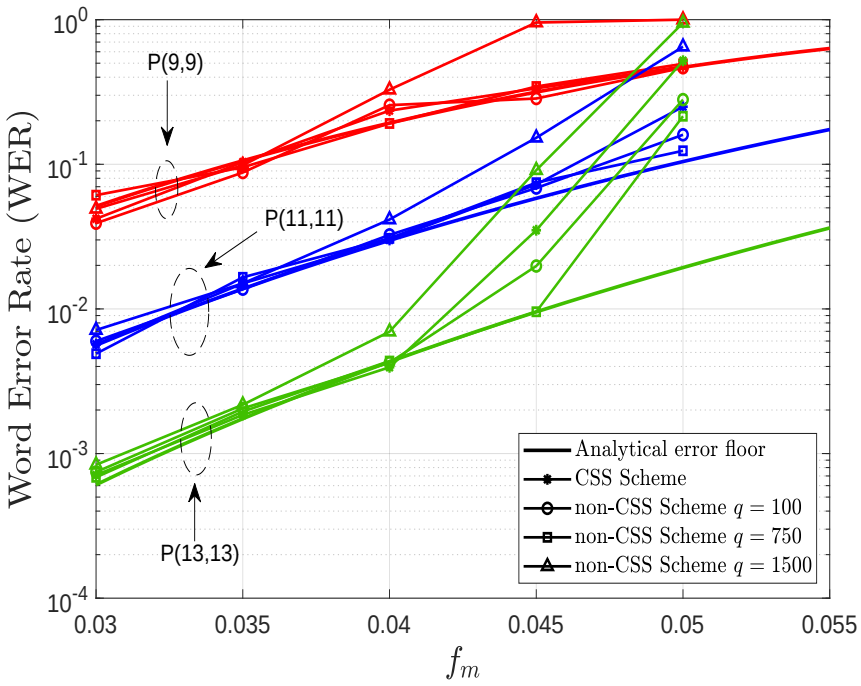


Figure 6.6: *Simulated WER for non-CSS QLDGM codes based on individual regular LDGM codes obtained using the first design method when they are applied over the flipping channel. f_m is the probability of error in each separate X and Z error channel.*

[155]. For $q = 1500$, the performance of the non-CSS codes is significantly worse. For instance, at a value of $f_m = 0.04$, the word error rate for the $q = 1500$ schemes is around an order of magnitude higher than for the $q \leq 750$ non-CSS schemes. This corroborates our intuition that there is an optimum range of values for q, and that utilizing values outside of that range degrades the code performance. Based on the results in Figure 6.7, the optimal value of q should be a small percentage of the total number of syndrome nodes m, $q < 0.1 \times m$ ($\frac{1500}{14262} \approx 0.1$).

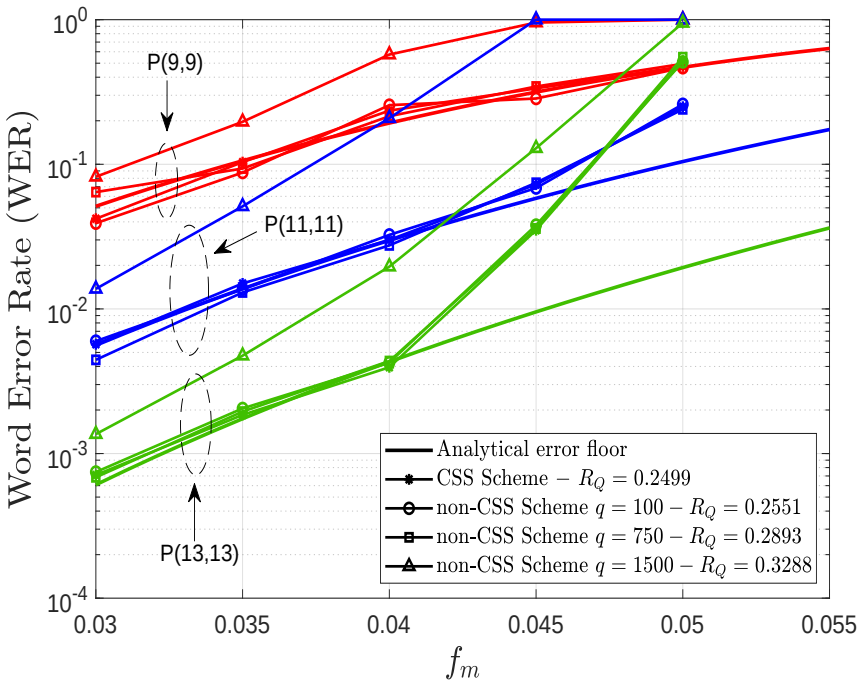


Figure 6.7: *Simulated WER for non-CSS QLDGM codes based on individual regular LDGM codes obtained using the second design method when they are applied over the flipping channel. f_m is the probability of error in each separate X and Z error channel.*

Table 6.1 presents the WER performance for the codes considered in Figure 6.7 when matrix P has degrees $(13, 13)$ and $f_m = 0.03$. The CSS scheme has rate $R_{Q-\text{CSS}} = \frac{19014-2\times7131}{19014} = 0.2499$. Results for the $q = 1500$ non-CSS scheme have not been included, since the performance of the original CSS scheme is better. As with the first method, the best result here is obtained for the $q = 750$ non-CSS scheme, which achieves similar performance to that of the CSS code with an approximately 15% higher rate ($R_I \approx 0.158$).

Overall, the results obtained for both design methods illustrate that our proposed non-CSS codes, regardless of the design method, outperform

QLDGM CSS codes when individual regular LDGM codes are utilized. The first design method provides a way to obtain non-CSS codes that outperform CSS codes of the same rate. The second method enables us to construct non-CSS codes with error correcting capabilities comparable to those of lower rate CSS schemes. Therefore, to design non-CSS codes of a fixed rate, we could apply the first design method to a CSS code of the same rate, or we could start with a CSS code of lower rate and apply the second method. In this case, we should carefully chose the rate of the original CSS code to obtain the desired rate in the designed non-CSS code.

Table 6.1: *Comparison between the codes shown in Figure 6.7. The WER data included in the table corresponds to the codes whose* P *matrix has degrees* $(13, 13)$.

Code Type	R_Q	q	WER @ $f_m = 0.03$
CSS	0.2499	-	$7.12 \times 10^{-}3$
non-CSS	0.2551	100	$7.26 \times 10^{-}3$
non-CSS	0.2893	750	$7.13 \times 10^{-}3$

6.4.2 I.I.D. X/Z CHANNEL - NON-CSS CODES BASED ON THE PARALLEL CONCATENATION OF LDGM CODES

As mentioned in section 6.1.4, regular LDGM codes used in classical channels present error floors. Fortunately, these error floors can be substantially lowered if we use the parallel concatenation of two regular LDGM codes. In [21], [22] CSS quantum codes based on the use of single regular LDGM codes were shown to also exhibit error floors. Inspired by the good performance displayed by parallel LDGM codes in the classical domain, a scheme based on the parallel concatenation of LDGM codes was designed and applied to the i.i.d. X/Z channel in [155] and [176]. Similar to the classical scenario, CSS codes built based on the parallel structure display lower error floors and better performance overall.

We now repeat the study carried out in the previous subsection, but using the parallel concatenation of regular LDGM codes as the starting point

to derive our proposed non-CSS codes. In [176], varios parallel[11] LDGM structures P$[(y_1, y_1); (y_2, z_2)]$ were employed. For our analysis we use the structure with the lowest degrees that appears in [176]: P$[(8, 8); (3, 60)]$. Although performance is better when the degrees of the second code (y_2, z_2) are larger, codes with smaller values for these degrees require less processing and simulation time. To ease simulation requirements even further, results in this subsection are presented in terms of the QBER. The best configuration for P$[(y_1, y_1); (y_2, z_2)]$ in [176] will be used later for the simulations over the depolarizing channel.

As we did for individual regular LDGM codes in the previous subsection, we analyze the performance of the non-CSS codes obtained using the two design methods proposed in section 6.3. The CSS code used as a starting point is the same for both methods, and utilizes the same matrix M as in the scenario for individual regular LDGM codes: $\mathrm{M}_{7131\times 9507}(3; 1, 8.72)$ and $t = 4361$. We consider the values of $q = \{100, 500, 1000, 1500\}$. As before, f_m is the probability of error in each separate X and Z error channel and the analytical error floors of the LDGM codes have been obtained as shown in [183].

Figure 6.8 presents the performance of the $R = \frac{1}{4}$ original CSS code and of the $R = \frac{1}{4}$ non-CSS codes obtained by applying the first design method. As shown in the figure, the non-CSS codes derived by selecting $q = 100$ and $q = 500$ outperform the original CSS structure. For $q > 500$, performance of the non-CSS codes gradually deteriorates, with the result for $q = 1500$ being substantially worse than that of the original CSS code.

Figure 6.9 shows the results for the non-CSS codes derived using the second proposed design method. The curves shown in this figure portray how the performance of the non-CSS codes is drastically degraded as we increase the value of q. This effect is much more noticeable than for codes based on a single regular LDGM code. In fact, the only value for which the non-CSS configuration based on parallel concatenation matches the performance of the original CSS scheme is $q = 100$, whereas in Figure 6.7 we could see that schemes based on individual regular LDGM codes matched the performance of the original CSS code at least up to $q = 750$. In essence, although the s_A node elimination step explained in section 6.3.1

[11] The notation P$[(y_1, y_1); (y_2, z_2)]$ indicates the degree distributions of the constituent regular LDGM codes utilized in the parallel concatenation. In [176], the parameters of the second code were typically chosen as $z_2 = 20y_2$.

still yields a small benefit, this is much lower than for the case of single regular LDGM codes.

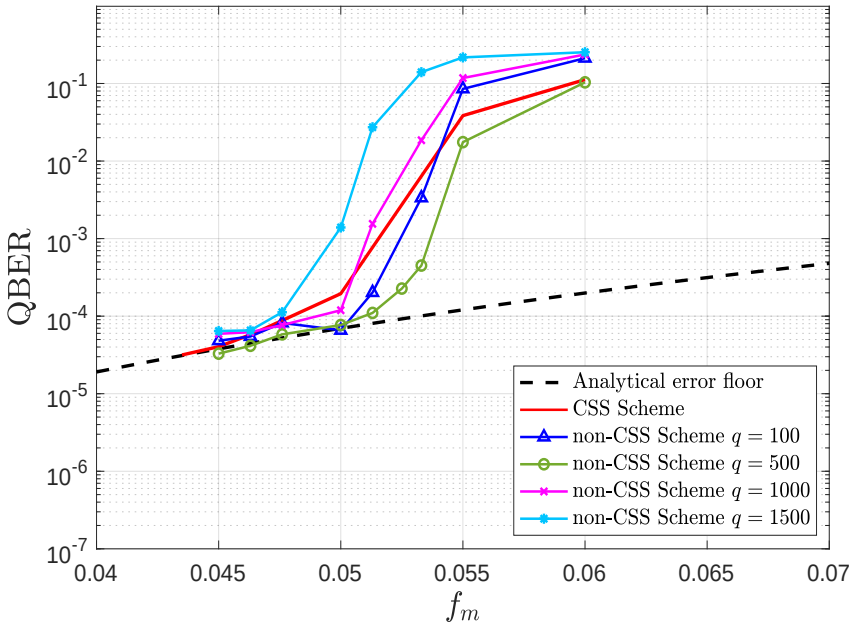


Figure 6.8: *Simulated QBER for a $R_Q = \frac{1}{4}$ CSS code and non-CSS codes of $R_Q = \frac{1}{4}$ code derived using the first design methodology. The underlying classical LDGM code is the same for all the codes and has degree distribution* $\mathrm{P}[(8,8);(3,60)]$. *f_m is the probability of error in each separate X and Z error channel.*

Throughout this chapter we have mentioned that the second non-CSS code design methodology can be used not only to obtain higher rate non-CSS codes with performance similar to that of lower rate CSS codes, but also to generate different non-CSS codes of a fixed rate by varying the CSS codes used as starting points and selecting the appropriate value of q. Analyzing the performance of $R_Q = \frac{1}{4}$ non-CSS codes obtained in this manner and comparing the results to those obtained using the first design method will allow us to determine which design technique yields codes with better performance.

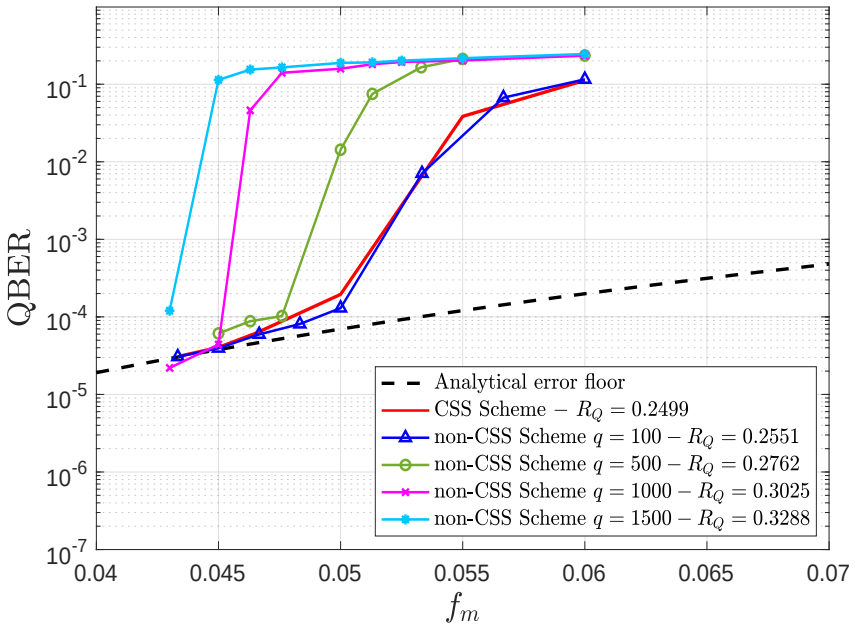


Figure 6.9: *Simulated QBER for a $R_Q = \frac{1}{4}$ CSS code and non-CSS codes of different rate derived using the second design methodology. The underlying classical LDGM code is the same for all the codes and has degree distribution* $\mathrm{P}[(8,8);(3,60)]$. *f_m is the probability of error (iid) in each separate X and Z error channel.*

This comparison is shown in Figure 6.10, where various $R_Q = \frac{1}{4}$ non-CSS codes obtained using the second design method are compared to the best $R_Q = \frac{1}{4}$ code generated by the first design method ($q = 500$ in Figure 6.8) and to the original $R_Q = \frac{1}{4}$ CSS structure. The non-CSS codes generated by method 2 are derived by building matrices M'_d using matrices M of different size, which are designed according to the analysis conducted in [176]. We introduce parameter q' to refer to the values of q used to derive these codes and to distinguish them from the code built using the first method.

The $q' = 100$ code uses $\mathrm{M}_{7181\times 9507}(3; 1, 8.56)$ with $t = 4361$ (using this M for a CSS QLDGM scheme would yield a code of rate $R_Q = 0.244$). The $q' = 500$ code uses $\mathrm{M}_{7381\times 9507}(3; 1, 8.36)$ with $t = 4561$ (using this M for a CSS QLDGM scheme would yield a code of rate $R_Q = 0.2236$). The $q' = 1000$ code uses $\mathrm{M}_{7631\times 9507}(3; 1, 8.27)$ with $t = 4761$ (using this M for a CSS QLDGM scheme would yield a code of rate $R_Q = 0.1973$).

Figure 6.10 shows that the non-CSS codes designed using the second method outperform the CSS scheme for all values of q'. However, the $q = 500$ non-CSS code designed using the first method is still better than any of the aforementioned codes, although the performance of the $q' = 100$ non-CSS code is only slightly worse.

In this subsection we have discussed results for the i.i.d X/Z Channel, where the X and Z operators are modelled independently according to the same distribution. Our analysis has allowed us to determine the design methodologies and the values of q that lead to the best performance. Making use of this knowledge, we will now consider performance over the depolarizing channel.

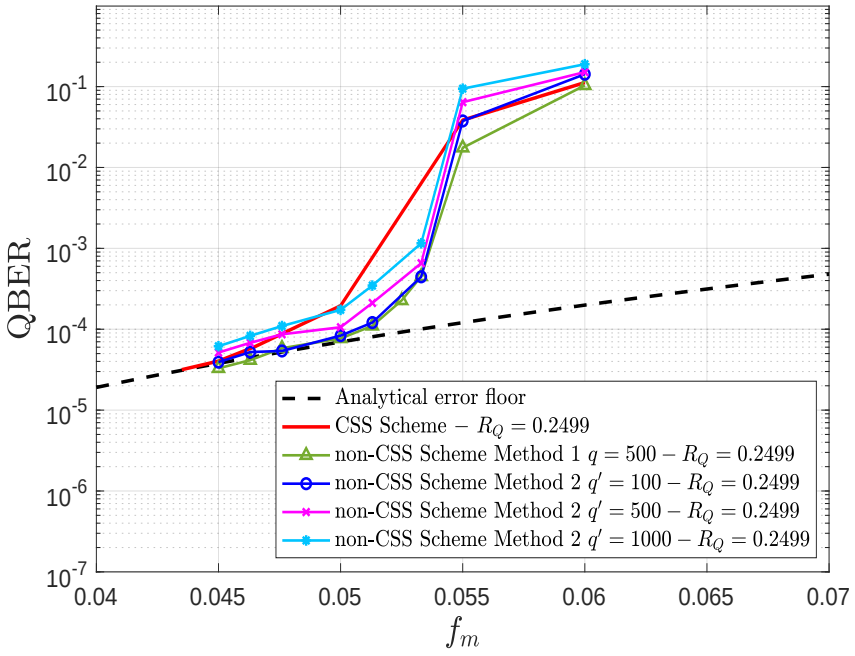


Figure 6.10: *Simulated QBER for a $R_Q = \frac{1}{4}$ CSS code and $R_Q = \frac{1}{4}$ non-CSS codes constructed using the two proposed design methods. q denotes how many s_A nodes have become s_C nodes in the upper layer of the decoding graph when method 1 is applied. q' denotes how many s_A nodes have become s_C nodes and how many s_A nodes have been removed from the upper layer of the decoding graph when method 2 is used. The underlying classical LDGM code is the same for all the codes and has degree distribution $\mathrm{P}[(8,8);(3,60)]$. f_m is the probability of error (iid) in each separate X and Z error channel.*

6.4.3 DEPOLARIZING CHANNEL

We now focus on the depolarizing channel introduced in section 6.1.1. We consider the best non-CSS codes obtained before for the i.i.d. X/Z channel: The $q = 500$ $R_Q = \frac{1}{4}$ non-CSS code obtained using the first design method and the $q = 100$ $R_Q = 0.255$ non-CSS code obtained using the second method. Figure 6.11 shows the QBER of the aforementioned codes for two different degree distributions of the parallel concatenated LDGM scheme. The curves associated to the original CSS codes are also included.

Similar to the results for the i.i.d. X/Z channel, Figure 6.11 portrays how the $R_Q = 0.255$ non-CSS codes display performance very close to that of the $R_Q = \frac{1}{4}$ CSS schemes. This phenomenon can be explained for both the depolarizing and the i.i.d. X/Z channels by the nature of the non-CSS construction, which adds a very small number of edges and removes very few syndrome nodes from the original CSS factor graph. As in the case of the X/Z channel model, the $q = 500$, $R_Q = \frac{1}{4}$ non-CSS codes designed utilizing method 1 also outperform the CSS code.

An important observation, which is reflected in Figure 6.11, is that the results are consistent regardless of the degrees (y_2, z_2) that are chosen. This is significant, since increasing y_2 and z_2 enables us to lower the error-floor of the QLDGM code. As shown in [155] and [176] for CSS codes, we can see from Figure 6.11 that selecting P_2 with larger degrees lowers the error floor but worsens the decoding threshold.

6.4.3.1 Distance to the theoretical limit

The most effective way to characterize the performance improvement of our proposed non-CSS codes is to measure their gap with respect to the Hashing bound. For this comparison, we will employ the design parameters that yield the best possible code. Such a scheme is obtained by using the first construction method with $t = 5000$, $q = 500$, $\mathrm{M}(3; 1, 11.04)$, and a parallel concatenated LDGM code of degree distribution $\mathrm{P}[(8, 8); (8, 160)]$.

Figure 6.12 depicts the performance of this non-CSS scheme as well as that of the CSS code used as a starting point for the design, which is the best CSS code proposed in [176]. The Hashing bound for $R_Q = \frac{1}{4}$ is also shown.

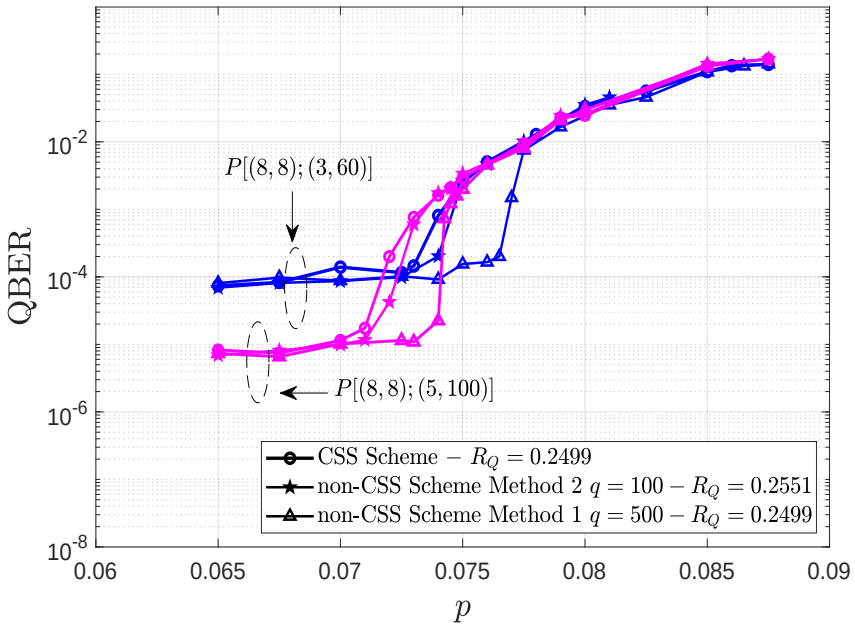


Figure 6.11: *Simulated QBER for three quantum codes: An* $R_Q = \frac{1}{4}$ *CSS QLDGM code, an* $R_Q = \frac{1}{4}$ *non-CSS QLDGM code obtained using the first proposed method and* $q = 500$*, and an* $R_Q = 0.255$ *non-CSS QLDGM code derived via the second design technique with* $q = 100$*. All three codes have been simulated for two different degree distributions of the underlying parallel concatenated LDGM scheme:* $\mathrm{P}[(8,8);(3,60)]$ *and* $\mathrm{P}[(8,8);(5,100)]$*.* p *is the depolarizing probability.*

We compute the distance to the Hashing bound δ as defined in (6.1), knowing that the the noise limit for $R_Q = \frac{1}{4}$ is $p^* \approx 0.127$, and taking $p_{\text{CSS}} = 0.0825$ and $p_{\text{non-CSS}} = 0.0865$ as the depolarizing probabilities at which the CSS and non-CSS codes enter the waterfall region, respectively. This yields $\delta_{\text{CSS}} = 1.873$ dB and $\delta_{\text{non-CSS}} = 1.668$ dB. In other words, the non-CSS scheme is about 0.2 dB closer to the Hashing bound. Thus, in

terms of overall performance over the depolarizing channel, the non-CSS codes proposed in this article outperform existing CSS techniques.

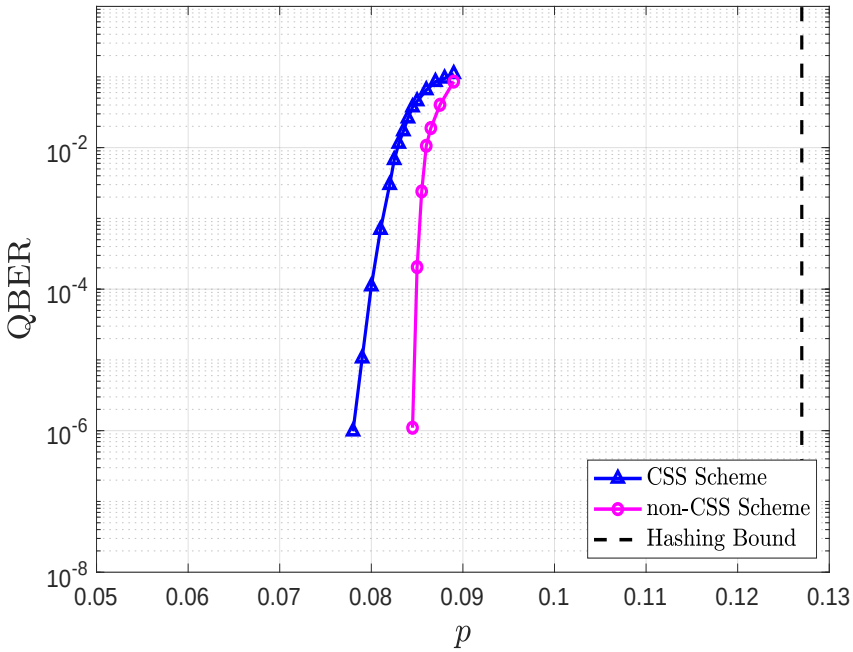


Figure 6.12: *Simulated QBER for the best $R = \frac{1}{4}$ CSS QLDGM code in [176] and the best non-CSS QLDGM code designed in this paper. The Hashing bound is also shown. The codes are based on the parallel concatenation of regular LDGM codes with degree distribution* $\mathrm{P}[(8,8);(8,160)]$. *$p$ is the depolarizing probability.*

6.4.3.2 Comparison with existing QLDPC schemes

We close out this section by studying how our proposed non-CSS codes measure up against other QLDPC schemes in the literature. For this purpose, we conduct two different comparisons. We begin by comparing the codes in Figure 6.12 to other quantum codes of rate $\frac{1}{4}$ and then use a second comparison to study how decoders capable of exploiting the correlation between X and Z operators match up to our non-CSS construction. The first comparison is shown in Figure 6.13, which includes the performance for the following codes:

- The CSS QLDGM code based on a single regular LDGM code from [22]. The degree distribution of the underlying classical LDGM code is $\mathrm{P}(13, 13)$. The block length of the code is 19014.
- The $K = 32$ bicycle code of block length 19014 introduced by MacKay et al. in [18].
- The quantum serial turbo Code of [28], with block length 4000.
- The non-CSS concatenated code (Code C) from [63], with block length 138240.

Figure 6.13 shows that our proposed non-CSS QLDGM codes outperform existing quantum turbo codes and previously proposed quantum LDPC codes.

Comparison to improved CSS decoding strategies

In Chapter 5 we mentioned that CSS decoders capable of improving performance by exploiting the correlation that exists between X and Z operators over the depolarizing channel have been studied in the literature. Some of the earliest work on this topic was conducted in [55], where a set of modified BP decoders for CSS codes that reintroduce X/Z correlations were proposed. The most notable of these decoding schemes is known as the random perturbation decoder. In [135], further work on the topic of modified BP decoders for CSS codes was conducted and two novel CSS decoders were presented: the adjusted decoder and the augmented decoder.

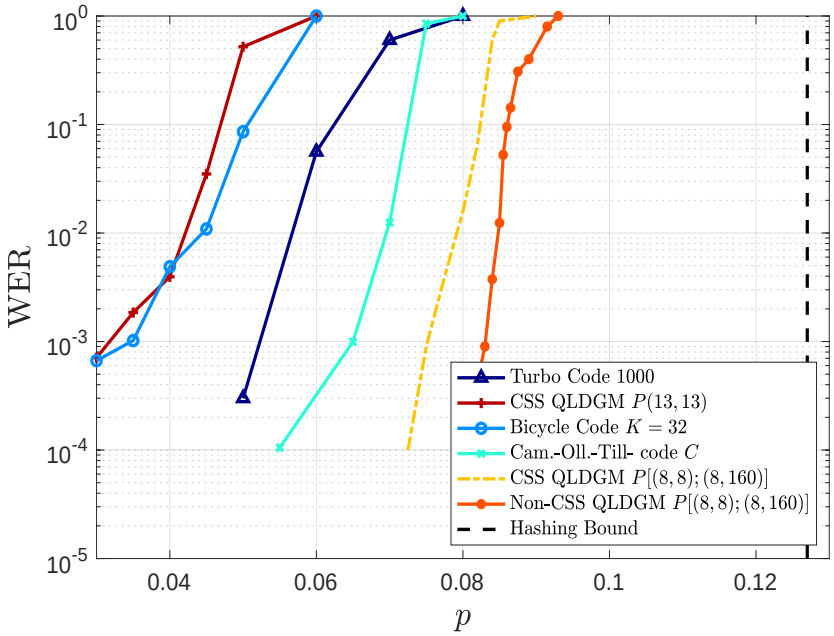


Figure 6.13: *Word error rate for various quantum codes when they are applied over the depolarizing channel. The Hashing bound is also included. p is the depolarizing probability.*

The adjusted decoder attempts to reintroduce the correlations between X and Z operators, neglected by a standard binary BP decoder, by adjusting prior probabilities. The augmented decoder operates by appending a specific subset of rows H_δ from the original PCM $\mathbf{H}$ into an exact copy of that matrix, resulting in an augmented PCM $\mathbf{H}_A = \begin{pmatrix} \mathbf{H} \\ \mathrm{H}_\delta \end{pmatrix}$ which is associated to a larger factor graph over which the decoding algorithm is run. Also in [135], the adjusted and augmentation techniques are combined to form a new CSS decoder known as the combined decoder. An in-depth look at some of these modified decoding strategies is provided in Appendix B.

In Figure 6.14, we compare the performance of the random perturbation decoder, as well as that of the adjusted, augmented, and combined decoders,

to that of our proposed non-CSS codes when they are applied over the depolarizing channel. The underlying LDGM code and CSS configuration is the same as in most of our previous simulations: the classical code is defined by the parallel concatenation P$[(8,8);(3,60)]$ and the CSS construction is achieved using M$(3;1,8.72)$ and $t = 4361$. Performance of a typical CSS decoder (separate decoding of X and Z operators) is also included in Figure 6.14. The non-CSS scheme is obtained from the CSS code by using the first design method and setting $q = 500$, which is the parameter configuration that produced the best results in our earlier simulations. Both the CSS code and the non-CSS configuration have rate $R_Q = \frac{1}{4}$.

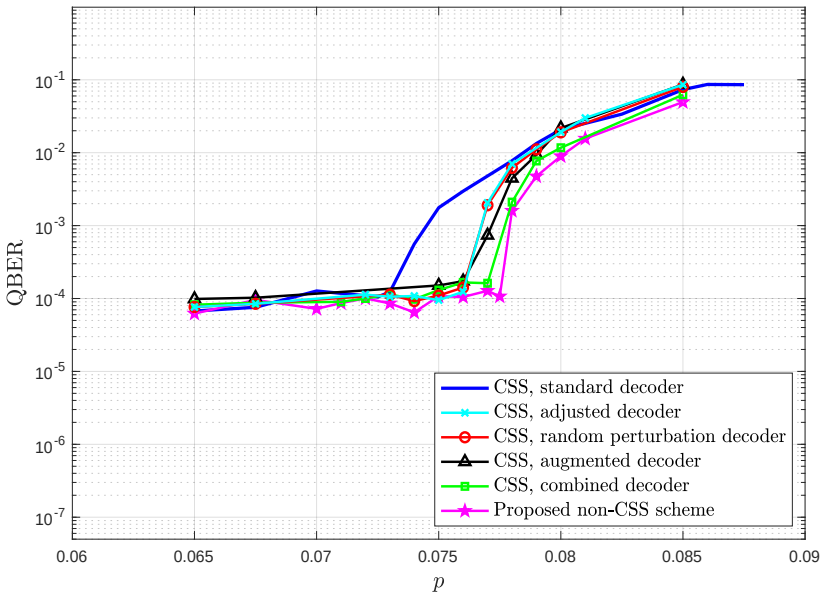


Figure 6.14: *Simulated QBER for different types of modified CSS BP decoders over the depolarizing channel. Results for the separate decoding of X and Z errors (standard CSS decoding) and for our proposed non-CSS scheme are also included. p is the depolarizing probability of the depolarizing channel.*

As shown in Figure 6.14, all of the modified CSS BP decoders yield performance improvements when compared to the generic CSS decoding scheme. Nonetheless, they are all outperformed by our proposed non-CSS construction, despite the close proximity between the QBER curves of the combined decoder and the non-CSS scheme. Importantly, we must note that although these modified decoders achieve performance close to that of our proposed scheme, they require specific modifications to the decoding algorithm that result in higher decoding complexity. This increase in decoding complexity arises from the fact that the modified CSS schemes require the execution of a standard decoder prior to their application. As mentioned in section 5.4, most modified decoding strategies rely on the failure of a standard decoder to then apply modifications to the factor graph according to the channel error estimate produced by the failed standard decoder. Once these modifications have been made (either to the a priori probabilities or to the factor graph itself) decoding is re-attempted. In some instances, decoding must be re-attempted multiple times before a correct syndrome estimation is produced. Moreover, the augmented decoder and the combined decoder operate over larger factor graphs, which further increases decoding complexity.

Since our proposed construction is decoded by running the SPA over the corresponding factor graph, decoding never has to be re-attempted, and so the complexity of our decoder is essentially the same as that of a standard CSS decoder. Therefore, it is clear that the scheme proposed in this chapter outperforms the aforementioned CSS decoders while displaying a lower decoding complexity.

6.5 CHAPTER SUMMARY

This chapter has introduced a technique to design non-CSS quantum codes based on the use of the generator and parity check matrices of LDGM codes. The proposed methods are based on modifying the upper layer of the decoding graph in CSS QLDGM constructions. The simplicity of the proposed scheme ensures that the high degree of flexibility in the choice of the quantum rate and the block length for the CSS code utilized as a starting point is translated to the non-CSS design. Compared to quantum CSS codes based on the use of LDGM codes, the proposed non-CSS scheme is 0.2 dB closer to the Hashing bound in the depolarizing channel and outperforms all other existing quantum codes of comparable complexity.

CHAPTER 7

Performance of QLDPC codes over Pauli channels

"The future can ever promise but one thing and one thing only: surprises"

Steven Erikson.

In Chapter 6 we studied and analyzed the performance of quantum codes over the depolarizing channel. However, despite serving as a practical abstraction that is useful to design and evaluate the performance of quantum codes, the depolarizing channel model we have worked with up to this point does not always provide an accurate representation of practical communication schemes or the real behaviour of quantum devices. A good example of this is the scenario known as channel mismatch, which describes the event in which the value of the depolarizing probability is not known prior to decoding. Notice how, up to this point, a perfect channel knowledge assumption was implicit in our simulations, i.e, we considered that the decoder knew the precise value of the depolarizing probability prior to decoding. Clearly, the depolarizing channel model that we have considered previously does not suffice to represent the channel mismatch scenario. Another instance in which a different channel model becomes necessary is when seeking to represent the behaviour of realistic quantum devices. Many of these devices behave asymmetrically: the probability of

a phase-flip taking place is actually orders of magnitude higher than the probability of a bit-flip; a type of behaviour that cannot be appropriately described using the depolarizing channel. For these reasons, in this chapter we introduce different Pauli channel models that are appropriate to represent realistic scenarios that the depolarizing channel cannot re-enact. We focus on the phenomenon of channel mismatch in section 7.1 and then analyze the performance of CSS codes over asymmetric Pauli channels in 7.2.

7.1 PERFORMANCE OF NON-CSS QLDPC CODES OVER THE MISIDENTIFIED DEPOLARIZING CHANNEL

In the previous chapter we introduced a non-CSS inspired LDGM-based strategy that outperforms all other CSS and non-CSS codes of similar complexity over the depolarizing channel. We did so, as is common throughout the literature of QEC, under the tacit premise that perfect knowledge of the quantum channel in question was available. In reality, such a scenario is highly unlikely, and the depolarizing probability must somehow be estimated prior to decoding. This makes it relevant to study how the behaviour of QEC codes can change in terms of the existing information about the quantum channel.

The phenomenon of channel mismatch in the context of QEC was first considered in [185], where the authors studied the impact of decoding QLDPC codes with imperfect channel knowledge over the depolarizing channel. In [186], the same authors designed an improved decoding strategy for QLDPC codes when only an estimate of the channel depolarizing probability was available. This last method made use of quantum channel identification, a technique that requires the introduction of a probe (a known quantum state) into the quantum channel and the subsequent measurement of the channel output state to produce an accurate estimate of the depolarizing probability. Quantum channel identification procedures typically require additional qubits and result in a latency increase, making the error correction procedure increasingly cumbersome. For this reason, researchers have since focused on the design of methodologies that can minimize this overhead (by avoiding the use of channel identification) while yielding performance similar to the perfect channel knowledge scenario. Such techniques have already been explored and successfully demonstrated for QTCs. In [31] the authors derive a so-called on-line depolarizing proba-

bility estimation technique that yields similar performance to that obtained when using the same QTCs with perfect channel information but without the need for additional resources. In light of this outcome and because an equivalent strategy has not been proposed for QLDPC codes, in this section we derive a similar on-line estimation procedure to that of [31] that can be applied to sparse quantum codes.

7.1.1 QUANTUM CHANNEL IDENTIFICATION

A common assumption in the field of QEC is that perfect knowledge of the quantum channel under consideration is available prior to decoding. In reality, this information cannot be readily obtained, and estimates of the corresponding quantum channel parameter must be derived. In the context of the depolarizing channel, this means that the decoder must be provided with an estimate of the channel depolarizing probability prior to the execution of the decoding process. If the estimated value of the depolarizing probability $\hat{p}$ is different to its actual value p, i.e., $\hat{p} \neq p$, we encounter the scenario known as *channel mismatch*, which typically results in the degradation of the performance of QEC codes (most codes performs worse at the channel noise value $\hat{p}$ than at the real value p).

To study the sensitivity of the non-CSS QLDGM scheme derived in the previous chapter to the channel mismatch phenomenon, we consider a scenario where $\hat{p}$ is varied while the true depolarizing probability remains fixed. We conduct these simulations using the $R_Q = \frac{1}{4}$ $q = 500$ non-CSS strategy derived based on the first methodology presented in section 6.3.1.1 and that employs an underlying parallel-concatenated LDGM code with degree distribution $P[(8, 8)(3, 60)]$. We have chosen the parallel concatenation with the smallest degrees tested in Chapter 6 in order to ease simulation requirements. As was mentioned in the previous chapter, the $q = 500$ M'_d matrix is obtained from two identical matrices defined by the configuration $M(3; 1, 8.72)$ and $t = 4361$.

The results of these simulations are shown in terms of the QBER in Figure 7.1. The values of p have been chosen so that they gradually get closer to the waterfall region[1] of the code ($p = [0.05, 0.06, 0.07, 0.075, 0.0775, 0.08]$), as this allows us to study the sensitivity of the code to the accuracy of the estimate $\hat{p}$ when the actual value of the depolarizing probability is varied.

[1]The decoding threshold or waterfall region of an error correction code is the region (value of p) where a sharp drop in the error rate takes place.

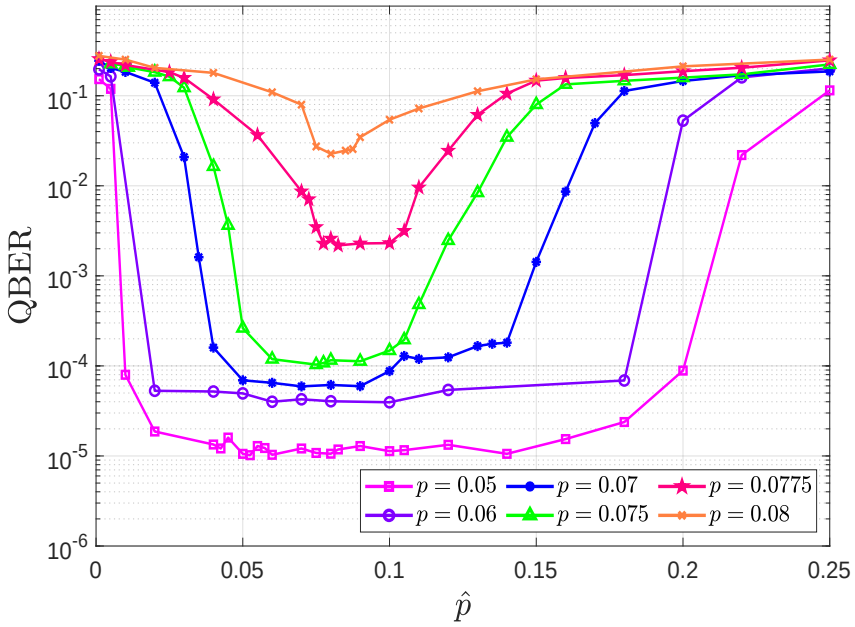


Figure 7.1: *Simulated QBER as a function of the estimated depolarizing probability $\hat{p}$ when the true depolarizing probability p is fixed.*

As shown in Figure 7.1, for smaller values of p ($p \leq 0.075$), the less accurate $\hat{p}$ needs to be to attain similar performance to when perfect channel knowledge is available (when $\hat{p} = p$). On the contrary, for larger values of the depolarizing probability that are closer to the waterfall region of the code ($p > 0.075$), higher accuracy of the estimate $\hat{p}$ will be necessary to achieve the best possible performance. This is reflected by the decrease in the width of the flat regions of the QBER curves as p is increased, where the flat region is defined as the part of the curve where the QBER is not significantly degraded [31]. This reduction in the size of the flat region indicates that the precision of the estimate $\hat{p}$ becomes increasingly important as p grows. For instance, if instead of estimating the value of the depolarizing probability we fix $\hat{p}$ to $p^* = 0.127$, which corresponds to the Hashing limit for a $R_Q = \frac{1}{4}$ quantum code, Figure 7.1 shows that for small values of p the resulting performance would be very close to that of a scheme

with perfect knowledge of the depolarizing probability. However, as larger values of p are considered, performance deteriorates further and further from the perfect channel knowledge scenario. For a noise level of $p = 0.075$, the QBER is 10^{-4} if $\hat{p} \approx 0.075$ but degrades to 10^{-2} if $\hat{p} = 0.0127$. It is clear from these results that techniques capable of providing good estimates of $\hat{p}$ are necessary when facing a channel mismatch scenario. This is discussed in the following subsections, where two different estimation methodologies than can be used to obtain $\hat{p}$ are presented.

7.1.1.1 Off-line Estimation Method

In quantum channel identification, estimates of the parameter that governs the behaviour of a specific quantum channel $\xi(p)$ are obtained by exposing a known quantum state σ, referred to as the probe, to the effects of said channel. Performing quantum measurements on the corrupted quantum probe, described by its density matrix $\rho_o(p)$, yields classical information from which an estimation of the parameter p can be obtained. Numerous experimental schemes have been devised to perform quantum channel identification: we can use unentangled quantum probes, probes entangled with ancilla qubits, or even probes entangled with other probes. Each of these strategies provides a unique set of advantages and disadvantages. However, because analyzing the performance of these schemes is outside the scope of this dissertation, in what follows we simply assume that an estimation set-up capable of obtaining the information-theoretical optimal performance is available.

Optimal estimation of the depolarizing probability of the depolarizing channel has previously been studied by making use of a metric known as the quantum Fisher information [187]. The quantum Fisher information of p is given by

$$J(p) = \mathrm{Tr}[\rho_o(p)\hat{\mathrm{L}}^2(p)],$$

where $\mathrm{Tr}[\rho_o(p)\hat{\mathrm{L}}^2(p)]$ represents the trace of the matrix $[\rho_o(p)\hat{\mathrm{L}}^2(p)]$, $\rho_o(p)$ is the quantum state at the output of the channel, and $\hat{\mathrm{L}}(p)$ is the symmetric logarithm derivative defined implicitly as

$$\frac{\partial \rho_o(p)}{\partial p} = \frac{1}{2}[\hat{\mathrm{L}}(p)\rho_o(p) + \rho_o(p)\hat{\mathrm{L}}(p)].$$

Since estimations of p are dependent on statistically distributed quantum measurements obtained from $\rho_o(p)$, the estimate of the depolarizing

probability $\hat{p}$ will be a random variable. Therefore, quantum channel identification comes down to selecting a procedure that provides the most accurate values of p. This is analogous to finding a method that minimizes the variance of the estimation $E\{(\hat{p}-p)^2\}$, assuming that the estimator is unbiased, $E\{\hat{p}\} = p$. In this context, the best possible performance of any estimator will be given by the quantum Cramér-Rao bound [188, 189], which states that the variance of the best possible estimator is bounded by

$$\mathrm{var}(\hat{p}) \geq \frac{1}{n_m J(p)} = \frac{1}{J_{n_m}(p)},$$

where $J_{n_m}(p) = n_m J(p)$ defines the overall Fisher information for n_m independent quantum measurements [189] and $J(p)$ denotes the Fisher information of p for a single measurement. As mentioned previously, because we operate under the assumption that our estimator attains the information-theoretical optimal performance, the variance of our estimator will be given by the quantum Cramér-Rao bound.

Based on the results shown in Figure 7.1, where we studied the QBER in terms of the mismatched depolarizing probability $\hat{p}$, we can now compute the average $\tilde{\mathrm{QBER}}(p)$ with regard to the real depolarizing probability of the channel p. This can be done as is shown in (7.1), where $P(\hat{p})$ is the probability density function of our optimal estimator.

$$\tilde{\mathrm{QBER}}(p) = \int \mathrm{QBER}(\hat{p}) P(\hat{p}) d\hat{p}. \tag{7.1}$$

As in [31], we assume that $P(\hat{p})$ is the truncated normal distribution defined between a and b, with mean μ, and variance $\frac{1}{J_{n_m}(p)}$. Once again, recall that the variance of $P(\hat{p})$ is chosen as the inverse of the asymptotically achievable Fisher information because we assume that we have the best possible quantum channel identification method at our disposal. The overall Fisher information $J_{n_m}(p)$ will vary in terms of the type of selected quantum probe. Although other works have studied the impact of using different types of probes [31], for the sake of simplicity, herein we will only consider the use of unentangled pure states as channel probes. In this case, the overall Fisher information is given by $J_{n_m}(p) = n_m\big(\frac{9}{8p(3-2p)}\big)$ [190].

Figure 7.2 shows the result of computing $\tilde{\mathrm{QBER}}(p)$ for the non-CSS QLDGM code considered previously as a function of the number of channel probes n_m. In [31], the off-line estimation protocol achieves the same

performance as the perfect channel information case when $n_m \approx 1000$. The results shown in Figure 7.2 indicate that convergence is faster for our codes, since performance close to the perfect information case is obtained for $n_m \approx 100$. As is also shown in [31], convergence to the perfect information case may be further improved by using maximally entangled pairs as probes (EPR pairs instead of pure states), i.e, less probes than when using unentangled pure states will be required.

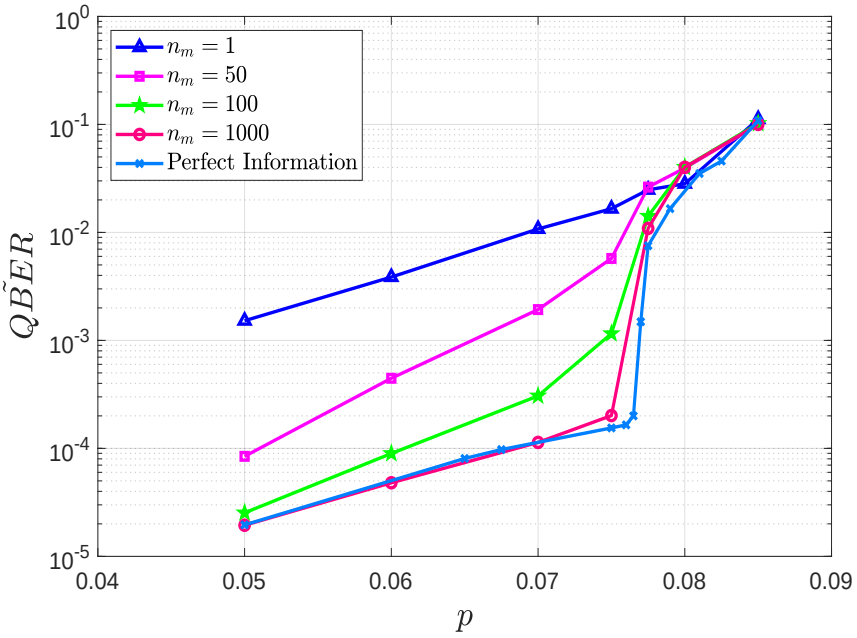


Figure 7.2: *Average QBER in terms of p when the number of used probes n_m is varied.*

Regardless of the type of quantum probe, the main handicap of off-line estimation protocols is that if the channel varies for every transmitted block, the overall rate of the QLDPC code that is being used will be severely reduced. Although this reduction in rate is asymptotically negligible for constant channels, it represents a significant drawback when using this estimation method in rapidly varying quantum channels [191].

7.1.1.2 Online Estimation Method

In similar fashion to what is done in [31] for QTCs, we apply slight modifications to the generic syndrome SPA iterative QLDPC decoder that allow us to estimate the depolarizing probability while decoding is taking place. This on-line estimation scheme does not require quantum channel identification, meaning that rate reduction is avoided regardless of the type of quantum channel under consideration, be it constant or block-to-block time varying.

In section 6.3.1 we saw how decoding of a non-CSS QLDGM code is performed by running the SPA over the factor graph associated to the equation $\mathbf{w} = \mathbf{H}_Q \odot \mathbf{e}$, where $\mathbf{w}$ is the measured syndrome, $\mathbf{H}_Q$ is the QPCM of the non-CSS code[2], and $\mathbf{e}$ is the symplectic representation of the error pattern induced by the quantum channel. The decoding objective is to find the most likely estimate of the channel error from the observed syndrome, i.e, the decoder must find the most likely estimate of the channel error, $\hat{\mathbf{E}}$, such that the estimated syndrome $\hat{\mathbf{w}} = \mathbf{H}_Q \odot \hat{\mathbf{e}} = (\mathrm{M}'_d \times \mathbf{H}_{\mathrm{CSS}}) \odot \hat{\mathbf{e}}$, is equal to the observed syndrome $\mathbf{w}$, where $\hat{\mathbf{e}}$ is the symplectic representation of $\hat{\mathbf{E}}$.

Against this backdrop, the online estimation decoding process works as follows:

1. First, the SPA is initialized using a "flooding" schedule in which the lower layer nodes of the factor graph transmit messages upwards in a layer-by-layer sequential manner until the top-most nodes are reached. These messages are based on an initial estimate of the depolarizing probability of the channel $\hat{p}^{(1)}$, which is used to compute the a priori log likelihood ratios of the algorithm.

2. Once information gets to the top layer, we say that the graph has been "flooded" and decoding can actually begin. Decoding then proceeds using a reversed schedule, in which, starting from the top-most syndrome nodes, messages are exchanged downwards and layer-by-layer until the bottom-most nodes are reached. The messages transmitted

[2]Recall that to reap the benefits of the non-CSS structure, the decoding algorithm must be run over the complete factor graph representation of the matrix product that defines the QPCM: $\mathrm{M}'_d \times \mathbf{H}_{\mathrm{CSS}}$. Decoding on the factor graph representation of the final matrix, $\mathbf{H}_Q$, results in worse performance.

by the syndrome nodes are computed considering information of the measured syndrome $\mathbf{w}$.

3. When two messages have been transmitted over every edge of the factor graph, an iteration of the decoding algorithm has been completed. At the end of each iteration, an estimate of the symplectic representation of the error pattern $\hat{\mathbf{e}}$ is produced and used to compute $\hat{\mathbf{w}}$.

4. If $\mathbf{w} = \hat{\mathbf{w}}$, then the algorithm has finished. If $\mathbf{w} \neq \hat{\mathbf{w}}$, the algorithm continues until it finds a matching syndrome or until a maximum number of iterations is reached. We can obtain an estimate of the depolarizing probability at each iteration j by assessing the number of X, Y, and Z operators present in the estimated error pattern[3] $\hat{\mathbf{E}}$ and dividing them by the block length of the code. This can be expressed as

$$\hat{p}^{(j)} = 1 - \frac{1}{N} \sum_{i=1}^{N} P^{(j)}(\hat{\mathrm{E}}_i = I|\hat{\mathbf{w}}), \tag{7.2}$$

where N is the block length of the code, I is the identity operator, $\hat{\mathrm{E}}_i$ is the i-th component of the estimated error pattern $\hat{\mathbf{E}}$, and $P^{(j)}(\hat{\mathrm{E}}_i = I|\hat{\mathbf{w}})$ is the probability at iteration j that the i-th component of the estimated error pattern is equal to the identity operator conditioned on the estimated syndrome.

5. Once $\hat{p}^{(j)}$ is obtained, it is used as the depolarizing probability to compute the necessary sum-product messages in the following iteration. Given the iterative nature of the decoding algorithm, we expect that each successive estimate $\hat{p}^{(j)}$ will get closer to the actual value of p, leading to better decoding performance.

A matter that we have yet to discuss is how an appropriate value for the initial estimate of the depolarizing probability $\hat{p}^{(1)}$ can be defined. It is possible that this initialization might affect the convergence of the estimates $\hat{p}^{(j)}$ to p, which may have an impact on decoder performance. Thus, we must conduct an analysis in which $\hat{p}^{(1)}$ is varied while p remains fixed. The

[3]Recall that we can obtain the Pauli operator representation from the symplectic representation by applying the inverse isomorphism β^{-1} to $\hat{\mathbf{e}}$ (see section 4.2).

results of this analysis are shown in Figure 7.3, where each dashed curve corresponds to a different value of the true depolarizing probability p. The curves associated to the original decoder, which where previously shown in Figure 7.1, are also included in Figure 7.3 as continuous lines.

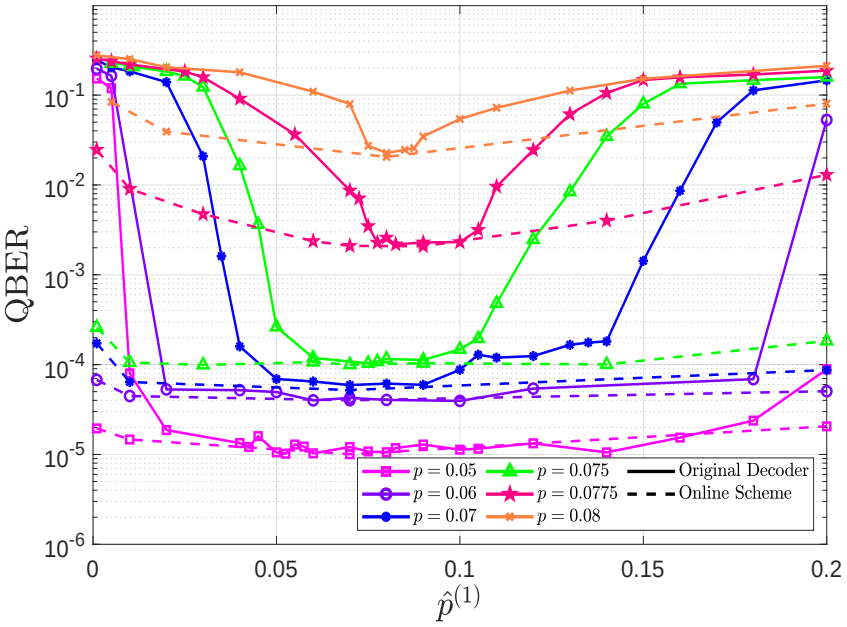


Figure 7.3: *Simulated QBER as a function of the initial estimate of the depolarizing probability of the channel $\hat{p}^{(1)}$. The continuous lines are associated to the original iterative QLDPC decoder which uses $\hat{p}^{(1)}$ for every iteration as if it were the true depolarizing probability. The dashed lines are obtained when the modified iterative decoder that uses the on-line estimation method is applied.*

Upon closer inspection of Figure 7.3, we can see that the performance of the on-line estimation method is similar to that of the perfect information scenario regardless of the value of $\hat{p}^{(1)}$. In fact, the modified on-line decoder significantly outperforms the original mismatched decoder, as is reflected by the flatter appearance of its performance curves. Figure 7.3 also shows how the sensitivity of the modified on-line decoder to the initial estimate

$\hat{p}^{(1)}$ increases as p grows. This is reflected in a reduction of the flatness of the on-line curves as p increases, which is most noticeable for $p = 0.0775$ and $p = 0.08$.

Once the depolarizing probability is higher than a certain threshold we enter the waterfall region of the code, where qubit errors occur with much higher probability than in the error-floor region. Even though the on-line method computes $\hat{p}^{(j)}$ during every iteration, our simulations results show that for sufficiently large values of p, if $\hat{p}^{(1)}$ is either too small or too large, the convergence of $\hat{p}^{(j)}$ to p is weakened and performance is hindered. This happens due to the combination of two factors: On the one hand, performance of the code is worse outside of the error floor region, and so the estimated error patterns are much more likely to have errors. On the other, small or large enough values of $\hat{p}^{(1)}$ make initial estimates of the error pattern contain either not enough or too many X, Y, and Z operators, which can corrupt the values of $P^{(j)}(\hat{\mathrm{E}}_i = I|\hat{\mathbf{w}})$ to the point that subsequent estimates $\hat{p}^{(j)}$ become increasingly inaccurate. This does not occur in the error floor region, where $\hat{p}^{(j)}$ converges to p regardless of the value of $\hat{p}^{(1)}$.

Ideally, we would like to define the value of $\hat{p}^{(1)}$ for which performance with the on-line estimation method is optimal. If we look at the curves corresponding to $p = 0.0775$ and $p = 0.08$, we can see performance is significantly degraded when $\hat{p}^{(1)} \leq 0.05$ or $\hat{p}^{(1)} \geq 0.13$. For the $R_Q = \frac{1}{4}$ code under consideration, the hashing limit is $p^* = 0.127$, which falls within the range $0.05 < p^* < 0.13$. Thus, performance on par with the perfect channel information scenario can be obtained with the on-line estimation method, regardless of the actual value of the depolarizing probability and without any additional resources or reductions in code rate, by setting $\hat{p}^{(1)} = p^*$.

7.2 DESIGN OF ASYMMETRIC QLDGM CODES

Throughout this chapter we have mentioned numerous times that most of the research in the field of QEC considers the symmetric instance of the general Pauli channel, commonly referred to as the depolarizing channel. However, realistic quantum devices often exhibit asymmetric behaviour, where the probability of a phase-flip taking place is orders of magnitude higher than the probability that a bit-flip occurs [97, 192, 193]. This asym-

metry stems from the nature of the physical mechanisms that define the behaviour of the materials that practical quantum processors are built from. These mechanisms are essentially governed by the single-qubit relaxation and dephasing times, which define the likelihood that a quantum device suffers bit-flips and phase-flips. Generally, relaxation causes both bit-flips and phase-flips, while dephasing only leads to phase-flip errors. In real quantum processors, the relaxation time can sometimes be orders of magnitude larger than the dephasing time. This difference in relaxation and dephasing times gives rise to the aforementioned asymmetric behaviour, where bit-flip errors are much less likely to occur than phase-flips.

Clearly, the depolarizing channel cannot be used to represent this type of behaviour. However, it can be accurately modelled using the general Pauli channel [97, 194, 195, 196, 197, 198]. Given the difference in the likelihood of bit-errors and phase-errors, it stands to reason that the QEC schemes built for this asymmetric channel must somehow exploit this asymmetry to achieve good performance. This has previously been studied in the context of QTCs in [199], where the authors introduce an EXIT-chart based methodology to design QTCs specifically for the general Pauli channel. This work was later extended in [192], where an online estimation protocol to decode QTCs over general Pauli channels was proposed. Because performance of the QTCs was shown to vary as a function of the asymmetry of the channel, these results speak to the merit of constructing a coding scheme tailored to the particular asymmetric characteristics of the quantum channel in question.

For this reason, in this section we study the performance of quantum CSS LDGM codes when they are applied over a general Pauli channel. We show how although they are not the best known codes for the depolarizing channel, their simplicity allows for them to be almost seamlessly adapted to the general Pauli channel. Based on this result, we introduce a simple yet effective method to derive CSS QLDGM codes that perform well over channels with varying degrees of asymmetry. Such a strategy is necessary because CSS codes designed for the depolarizing channel perform poorly over its asymmetric counterpart. To the extent of our knowledge and at the time of writing, the research on designing quantum codes specifically for asymmetric quantum channels is quite limited [200], especially when compared to results regarding the depolarizing channel. Thus, the work included herein represents one of the first attempts at designing QLDPC codes specifically for asymmetric quantum channels.

7.2.1 REALISTIC PAULI CHANNEL MODEL

Having established that the materials that make up quantum processors cause these devices to behave in an asymmetric manner, we must now build a model that can re-enact this particular behaviour. Asymmetry in the behaviour of quantum devices is embodied by the probability of a bit-flip p_x being orders of magnitude smaller than the probability of a phase-flip p_z. We know from our previous discussions regarding the general Pauli channel (see Chapters 3 and 6) that it is completely characterized by the probabilities p_x, p_y, and p_z. Because it allows us to vary these probabilities, the Pauli channel can be adapted to accurately represent the asymmetric behaviour exhibited by realistic quantum devices. To derive this so-called realistic Pauli channel model, we must first establish the asymmetric relationship between the probabilities p_x, p_y, p_z, and p. This can be done by introducing the parameter α, known as the channel's ratio of asymmetry [199], which represents the ratio of the phase-flip probability and the bit-flip probability as [201, 202, 203]

$$\alpha = \frac{p_z}{p_x} = 1 + 2\frac{e^{\frac{-t}{T_1}} - e^{(\frac{-t}{2T_1} - \frac{2t}{T_2})}}{1 - e^{\frac{-t}{T_1}}}, \tag{7.3}$$

where T_1 is the relaxation time, T_2 represents the dephasing time, and t is the coherent operation duration of a physical quantum gate [204]. In [97, 201] expressions for p_x, p_y, and p_z are given in (7.4), where the bit-flip probability and bit-and-phase-flip probability can be considered to be equal.

$$\begin{aligned} p_x = p_y &= \frac{1 - e^{\frac{-t}{T_1}}}{4} \\ p_z &= \frac{1}{2} - p_x - \frac{e^{\frac{-t}{T_2}}}{2}. \end{aligned} \tag{7.4}$$

If the coherent operation duration t is assumed to be reasonably short, i.e, $t << T_1$ [201], then from (7.3) the ratio of the phase and bit-flip probabilities can be approximated by $\alpha \approx 2\frac{T_1}{T_2} - 1$. In consequence, this model allows us to completely determine the values of p_z, p_x, and p_y from α and p. Common values for the ratio of asymmetry are given in [97, 192, 199], with most materials used to build quantum devices having $\alpha = [10^2, 10^4, 10^6]$. Notice that, if we select $\alpha = 1$, we obtain the depo-

larizing channel model that is considered in most circumstances and that satisfies $p_x = p_y = p_z = \frac{p}{3}$. Although we have not mentioned it explicitly, the case of $\alpha = 1$ has been observed for specific types of devices, meaning that the depolarizing channel can sometimes also provide a realistic representation of the behaviour of quantum machines.

7.2.2 PERFORMANCE COMPARISON WITH OTHER QLDPC CODES

From our discussion in Chapter 6 (see section 6.2), we know that designing LDGM-based CSS quantum codes is a fairly straightforward procedure. In fact, all that is needed to build the factor graphs of such codes is a set of four matrices: the generator and parity check matrices of a classical LDGM code, $\tilde{\mathrm{G}}$ and $\tilde{\mathrm{H}}$, and the matrices M_1 and M_2 described in theorem 3. These matrices define code parameters such as the rate R_Q or distance of the code. In Chapter 6 we designed these matrices for optimal performance over the depolarizing channel, but given how simple it is to modify them, it is likely that we can also construct them for good performance under different channel requirements. As will be shown in the following subsection, this comes in handy when designing codes for the asymmetric Pauli channel.

Since CSS QLDGM codes were originally designed to operate over the depolarizing channel, it is important to analyze how they perform over the depolarizing channel when compared to other existing QLDPC codes prior to modifying them for performance over different quantum channels. Although we mentioned them in Chapter 6, we have not yet analyzed the performance of CSS QLDGM codes in detail. For this purpose, in Figure 7.4 we show the highest possible coding rate at which various QLDPC codes that have appeared in the literature can achieve a WER of 10^{-3} over the depolarizing channel. This figure compares the performance of the following codes:

1. Symmetric CSS QLDGM code based on the structure $\mathrm{M}(3; 1, 11.04)$ and $\mathrm{P}[(8, 8); (8, 160)]$ with block length $n = 19014$ and rate $R_Q = \frac{1}{2}$.

2. Symmetric CSS QLDGM code based on the structure $\mathrm{M}(3; 1, 11.04)$ and $\mathrm{P}[(8, 8); (8, 160)]$ with block length $n = 19014$ and rate $R_Q = \frac{1}{4}$.

3. The $K = 32$ bicycle code with block length $n = 19014$ and rate $R_Q = \frac{1}{4}$ proposed in [18].
4. The Spatially Coupled (SC) Quasi-Cyclic (QC) QLDPC code of rate $R_Q = 0.49$ and block length $n = 181000$ given in [145].
5. The non-binary QC-QLDPC GF(2^{10}) code of rate $R_Q = \frac{1}{2}$ and block length $n = 20560$ proposed by Kasai et al. in [146, 147].
6. The SC-QLDGM code of $R_Q = \frac{1}{4}$ and block length $n = 76800$ proposed in [64].
7. The QTC-assisted SC-QLDGM code of $R_Q = \frac{1}{4}$ and block length $n = 821760$ of [62].
8. The non-CSS concatenated code (code C) of $R_Q = \frac{1}{4}$ and block length $n = 138240$ of [63].
9. The $t = 5000$, $q = 500$, M$(3; 1, 11.04)$, P$[(8, 8); (8, 160)]$ non-CSS QLDGM code of $R_Q = \frac{1}{4}$ and block length $n = 19014$ of [23] that was introduced in Chapter 6.

As can be seen in Figure 7.4, at both of the considered rates, the symmetric CSS QLDGM code is outperformed by some of the other codes. At a rate of $R_Q = \frac{1}{4}$, the symmetric CSS QLDGM code is beaten by the non-CSS implementation of [23] (as was shown in the previous chapter), the SC-QLDGM code of [64], and the QTC-assisted SC-QLDGM code of [156]. However, this comes as no surprise, since all three of these codes take a symmetric CSS QLDGM code as their starting point and then modify it (by changing the factor graph or combining them with a QTC) with the purpose of improving performance. For a rate of $R_Q = \frac{1}{2}$, the symmetric CSS QLDGM code is once again outperformed by the non-binary QC-QLDPC GF(2^{10}) code of [145, 146]. We can further expand this comparison by looking at the distance of each code to the Hashing bound. This can be done as in Chapter 6 by computing

$$\delta = 10 \log_{10} \left(\frac{p^*}{p} \right), \tag{7.5}$$

where p^* is the noise limit of the depolarizing channel for a specific quantum rate R_Q, and p is the highest depolarizing probability at which the code in question can operate with a WER of 10^{-3}.

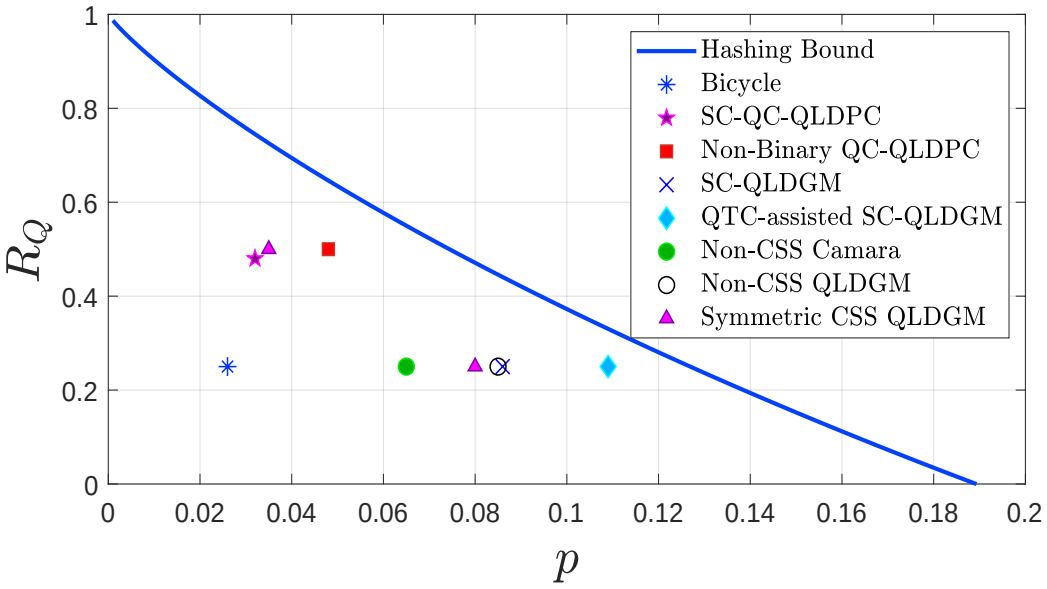


Figure 7.4: *Achievable quantum coding rate at a WER of* 10^{-3} *for different types of QLDPC codes.*

At a rate of $R_Q = \frac{1}{2}$, the QC-QLDPC code of [146, 147] is $\delta_{\text{QC-QLDPC}} = 1.9$ dB away from the Hashing bound. At this same rate, the distance for the symmetric CSS QLDGM code is $\delta_{\text{CSS-QLDGM}} = 2.86$ dB. These values make it clear that when $R_Q = \frac{1}{2}$, QC-QLDPC codes significantly outperform CSS QLDGM codes. In contrast, at a rate of $R_Q = \frac{1}{4}$, the symmetric CSS QLDGM code is $\delta_{\text{CSS-QLDGM}} = 1.95$ dB away from the Hashing bound. Once more, despite the improvement in performance at a this lower rate, the non-CSS and the SC-QLDGM codes of [23] and [64] which exhibit approximately the same distance to the Hashing Bound $\delta_{\text{non-CSS}} \approx \delta_{\text{SC-QLDGM}} = 1.69$ dB, outperform the CSS QLDGM code. However, it is worth mentioning that at this rate the difference in performance is notably less significant than when $R_Q = \frac{1}{2}$. This means that the performance of CSS QLDGM codes is better at a lower rate, which implies one of two things: that a different construction strategy must be employed when constructing higher rate CSS QLDGM codes, similar to what is done classically in [205], or that they should not be used for error correction when high rates are necessary.

At this point, it should be noted that these improvements in performance come at the expense of the complexity of the error correction schemes. This is especially true for the best known code for $R_Q = \frac{1}{4}$, the QTC-assisted SC-QLDGM code of [62], which requires a QTC and a large block length in order to get closer to the Hashing bound. Herein lies the main appeal of CSS QLDGM codes, because although they are slightly worse than the state-of-the art codes at $R_Q = \frac{1}{4}$, the simplicity with which their design parameters can be manipulated allows for them to be seamlessly adapted to different channels. Doing so for the other codes included in this discussion is a much more difficult task, since their increased complexity does not allow direct modifications like those permitted by CSS QLDGM codes. For this reason, and knowing that performance of CSS QLDGM codes over the depolarizing channel is acceptable at $R_Q = \frac{1}{4}$, we use these codes in the following section as the basis to design asymmetric CSS QLDGM codes for a more realistic Pauli channel model.

7.2.3 ASYMMETRIC QLDGM CSS CODES

Over Pauli channels that model practical quantum devices, a phase-flip is generally much more likely to occur than a bit-flip. Therefore, it is reasonable to assume that for a quantum error correction scheme to be optimal for this type of channel, it must be capable of appropriately exploiting the channel's asymmetry. To be more precise, attaining the best performance over such channels requires a more complex strategy (by modifying the code construction) than just adapting[4] the decoding strategy employed over the depolarizing channel by modifying the flip probabilities according to the new channel model. This means that decoding a symmetric QLDGM CSS scheme by feeding the a priori bit and phase error probabilities $f_m^1 = p_x$ and $f_m^2 = p_z$ of an asymmetric channel to the corresponding bit and phase error decoders, while certainly an improvement to decoding over an asymmetric channel based on the mismatched probability of the original i.i.d. channel $f_m = \frac{2p}{3}$, will not result in noticeable performance improvements. This is shown in the following section, which also portrays the significant improvement yielded by asymmetric CSS schemes that tailor specifically to the Pauli channel model for asymmetry.

[4]The simplest decoding strategy for an asymmetric channel would be to use a symmetric CSS QLDGM code in which the a priori probabilites of each individual decoder are the bit and phase flip probabilities instead of $\frac{2p}{3}$.

7.2.3.1 Adaptation of symmetric CSS QLDGM codes to the Pauli channel

The QLDGM CSS scheme introduced in section 6.2 can be adapted to an asymmetric channel by increasing the number of syndrome nodes used to decode the Z operators and decreasing the number of syndrome nodes used to decode the X operators. In this way, the decoder can take advantage of the channel's asymmetric behaviour and improve the performance of the error correcting scheme. The factor graph of a QLDGM CSS code tailored to the Pauli channel model for asymmetry is shown in Figure 7.5.

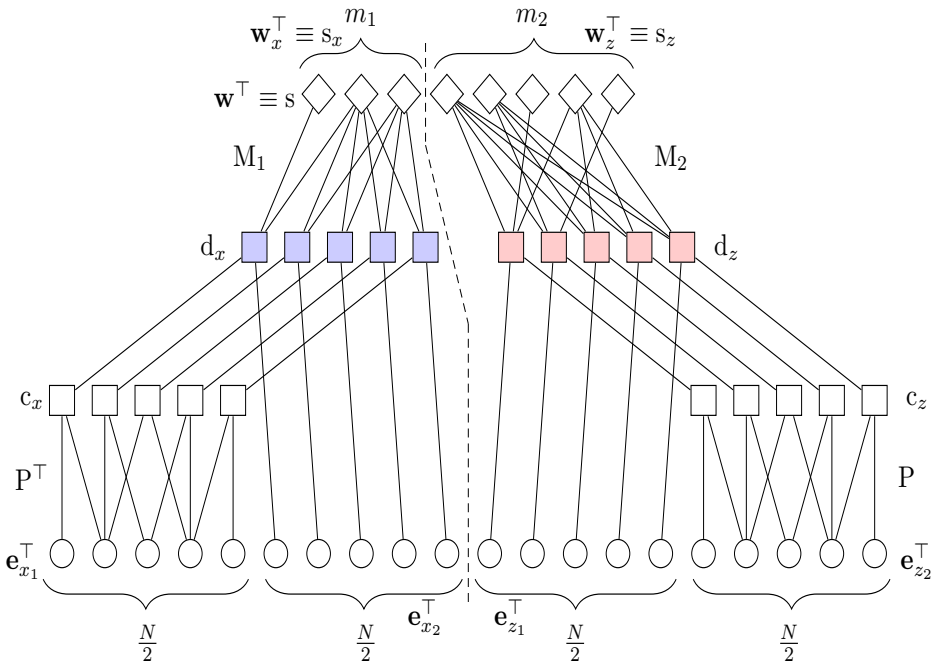


Figure 7.5: *Decoding graph for an asymmetric QLDGM CSS scheme. The dashed line is included to emphasize the separation of the two constituent subgraphs. The leftmost subgraph decodes the X errors while the one on the right decodes the Z errors. We have assumed that $n_1 = n_2 = \frac{N}{2}$, and $m = m_1 + m_2$.*

Despite how intuitive the idea appears, it is worth discussing why utilizing more syndrome nodes to decode Z operators and less syndrome nodes to decode X operators is beneficial when the considered channel is asymmetric. The asymmetry-integrating Pauli channel model causes phase-flips (Z errors) with much higher probability than bit-flips or bit-and-phase-flips (X and Y errors, respectively). Thus, the symplectic error representation of a pattern induced by this asymmetric channel will have a much higher number of non-zero elements in its $\mathbf{e}_z$ string than in its $\mathbf{e}_x$ string. In contrast, when $\mathbf{e}$ is produced by a depolarizing channel, $\mathbf{e}_x$ and $\mathbf{e}_z$ will, on average, have the same number of non-zero entries. This presence of similar amounts of X and Z errors in error sequences produced by the depolarizing channel is the reason why the decoding graph of Figure 6.1 uses the same amount of syndrome information to decode the $\mathbf{e}_x$ and $\mathbf{e}_z$ nodes. However, such a graph will more than likely not be optimal for an asymmetric scenario in which the distribution of non-zero entries over the length N constituent strings of the symplectic representation of an error pattern is not equal like for the depolarizing channel.

Let $\mathbf{e}^{\text{dep}} = (\mathbf{e}_x^{\text{dep}}|\mathbf{e}_z^{\text{dep}})$ and $\mathbf{e}^{\text{asym}} = (\mathbf{e}_x^{\text{asym}}|\mathbf{e}_z^{\text{asym}})$ denote the symplectic representations of two error patterns induced by a depolarizing channel and a Pauli channel that models a realistic quantum device, respectively. Let us also define the operator $\sigma(\mathbf{a})$, which computes the number of non-zero entries in a binary string $\mathbf{a}$. Finally, assume that the asymmetry coefficient of the asymmetric Pauli channel in question satisfies $\alpha \geq 10^2$. For the same value of p, $\sigma(\mathbf{e}^{\text{dep}}) \approx \sigma(\mathbf{e}^{\text{asym}})$. However, while $\sigma(\mathbf{e}_x^{\text{dep}}) \approx \sigma(\mathbf{e}_z^{\text{dep}})$, the same does not occur for the asymmetric channel, $\sigma(\mathbf{e}_z^{\text{asym}}) \ggg \sigma(\mathbf{e}_x^{\text{asym}})$. Additionally, $\sigma(\mathbf{e}_z^{\text{asym}}) \gg \sigma(\mathbf{e}_z^{\text{dep}})$ and $\sigma(\mathbf{e}_x^{\text{asym}}) \ll \sigma(\mathbf{e}_x^{\text{dep}})$. In consequence, it is quite obvious that a decoder tasked with decoding error patterns induced by a general Pauli channel will benefit from an uneven decoding graph in which more syndrome information is employed to decode $\mathbf{e}_z$ and less syndrome nodes are utilized to decode $\mathbf{e}_x$.

The design of such a decoding graph, an example of which is shown in Figure 7.5, gives rise to a whole new set of questions. The first and most significant one is how can the optimum values for m_1 and m_2 be determined, where m_1 and m_2 denote the number of syndrome nodes used to decode the X and Z operators, respectively. It is evident that $m_2 > m_1$, with said difference growing larger as the asymmetry of the channel increases. Ideally, we would like to devise a mathematical formulation from which

the values of m_2 and m_1 that yield the best possible performance could be obtained.

Another important matter, of which little insight is possessed, is which configuration of the M_1 and M_2 matrices will yield the best results. However, establishing which values of $(y; 1, x)$ and t for each of these matrices is optimal, further augments the complexity of the asymmetric design procedure when compared to the symmetric scenario. This increase in complexity is caused by the fact that exploiting the asymmetry of the channel requires $\mathrm{M}_1 \neq \mathrm{M}_2$, which allows for myriads of different configurations in terms of the values chosen for m_i, y_i, x_i, and t_i, where $i = 1, 2$. Finding the optimum configuration through a brute force search requires such a plethora of simulations that the issue becomes computationally intractable.

In order to simplify our search for these matrices, we recover the design methodology used in Chapter 6 (that of [155, 176, 183]) to construct symmetric CSS codes. This procedure reduces the complexity of the construction of the M_1 and M_2 matrices by considering $\frac{m}{2} = m_1 = m_2$ and $\mathrm{M}_1 = \mathrm{M}_2 = \mathrm{M}$. This means that instead of building two different matrices, the same matrix M is used to define the upper layers of both CSS subgraphs.

The methodology to construct $[\mathrm{M}(y; 1, x)]_{\frac{m}{2} \times \frac{N}{2}}$ begins by defining the values of m, the total number of syndrome nodes of the decoding graph, and N, the block length of the code. N is chosen to be sufficiently large so as to ensure that the code will possess good error correcting capabilities, while m is selected to guarantee that the code has the desired quantum rate, which for these CSS QLDGM codes is given by $R_Q = \frac{N-m}{N}$. Once again, since these codes are built for the symmetric Pauli channel, $\frac{m}{2}$ syndrome nodes are assigned to each CSS subgraph. Following this, y is set as a natural number to make sure that the d nodes of the CSS subgraphs have the same number of edges, and the number of s_A nodes is chosen as $t \leq \frac{m}{2}$. Finally, x is obtained from the following equation:

$$\left(\frac{m}{2} - t\right)x + t = y\frac{N}{2}. \tag{7.6}$$

In [155, 176, 183], where $R_Q = \frac{1}{4}$ codes with $N = 19014$ and $m = 14262$ are considered, the configurations of $[y, x, t]$ that achieved the best performance were: $[3, 8.72, 4161]$ and $[3, 11.04, 5000]$. These configurations were

also shown to be slightly dependant on the characteristics of the underlying parallel-concatenated classical LDGM code.

Finding the combination of the parameters involved in (7.6) that produces the code with best possible performance is no easy task. Nonetheless, as was done in the previous chapter, certain assumptions can be made in order to reduce the complexity of this endeavour. To begin with, we know that N and m are fixed in order to define the desired rate of the code. If N is appropriately chosen (it is large enough to guarantee good error correction potential of the code), this simplifies matters and reduces the number of parameters from (7.6) that must be studied. Now, recall that y must be set as a natural number to ensure the regularity of the d nodes. In [155, 176, 183] results showed that only a single value of this parameter yielded positive outcomes[5], $y = 3$. Given that the codes we will construct in this chapter are based on the structures proposed in [155, 176, 183], it is reasonable to adopt the same value for y in our constructions. In consequence, this results in N, m, and y being fixed to specific values, implying that the only parameters in equation (7.6) that can actually be modified are t, the number of s_A nodes, and x, the degree of the s_B nodes. Several insights regarding the value of these parameters can be obtained based on our previous discussions and the aforementioned equation:

- For large values of x, the reliability of the messages transmitted by the s_B nodes in the decoding process is significantly reduced. This occurs because when nodes have many edges in SPA-based decoding, the messages that are considered in the computations of each of these nodes are numerous enough to have an "averaging" effect and reduce the impact of any one given message. Naturally, this should hinder the overall performance of the code.

- As the values of x grow, given that the RHS of (7.6) is fixed, the value of t will also be larger. Note that this is intuitive: the more degree-1 syndrome nodes that there are (the larger the value of t), the larger the degree of the remaining s_B nodes (the larger the value of x) will be because the degree y of the lower layer d nodes must remain the same, which can only be guaranteed by adding more edges to the s_B nodes. Growth in the value of t should have a positive impact on

[5]Choosing $y = 2$ resulted in too little syndrome information being propagated throughout the graph and applying $y = 4$ resulted in worse and slower decoding due to the large amount of messages exchanged over the graph.

performance, as having more s_A nodes in the decoding graph means that more "perfect" information from these degree-1 syndrome nodes will be transmitted to the lower layer nodes in the initial decoding iterations.

Against this backdrop, it seems likely that the optimum values of x and t will be dictated by a trade-off between these two effects and that the choice of x will be inherently linked to the choice of t. As was mentioned previously, this search for the best combination of t and x is further complicated by the fact that our goal is to design asymmetric CSS codes. This is more difficult than building CSS QLDGM codes for the depolarizing channel because the matrices M_1 and M_2 must now be different in order to exploit the asymmetry of the channel. Because more syndrome nodes are used to decode phase flips and less syndrome nodes are used to decode bit flips ($m_1 \neq m_2$), the x and t parameters corresponding to each of these matrices must now be optimized. Therefore, to appropriately design LDGM-based CSS codes for the Pauli channel model presented in section 7.2.1, we have to adapt equation (7.6) into the new version shown below

$$\left(\frac{m_i}{2} - t_i\right) x_i + t_i = y_i \frac{N}{2}, \tag{7.7}$$

where $i = 1, 2$. Essentially, two designs have to be optimized instead of one: we must now find the configurations of $M_{m_1 \times \frac{N}{2}}(y_1; 1, x_1)$ and t_1 and $M_{m_2 \times \frac{N}{2}}(y_2; 1, x_2)$ and t_2 that yield the CSS codes with the best performance. Recall that the rate of the scheme will remain fixed since the sum $m = m_1 + m_2$ does not change. The demands of this process are discussed in the following section, where we show how in reality, most of the parameter optimization is only needed for one of these matrices.

7.2.4 SIMULATIONS

We will now study the performance of the proposed asymmetric CSS scheme over the Pauli channel model for asymmetry. First, we perform simulations to analyze the behaviour of a variety of asymmetric schemes over a Pauli channel with a specific degree of asymmetry and compare these results to the performance of a symmetric CSS code when it is applied over that same channel. Based on these results, we proceed by studying

those schemes that yield the best performance and narrowing down the search for the optimum size and configuration of the M_1 and M_2 matrices. Then, we propose a methodology to design asymmetric CSS QLDGM codes based on the asymmetry coefficient of the channel. Finally, we compare the performance of the proposed asymmetric schemes to the theoretical limits of the Pauli channel and study how they measure up against other codes that are found in the literature.

7.2.4.1 Performance over the asymmetric Pauli Channel

Realistic Pauli channel models for quantum devices induce phase-flips (Z errors) with much higher probability than bit-flips (X errors) [97]. Asymmetric CSS QLDGM schemes can exploit this phenomenon by utilizing more syndrome nodes to decode Z errors and employing less syndrome information to decode X errors. Given that the only design guideline we possess to begin this analysis is that m_2 should be larger than m_1, we start by fixing the asymmetry coefficient of the channel to $\alpha = 10^2$, and simulating different configurations of the proposed asymmetric CSS scheme. For comparison purposes, we also simulate a symmetric CSS code over the Pauli channel with $\alpha = 10^2$. We select the value $\alpha = 10^2$ because it is the smallest out of the set of realistic values for the asymmetric coefficient provided in [97, 199]. Performance of the proposed schemes for channels with other degrees of asymmetry is studied in the last part of this section.

For our simulations, we build codes of quantum rate $R_Q = \frac{1}{4}$ and block length $N = 19014$ that encode $k = 4752$ qubits into N qubits. The pseudo-random matrix P of the underlying LDGM code has size 9507×9507, has the same degree distribution as its transpose $\mathrm{P}^\top$, and corresponds to a rate $\frac{1}{2}$ classical irregular LDGM code. The irregular LDGM code is designed via the parallel concatenation of two regular LDGM codes. As was done in Chapter 6, we use the particular concatenation $\mathrm{P}[(8,8)(3,60)]$ because of its relatively small number of degrees, which reduces simulation time substantially. Once the optimum M_1 and M_2 configuration has been found, a parallel concatenated LDGM code of larger degrees can be used to improve performance. Figure 7.6 shows the performance of the simulated schemes and table 7.1 outlines the details of each specific design. The results are depicted using the QBER and the WER.

As was expected, the results shown in Figure 7.6 portray how the asymmetric CSS schemes outperform the symmetric CSS code over the Pauli

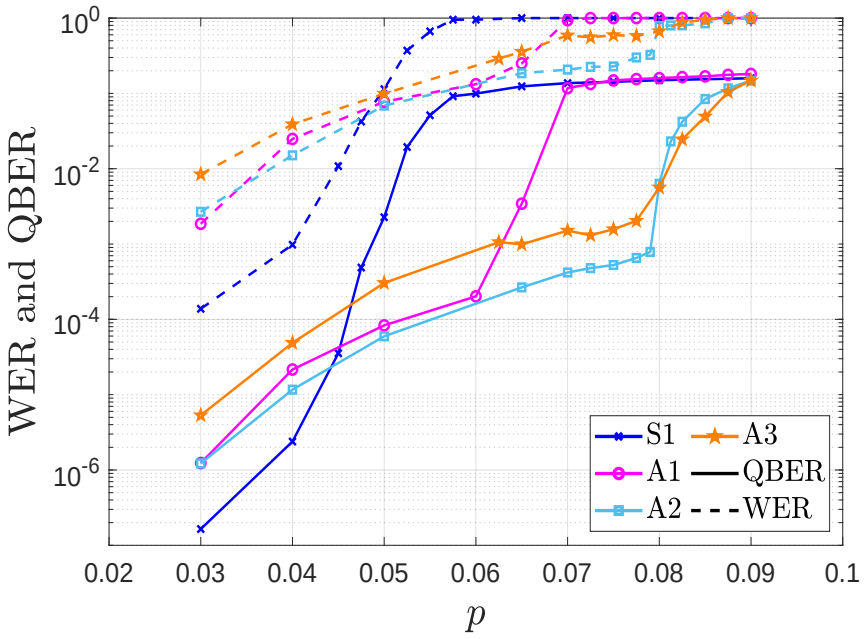


Figure 7.6: *Simulated WER and QBER for different CSS QLDGM schemes. p represents the gross flip probability of the Pauli channel with asymmetry coefficient $\alpha = 10^2$.*

Table 7.1: *Parameter values and configurations of the CSS codes simulated over an Pauli channel with $\alpha = 10^2$. The results of these simulations are shown in Figure 7.6.*

Code Type	#	$[m_1, t_1, x_1, y_1]$	$[m_2, t_2, x_2, y_2]$
Symmetric	S1	$[7131, 4161, 8.22, 3]$	$[7131, 4161, 8.22, 3]$
Asymmetric	A1	$[6262, 3000, 7.82, 3]$	$[8000, 5100, 6.33, 3]$
Asymmetric	A2	$[3262, 1500, 12.49, 3]$	$[11000, 8497, 8, 3]$
Asymmetric	A3	$[1262, 750, 54.24, 3]$	$[13000, 9507, 6.9, 3]$

channel with $\alpha = 10^2$. This can be appreciated by observing how the QBER and WER curves of the symmetric CSS code (code S1) enter the waterfall region and experience significant degradation at a substantially lower value of p than the asymmetric codes.

The simulation outcomes depicted in Figure 7.6 also serve to provide insight regarding the performance determining factors of our asymmetric schemes. For instance, defining an increasingly unbalanced configuration of the upper layer of the decoding graph by selecting larger values of m_2 and decreasing the values of m_1 appears to have a positive impact on our construction. This is reflected by the betterment in QBER and WER results of the codes of table 7.1 as m_2 grows. However, the performance curves of code A3, which has the largest value of m_2, exhibit a higher error floor than all the other simulated codes. This may occur because selecting such a large value for m_2 reduces the number of syndrome nodes leftover to decode the X operators to such an extent that the corresponding decoder, despite the low probability of bit-flips over the channel, sees an inevitable increase in its error rate. At the same time, it may also be that for such a small value of m_1, the choice of $[t_1, x_1, y_1]$ is so critical that inappropriate selection of these values degrades performance significantly. Thus, although large values of m_2 provide more syndrome information to decode the Z operators and improve the ability of the code to correct phase-flips, they come at the expense of using less information to correct bit-flips (values of m_1 that are too low), which results in increased error floors and worse overall performance if the bit-flip decoder is not correctly designed.

Another aspect of the proposed asymmetric CSS scheme that is integral to its performance and which was discussed in the previous section, is the relationship between the degree of the s_B nodes of the CSS decoding subgraphs, denoted by x_i, and the number of degree-1 syndrome nodes t_i, where $i = 1, 2$. The impact of this relationship, and specifically its aforementioned trade-off nature, is significant, as it ties into the selection of m_i. For the symmetric CSS schemes of [155, 176] the best performance was obtained for codes that utilized $\mathrm{M}_1 = \mathrm{M}_2 = \mathrm{M}(3; 1, 11.02)_{7131\times 9507}$ and $t = 5000$. Let us assume that the optimal degree of x for matrix M of a symmetric CSS code will still be optimal for the M_1 and M_2 matrices used to build each of the subgraphs of an asymmetric CSS code. In reality, achieving a configuration where $x_i = 11.02$ for a scenario in which $\mathrm{M}_1 \neq \mathrm{M}_2$ will not always be possible. If we revisit equation (7.7) we can understand why this happens. Once the quantum rate of the code has been selected,

aside from t_i, the only other parameter we can modify is y_i. Recall, that in [155, 176], results showed that only a single value of this parameter yielded good outcomes, $y = 3$. Thus, since N, m_i, and y_i are fixed and t_i is bounded[6] by $\frac{N}{2}$, it will not always be possible to build matrices that have $x_i = 11.02$.

We illustrate this with an example: Introducing $N = 19014$, $y_2 = 3$, $m_2 = 11500$ and the maximum possible value of $t_2 = 9507$ into (7.7), we obtain $x_2 = 9.54$. Since t_2 cannot be increased further, a scheme with this parameter configuration will have a maximum s_B node degree of $x_2 = 9.54$. In consequence, it becomes apparent that our choice of m_1 and m_2 also affects the values of x_1 and x_2.

Although this entire design procedure for M_1 and M_2 may seem overwhelming, this last remark regarding the value of x_i is actually a positive outcome. If we can show that performance of the asymmetric schemes is optimized for a specific value of x_i, then the asymmetric CSS code design procedure can be reduced to finding the values of m_i and t_i that produce this particular value of x_i.

The last detail worthy of mention related to the results shown in Figure 7.6 is that, aside from code A3, whose decoding errors are overwhelmingly caused by X operators, the entirety of the decoding errors of all the other asymmetric codes are caused by phase-flips. Although not surprising given the nature of an asymmetric Pauli channel, it would be ideal to design an asymmetric scheme in which errors are equally distributed, as is the case over the depolarizing channel. In other words, we would like the X and Z operator decoders of our asymmetric CSS codes to fail with similar rates, instead of all the decoding errors being attributed to one of them.

Impact of CSS decoding

Generally, CSS codes are decoded separately by executing the SPA over each of the subgraphs of the overall CSS factor graph. Over the depolarizing channel, given the equal likelihood of X and Z error events, the error contributions of each individual decoder to the overall code are essentially identical. Over the asymmetric channel, however, maintenance of this sep-

[6] The parameter t_i can theoretically be as large as m_i. However, since each decoding subgraph only has $\frac{N}{2}$ d nodes, it is does not make sense to choose $t_i > \frac{N}{2}$ since with $t_i = \frac{N}{2}$ all the d nodes are already d_a nodes (they receive perfect syndrome information).

arate decoding policy results in each of the CSS decoders impinging on the error correcting capabilities of the overall CSS code in a different manner. In what follows we will study the impact of each of the individual decoders on the performance of the asymmetric CSS code by simulating different configurations. Throughout this discussion we will use the terms X operator decoder and Z operator decoder to refer to the decoding process that unfolds over the CSS subgraphs associated to the X and Z operators, respectively. Note that we are slightly abusing the concept of a decoder since what we actually simulate are different subgraph configurations, not different decoders.

Let us first consider the X operator decoder. This decoder can cause an increment in the error floor of the code if the values of m_1 and $[y_1, x_1, t_1]$ are not chosen correctly. For sufficiently large values of $m_1 < \frac{N}{2}$, due to the low likelihood of X errors, the bit-flip decoder performs well regardless of the values of $[y_1, x_1, t_1]$, but when m_1 becomes too small, performance of the decoder is only acceptable if $[y_1, x_1, t_1]$ are chosen appropriately. This can be seen in Figure 7.7, where the QBER and WER curves of X operator decoders with $m_1 = 1262$ and different configurations of $[y_1, x_1, t_1]$ are shown. The complete characteristics of these X decoders are detailed in table 7.2.

Table 7.2: *Parameter values and configurations of X operator decoders of asymmetric CSS codes simulated over a Pauli channel with $\alpha = 10^2$. The results of these simulations are shown in Figure 7.7.*

Decoder	m_1	t_1	x_1	y_1
A1	1262	900	76.3	3
A2	1262	750	54.2	3
A3	1262	300	29.3	3
A4	1262	100	24.4	3

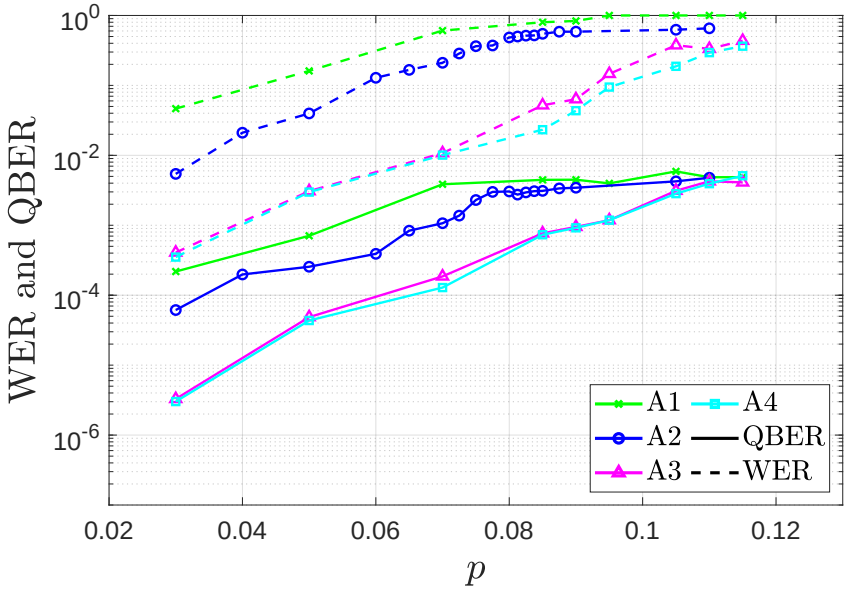


Figure 7.7: *Simulated WER and QBER for different X operator decoders. p represents the gross flip probability of the Pauli channel with asymmetry coefficient $\alpha = 10^2$.*

As is shown in Figure 7.7, performance of the X decoders varies significantly depending on the values of $[y_1, x_1, t_1]$. Decoders A1 and A2 are substantially worse than decoders A3 and A4. They mainly differ in the values of the parameters x_1 and t_1, which are much larger for decoders A1 and A2 than for decoders A3 and A4. This implies that the performance of X decoders with smaller values of m_1 will be better when lower values of x_1 are chosen by using less degree-1 syndrome nodes in the decoding graph, i.e, selecting smaller values for t_1. This is logical, since as was mentioned in the previous section, despite the fact that selecting larger values of t_1 will increase the amount of exact information transmitted from the upper layer nodes during initial iterations, it will also make the degree of the s_B nodes so large that the impact of these "perfect" messages might be mitigated and message passing may not operate successfully. For a decoder with a small value of m_1 ($m_1 = 1262$), if $[y_1, x_1, t_1]$ are chosen correctly, performance of the X decoder is excellent, with its QBER and WER curves

increasing in an almost linear fashion as functions of p (codes A3 and A4 of Figure 7.7). We will later see how the performance curves of such aptly built X decoders can sometimes be orders of magnitude below those of the corresponding Z decoder. On the contrary, if the selected configuration of $[y_1, x_1, t_1]$ yields a value of x_1 that is too large, performance of the decoder will be degraded enough to cause an increase in the error floor of the overall CSS code.

The Z operator decoder faces the daunting task of correcting the much more frequent Z errors. Considering the previous discussion regarding the X operator decoders, if we assume that we have an appropriately designed X operator decoder, we hypothesize that the performance of the CSS code as a whole will be majorly determined by the quality of its phase-flip decoder. To evaluate this hypothesis we study the performance curves shown in Figure 7.8, which correspond to the Z decoder of code A3 of table 7.1, the X decoder A3 of table 7.2, and the the CSS code that arises when using these two decoders simultaneously.[7]

Figure 7.8 shows how the error contribution of the X decoder to the QBER and WER curves of the overall CSS code is almost negligible when compared to the Z decoder. This occurs because the X decoder is correctly designed (the design parameters m_1 and $[y_1, x_1, t_1]$ have been selected appropriately), contrary to the X decoder of code A3 in Figure 7.6. Additionally, the results of Figure 7.8 show that the performance curves of the CSS code and its Z operator decoder are very similar, especially at the decoding threshold[8]. Two factors play a role in defining when the CSS code reaches its decoding threshold. The first, which will be discussed later on in this section, is the block length of the code itself. The second, is the design of the individual CSS decoders of the code. The abrupt increase in the error rate for the Z decoder (both for the WER and the QBER) while the X decoder maintains performance at the error floor, proves that for our proposed scheme, the decoding threshold is essentially defined by the Z operator decoder. Therefore, this confirms our hypothesis that the performance of an asymmetric CSS code, if it is aptly designed (the values

[7]The performance curves of the CSS code in Figure 7.8 have been simulated. Nonetheless, summing the WER/QBER of each constituent CSS decoder (the X and Z operator decoders) is also a valid method to obtain the performance curves of the overall code.

[8]Recall that the decoding threshold or waterfall region of an error correction code is the region where a sharp drop in the error rate takes place.

of m_1 and $[y_1, x_1, t_1]$ are chosen correctly), will be determined by the error correcting capabilities of its phase-flip decoder.

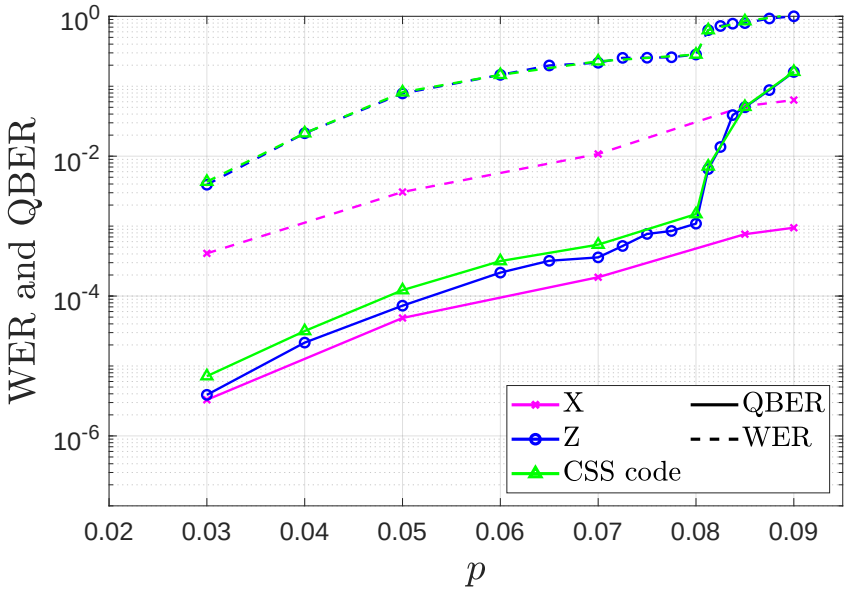


Figure 7.8: *Simulated WER and QBER for the constituent decoders of a CSS code. p represents the gross flip probability of the Pauli channel with asymmetry coefficient $\alpha = 10^2$.*

In short, this implies that the design of the best possible decoder for the proposed asymmetric CSS scheme can be approached through the separate optimization of its constituent Z and X decoders. This can be done by conducting simulations of different configurations of $[y_2, x_2, t_2]$ and m_2 that allow a sufficiently large value of m_1 or configuration of $[y_1, x_1, t_1]$ for which the bit-flip decoder exhibits few errors. In this manner, even if an equal distribution of X and Z errors is not obtained, we avoid the increased error-floor associated to bad X operator decoders while optimum performance, in terms of at what noise levels the waterfall region is entered, is achieved. Additionally, this serves to simplify matters, since by having shown that the performance of these asymmetric CSS codes is overwhelmingly determined by the quality of the phase flip decoder (assuming the the bit flip decoder is aptly built), we can now focus only on optimizing the parameter

configuration of a single decoder. With this goal in mind, in the sequel we simulate different configurations of m_2 and $[y_2, x_2, t_2]$ and determine which one results in the best performance.

7.2.4.2 Optimization of the Z decoder

In the previous subsection we showed that the performance of these asymmetric CSS codes is determined by the behaviour of its constituent decoders. A faulty X operator decoder can degrade performance by raising the error floor of the code, while its decoding threshold can change depending on the quality of the Z operator decoder. Having previously established that if the X operator decoder is designed appropriately performance is completely determined by the Z operator decoder, we now conduct simulations for various Z operator decoders in an attempt to discover the optimum values of m_2 and $[y_2, x_2, t_2]$.

Table 7.3: *Parameter values and configurations of Z operator decoders of asymmetric CSS codes simulated over a Pauli channel with $\alpha = 10^2$. The results of these simulations are shown in Figure 7.9.*

Decoder	m_2	t_2	x_2	y_2
Z1	12262	8198	5	3
Z2	12262	9010	6	3
Z3	12262	9507	6.9	3
Z4	11000	8497	8	3
Z5	11000	8810	9	3
Z6	11000	9060	10	3

Given the flexibility the design of these Z error decoders allows, it is important to provide structure to the simulation process. Earlier in this paper we mentioned that if x_2 could be shown to be a good indicator for the performance of the overall scheme, the design process could be reduced to simply finding the parameter configuration that would yield the optimum value of x_2. The value of x_2 is representative of its relationship with the parameter t_2, given that growth or reduction in one of these parameters will have the same effect on the other. In fact, when all the other parameters

are fixed, the only way we have to modify the value of x_2 is by changing t_2. At the same time, the parameter m_2 is intricately related to the value of x_2, which means that the relationship between these parameters will also play a part in the performance of the Z decoder. In an attempt to verify how good a performance indicator x_2 is and the nature of the relationship between this parameter and m_2, we test the Z decoder schemes detailed in table 7.3. These decoders differ in unit increments of x_2 while the value of m_2 is maintained equal as long as the design process permits[9]. The performance curves of these decoders are portrayed in Figure 7.9.

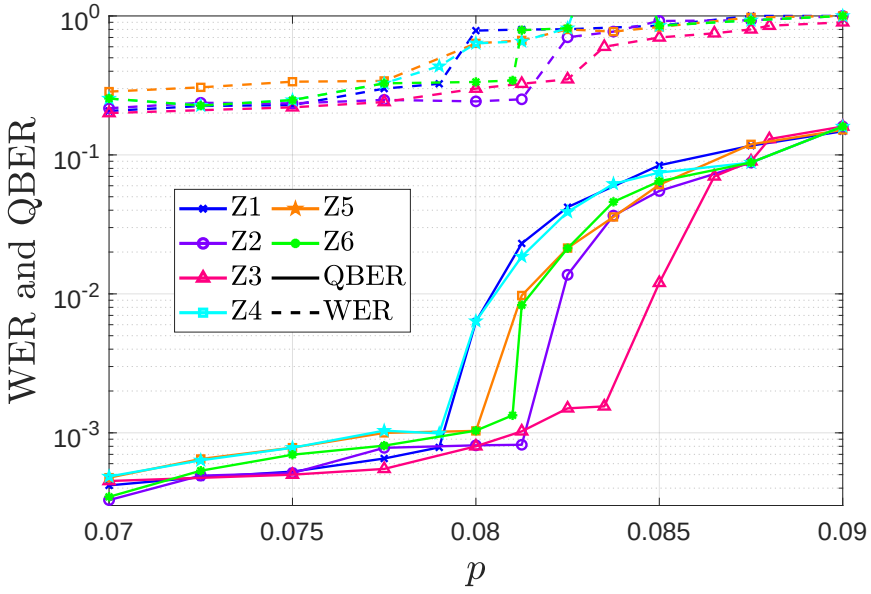


Figure 7.9: *Simulated WER and QBER for different CSS QLDGM schemes. p represents the gross flip probability of the Pauli channel with asymmetry coefficient $\alpha = 10^2$.*

The results shown in Figure 7.9 shed light on the relationship between m_2 and x_2, as well as how these parameters are linked to the performance of the decoder. For starters, consider decoders Z1, Z2, and Z3. All three

[9]Recall that for a specific value of m_2, given that $t_2 \leq \frac{N}{2}$, there is a maximum value of x_2 that can be obtained.

of them have the same value of m_2, but by selecting larger values of t_2, each scheme attains a higher value of x_2, with $x_2 = 6.9 \approx 7$ representing the largest possible value of the parameter for $m_2 = 12262$. The performance curves of these decoders show how the waterfall region is entered for subsequently higher values of p (the decoding threshold improves) as x_2 grows. For instance, code Z1 which has $x_2 = 5$, enters the waterfall region at roughly $p = 0.081$. In contrast, code Z3 which has $x_2 \approx 7$, enters the waterfall region at approximately $p = 0.085$. The same trend of performance improvement as x_2 becomes larger can be observed by looking at decoders Z4, Z5 and Z6. All three decoders have the same value of m_2 and the performance curves of decoder Z6, which has the largest value of x_2, are slightly better than those of its counterparts. In consequence, this outcome proves that for a given value of m_2 the decoder that will attain the best performance will be the one for which the value of x_2 is maximized. Notice that maximizing x_2 also means maximizing t_2, which, in this particular instance means that the negative effects associated to having larger degree s_B nodes (x_2 is maximized) are outweighed by the positive impact of having the largest possible amount of s_A nodes (t_2 is maximized). This is the exact opposite of what happened in the previous subsection when studying the bit flip decoder, where maximizing the parameter x yielded worse results.

Let us now compare the decoders of Table 7.3 in terms of their value of m_2. The results of Figure 7.9 show that performance of the decoder improves as the value of m_2 becomes larger. In fact, even though decoder Z6 has the highest value of x_2, it is outperformed by both decoders Z2 and Z3. These results present a new conundrum to which we must now give answer: which decoders will perform better, those with larger values of m_2 and lower values of x_2 or those with larger values of x_2 and smaller values of m_2? We can also formulate this question as: how is the trade-off relationship between x_2 and t_2 affected by the value of m_2? To analyze this, we simulate the Z decoders shown in table 7.4.

Figure 7.10 portrays the simulation results for the Z decoders of table 7.4. Performance is similar for all the simulated decoders, with decoder Z10 having the best decoding threshold. In terms of the relationship between x_2 and m_2, the curves shown in Figure 7.10 show that, up to a certain point, better performance is obtained by maximizing m_2 over x_2. In other words, increasing the number of syndrome nodes used to decode Z errors is more important, within certain limits, than trying to obtain the largest

value of x_2. This is reflected by decoder Z10, which has $m_2 = 12262$, outperforming decoders Z11 and Z12, which have $m_2 = 12676$ and $m_2 = 13262$. Therefore, the best performance of the proposed scheme over the asymmetric channel with $\alpha = 10^2$ is obtained by setting $m_2 = 12262$ and maximizing t_2 ($t_2 = 9507$ in this case) so that the largest possible value of x_2 is obtained for the selected m_2 value. This outcome tells us that increasing m_2 beyond a certain value has a negative impact on performance, despite the maximization of t_2 and x_2.

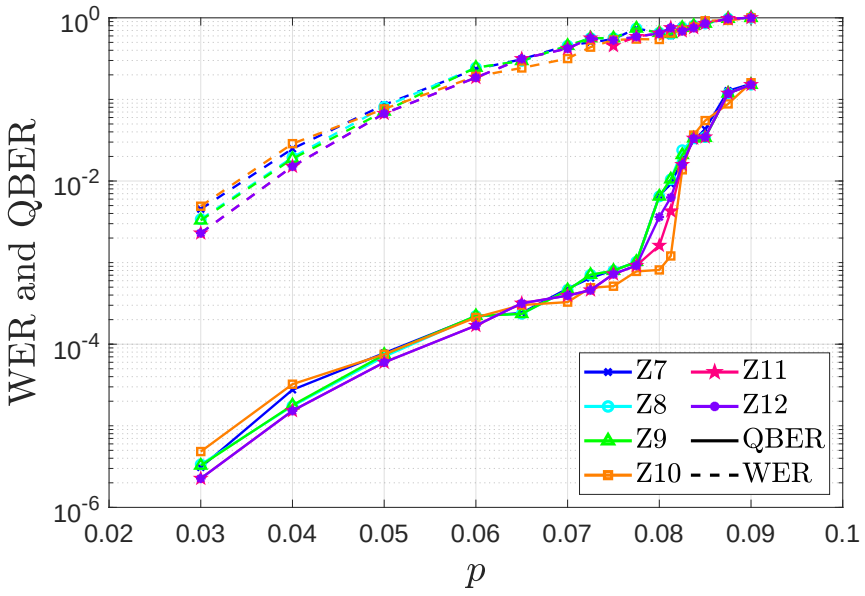


Figure 7.10: *Simulated WER and QBER for the decoders of table 7.4. p represents the gross flip probability of the Pauli channel with asymmetry coefficient $\alpha = 10^2$.*

A plausible cause for this is that when $m_2 > 12262$, there is an increased number of s_B nodes and a reduced percentage of degree-1 s_A nodes (since t_2 is bounded). A smaller percentage of s_A nodes means that a lower amount of perfect information will be propagated during initial decoding interations, which coupled with the increased number of s_B nodes, explains the degradation in the performance of the message passing decoding algo-

rithm for the $m_2 > 12262$ schemes. Nonetheless, the performance of all the decoders shown in Figure 7.10 differs by such a small margin that we can confidently state the following: *The best or near-best configuration of the proposed asymmetric CSS schemes is obtained by selecting $m_2 = \beta m$, where $\beta \in (0, 1)$, that allows a sufficiently large value of $m_1 = (1-\beta)m$ for the X operator decoder to function well. In terms of doping, t_2 should be maximized as $t_2 = \frac{N}{2}$, and $t_1 = 0.3m_2$.*

Table 7.4: *Parameter values and configurations of Z operator decoders of asymmetric CSS codes simulated over a Pauli channel with $\alpha = 10^2$. The results of these simulations are shown in Figure 7.10.*

Decoder	m_2	t_2	x_2	y_2
Z7	10410	9507	21	3
Z8	10600	8972	12	3
Z9	11232	9507	11	3
Z10	12262	9507	6.9	3
Z11	12676	9507	6	3
Z12	13262	9507	5	3

For $\alpha = 10^2$, from the decoders of table 7.4 we can ascertain that setting $0.73 \leq \beta \leq 0.9$ results in good performance, provided that x_2 is maximized by setting $t_2 = \frac{N}{2}$ once m_2 is chosen. We expect the value of the parameter β to vary with the degree of asymmetry of the channel, becoming larger as α grows. In terms of the value of t_1, the results of Figure 7.7 show that setting 30% of the m_1 nodes to be degree-1 syndrome nodes produces the best results. We test the validity of these statements for channels with larger degrees of asymmetry in the final subsection of this chapter. Prior to doing so, we show how the error floor of the CSS codes can be reduced by increasing the degrees of the underlying classical LDGM code, and we also show how the decoding threshold of our schemes can be improved by selecting larger values for the block length.

Error floor reduction

Most of the CSS and non-CSS QLDGM schemes we have seen throughout this dissertation are based on an underlying classical irregular LDGM code constructed through the parallel concatenation of two regular LDGM codes. The motivation behind such a construction is that the error floor of a single regular LDGM code can be substantially reduced when it is parallel-concatenated with another regular LDGM code of much higher degree. This results in an irregular configuration in which using a second LDGM code with larger degrees serves to lower the error floor of the whole scheme. However, as has been observed in [23], the use of larger degrees in the second code of the concatenation may worsen the decoding threshold. To study this behaviour in the context of asymmetric CSS codes we conduct simulations in which the optimum configuration of the CSS decoders devised for the Pauli channel with $\alpha = 10^2$ is employed: $[m_1, t_1, x_1, y_1] = [2000, 700, 11.03, 3]$ and $[m_2, t_2, x_2, y_2] = [12262, 9507, 6.9, 3]$.

As is done in [23], we utilize the irregular LDGM codes described by the concatenations P$[(8,8);(3,60)]$, P$[(8,8);(5,100)]$ and P$[(8,8);(8,160)]$. The simulation results are shown in Figures 7.11 and 7.12, where the best symmetric scheme of [155], [176] is included for comparison purposes.

It is easy to see from Figure 7.11 that increasing the degrees of the underlying irregular LDGM code lowers the overall error floor of the CSS code. Moreover, these results also show that as the degrees of the second regular LDGM code used in the parallel concatenation become larger, in accordance with what has been shown throughout the literature, the decoding threshold of the scheme (more visible in terms of the QBER) begins to deteriorate. Despite the slight worsening of the decoding threshold, the error floor yielded by the CSS code that uses $P[(8,8);(8,160)]$ (the irregular LDGM code with the largest degrees) is orders of magnitude better than for the other simulated concatenations, both in terms of the WER and the QBER. Hence, as occurs for quantum LDGM-based CSS codes designed for the depolarizing channel, using irregular LDGM codes of larger degrees is a valuable technique to improve the performance of codes designed for the general Pauli channel.

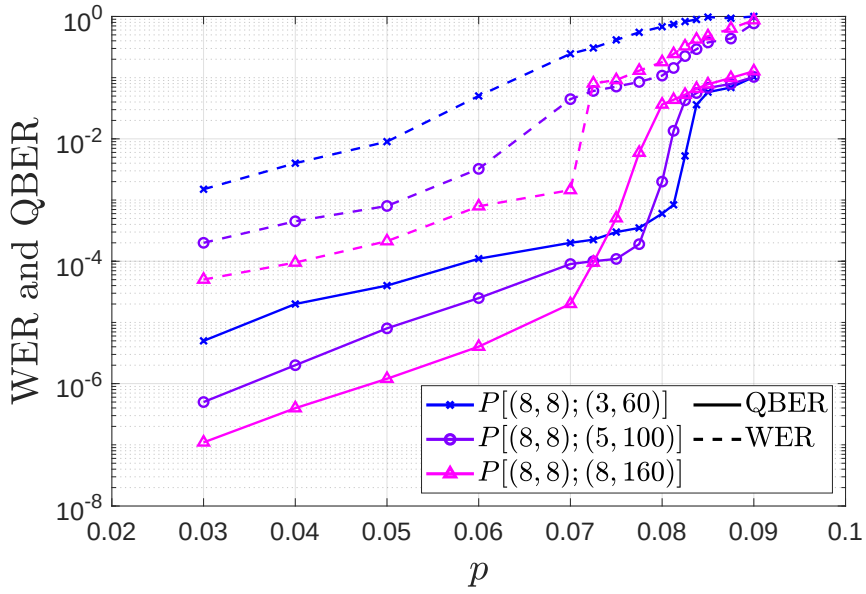


Figure 7.11: *Simulated WER and QBER for asymmetric CSS schemes with* $[m_1, t_1, x_1, y_1] = [12262, 9507, 6.9, 3]$, $[m_2, t_2, x_2, y_2] = [2000, 700, 11.03, 3]$. *Different degrees of the underlying irregular LDGM code have been tested.* p *represents the gross flip probability of the Pauli channel with asymmetry coefficient* $\alpha = 10^2$.

Additionally, we include Figure 7.12 to showcase the improvements provided by designing CSS codes specifically for the Pauli channel with $\alpha = 10^2$. In this figure, the performance curves of the best symmetric CSS QLDGM scheme in the literature are compared to those of our best asymmetric CSS QLDGM code. Both constructions are based on the parallel concatenation described by $P[(8,8);(8,160)]$ and both have block length $N = 19014$. Consider the decoding threshold of the symmetric CSS scheme: the code enters the waterfall region at approximately $p_{\text{sym}} = 0.0525$. The asymmetric scheme enters the waterfall region at $p_{\text{asym}} = 0.08$, which when compared to the symmetric code, is equivalent to an improvement of approximately 41 %.

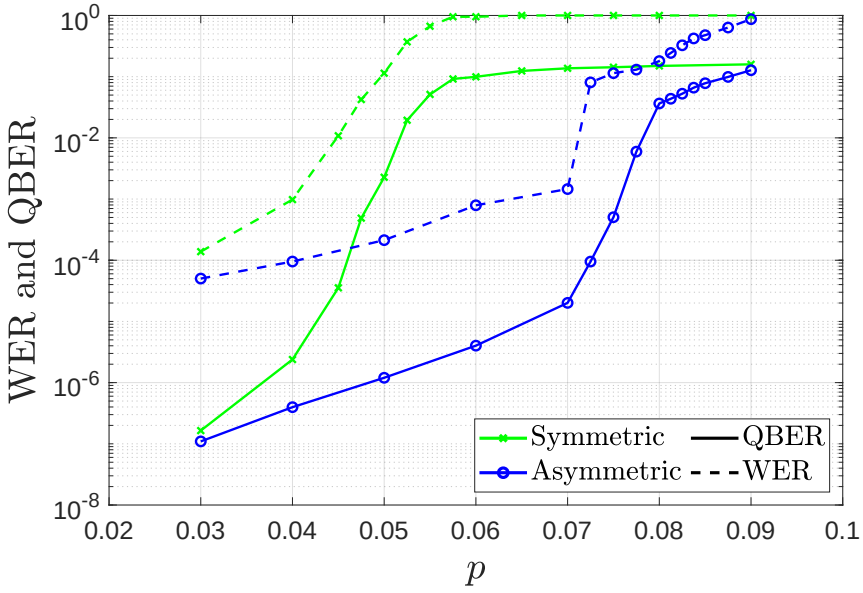


Figure 7.12: *Simulated WER and QBER for an asymmetric CSS scheme with* $[m_1, t_1, x_1, y_1] = [12262, 9507, 6.9, 3]$, $[m_2, t_2, x_2, y_2] = [2000, 700, 11.03, 3]$, *and a symmetric CSS scheme of the same block length* ($N = 19014$). *The degree of the underlying LDGM code is the same for both schemes* $\mathrm{P}[(8,8);(8,160)]$. p *represents the gross flip probability of the Pauli channel with asymmetry coefficient* $\alpha = 10^2$.

Decoding threshold improvements

We know from the work of Shannon [104] that the error rate of error correction codes vanishes asymptotically as their block length grows. Naturally, this means that real error correction codes will never achieve this performance as infinite block lengths cannot be employed in practice [183]. However, this also means that increasing the value of N may lead to performance improvements.

We close out this subsection by showing how an augmentation of the block length of our asymmetric scheme results in an improvement of its de-

coding threshold. For this purpose we compare the code with $[m_1, t_1, x_1, y_1] = [2000, 700, 11.03, 3]$ and $[m_2, t_2, x_2, y_2] = [12262, 9507, 6.9, 3]$ that uses the concatenation $P[(8, 8); (5, 100)]$ to its equivalent when the block length is doubled ($N = 2 \times 19014 = 38028$), i.e $[m_1, t_1, x_1, y_1] = [4000, 1400, 11.03, 3]$ and $[m_2, t_2, x_2, y_2] = [24524, 19014, 6.9, 3]$. Once again, the considered channel is the Pauli channel with $\alpha = 10^2$. The results are shown in Figure 7.13, where the betterment of the decoding threshold associated to a larger block size can be clearly observed.

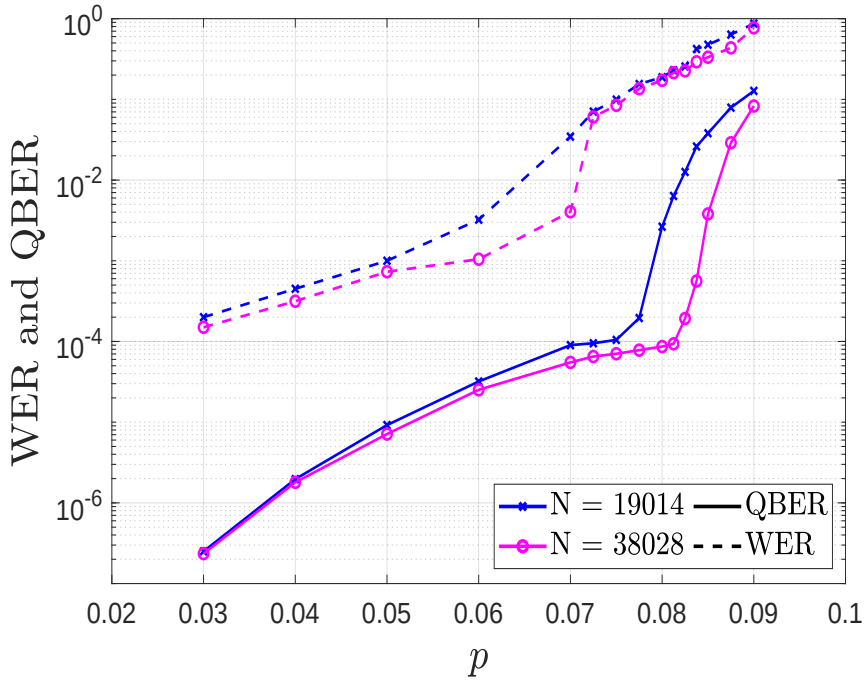


Figure 7.13: *Simulated WER and QBER for two asymmetric CSS codes with block lengths $N = 19014$ and $N = 39028$. p represents the gross flip probability of the Pauli channel with asymmetry coefficient $\alpha = 10^2$.*

7.2.4.3 Simulations and adaptation to other asymmetric parameter values

A matter that has yet to be discussed is the behaviour of our scheme over Pauli channels with different degrees of asymmetry. In [97, 192, 199], the values of the asymmetry coefficient $\alpha = [1, 10^2, 10^4, 10^6]$ are said to provide a realistic representation of practical quantum devices. Earlier in this work, we predicted that for larger degrees of asymmetry our proposed schemes would benefit from allowing more syndrome information to be used to decode Z errors (increasing the value of m_2). In a similar manner, this implies that for smaller degrees of asymmetry, the asymmetric CSS codes should provide more syndrome information to the X decoder. Essentially, asymmetric CSS QLDGM schemes should have larger Z operator decoding subgraphs and smaller X operator decoding subgraphs as the parameter α grows. To verify this hypothesis, we simulate different asymmetric CSS codes for the asymmetry coefficients $\alpha = [10, 10^4, 10^6]$. The particular configurations of the considered CSS codes are shown in table 7.5, while the performance of these codes over the Pauli channels with asymmetry coefficients $\alpha = [10, 10^4, 10^6]$ is shown in Figure 7.14.

Table 7.5: *Parameter values and configurations of the CSS codes simulated over Pauli channels with* $\alpha = [10, 10^4, 10^6]$. *The results of these simulations are shown in Figure 7.14.*

Code #	$[m_1, t_1, x_1, y_1]$	$[m_2, t_2, x_2, y_2]$
C1	$[2000, 700, 11.03, 3]$	$[12262, 9507, 6.9, 3]$
C2	$[1262, 300, 29.3, 3]$	$[12676, 9507, 6, 3]$
C3	$[1000, 100, 31.5, 3]$	$[13262, 9507, 5, 3]$

These results prove that our hypothesis is correct. For $\alpha = 10$, code C1 outperforms C2 and C3, but as alpha grows, the performance of C1 becomes increasingly degraded while that of C2 and C3 improves. This outcome is consistent with our initial hypothesis because for the smallest value of α that we have simulated, $\alpha = 10$, the code with the best performance is C1 which has the smallest value of m_2 (number of syndrome nodes used by the Z decoder), whereas for $\alpha = 10^6$ codes C2 and C3, which have larger values of m_2, overtake code C1. This behaviour is also congruous with the

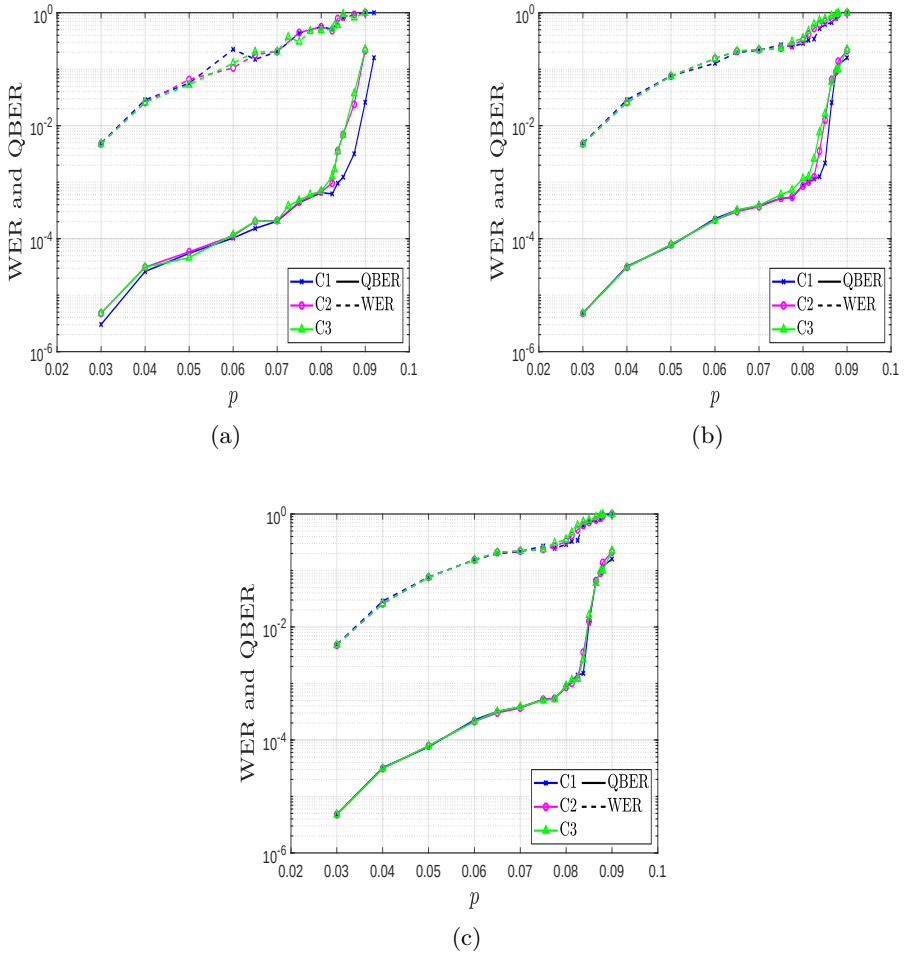


Figure 7.14: *Simulated WER and QBER for the asymmetric CSS schemes of table 7.5:* ***(a)*** *p represents the gross flip probability of the Pauli channel with an asymmetry coefficient $\alpha = 10$.* ***(b)*** *p represents the gross flip probability of the Pauli channel with asymmetry coefficient $\alpha = 10^4$.* ***(c)*** *p represents the gross flip probability of the Pauli channel with asymmetry coefficient $\alpha = 10^6$.*

fact that the hashing bound of a Pauli channel increases as this channel becomes more asymmetric [199].

However, the results of Figure 7.14 are somewhat surprising in the sense that, even for the most asymmetric instance that we have simulated $\alpha = 10^6$, the performance improvement provided by the more asymmetric codes, C2 and C3, is relatively small. In fact, for $\alpha = 10^4$ codes C2 and C3, which have larger values of m_2, do not outperform the optimal scheme (code C1) that was derived for a Pauli channel with $\alpha = 10^2$. Let us discuss why this happens.

It is clear from our initial simulation results (Figure 7.6) that asymmetric CSS schemes in which more syndrome information is used to decode Z operators perform better over a general Pauli channel than symmetric CSS codes. The extent to which m_2 must be increased has been thoroughly discussed throughout this section and has been shown to also be dependant on other characteristics of the asymmetric CSS construction. The entirety of this analysis has been performed considering a Pauli channel with $\alpha = 10^2$. We can now use the knowledge we have obtained for channels with $\alpha = 10^2$ to try and understand what happens over Pauli channels with $\alpha = 10^4$ and $\alpha = 10^6$.

In reality, the only difference between these channels lies in the error probabilities of the X and Z operators. For instance, if we assume a channel gross flip probability of $p = 0.075$, we will go from $p_z = 0.0735$ and $p_x = 7 \times 10^{-4}$ when $\alpha = 10^2$, to $p_z \approx 0.075$ and $p_x \approx 7 \times 10^{-6}$ when $\alpha = 10^4$, to $p_z \approx 0.075$ and $p_x \approx 7 \times 10^{-8}$ when $\alpha = 10^6$. These changes in the values of p_z and p_x as α grows, show why in Figure 7.14, the optimum code for $\alpha = 10^2$ performs as well as C2 and C3 when $\alpha = 10^4$ and $\alpha = 10^6$. The change in p_z when going from $\alpha = 10^2$ to $\alpha = 10^4$ or $\alpha = 10^6$ is too subtle to appreciate improvements[10] when increasing the value of m_2 from 12262 to $m_2 = 12676$ and $m_2 = 13262$. In stark contrast, when we go from $\alpha = 1$ to $\alpha = 10^2$, i.e, we compare our schemes to symmetric CSS codes, the improvements in performance when increasing m_2 are evident throughout the results provided in this paper. This happens because the change in p_z and p_x is substantial enough when going from $\alpha = 1$ to $\alpha = 10^2$, that changing $m_2 = 7131$ to $m_2 = 12262$ results in a palpable boost in performance. This also coincides with what is shown in [199], where the

[10] It may even slightly impinge on performance due to imperfect configuration of the other parameters of the scheme.

capacity of a Pauli channel grows sharply when the asymmetry coefficient goes from $\alpha = 1$ to $\alpha = 10^2$, but only a marginal capacity improvement is observed when α increases further beyond 10^2. Thus, the main reason for the almost negligible improvements that C2 and C3 provide with regard to C1 when $\alpha > 10^2$ stems from the fact that the change in the nature of the asymmetric channel when increasing α beyond 10^2 is too small to allow us to appreciate improvements in performance when increasing m_2 for the selected block length.

An interesting future research problem will be to study whether more drastic performance improvements can be obtained by increasing the block length of the code, as it may be that for a sufficiently large value of N (for the same quantum rate, increasing the block length implies an increase in the total number of syndrome nodes), for channels with values of $\alpha > 10^2$, the improvements provided by codes with larger values of m_2 when compared to the optimum scheme for the channel with $\alpha = 10^2$ will be more noticeable.

7.2.4.4 Distance to the Hashing bound of the Pauli channel model for asymmetry

We close this section by benchmarking the performance of our proposed schemes against the theoretical limits of the Pauli channel. The Hashing bound for a Pauli channel with asymmetry coefficient α [192] can be computed as

$$C_Q(p,\alpha) = 1 + (1-p)\log_2(1-p) + \left(\frac{2p}{\alpha+2}\right)\log_2\left(\frac{p}{\alpha+2}\right) + \left(\frac{\alpha p}{\alpha+2}\right)\log_2\left(\frac{\alpha p}{\alpha+2}\right).$$

We can use this expression to assess the distance to the Hashing bound for a specific rate and asymmetry coefficient. This is reflected in Figure 7.15, where the Hashing bounds for two Pauli channels with asymmetry coefficients $\alpha = 10$ and $\alpha = 10^2$ are shown alongside the points at which the best $R_Q = \frac{1}{4}$ asymmetric schemes[11] designed in the previous sections

[11] These schemes are defined by the parameters $[m_1, t_1, x_1, y_1] = [4000, 1400, 11.03, 3]$ and $[m_2, t_2, x_2, y_2] = [24524, 19014, 6.9, 3]$, $N = 38028$, and $P[(8,8);(8,160)]$.

for each of these channels can function with WER $= 10^{-3}$. As was done earlier in section 7.2.2, in Figure 7.15 we also show the highest possible coding rate at which the asymmetric codes that have been proposed in the literature can function with WER $= 10^{-3}$ over each of these asymmetric channels. These codes are:

1. The [[255, 159, $\frac{5}{17}$]] asymmetric QLDPC code of rate $R_Q \approx 0.624$ introduced in [206].
2. The [[1023, 731, $\frac{11}{33}$]] asymmetric QLDPC code of rate $R_Q \approx 0.714$ introduced in [201].
3. The $[[13, 1]]$ asymmetric short code of rate $R_Q \approx 0.077$ introduced in [200].

To provide further context, we also include the coding rates that can be achieved while maintaining WER $= 10^{-3}$ by the symmetric CSS QLDGM codes with $N = 19014$ of [155, 176] and the non-binary QC-QLDPC codes with $N = 20560$ of [145, 146] over the Pauli channels with asymmetry coefficients $\alpha = 10$ and $\alpha = 10^2$. The results for the symmetric CSS QLDGM codes have been obtained via Monte Carlo simulations (see Appendix C). The results for the codes of [145, 146] have been derived as follows.

As is mentioned in section 7.2.1, the most commonly employed noise model in the literature of CSS codes [18, 155, 176, 145, 146] approximates the action of a depolarizing channel by means of two independent BSCs with marginal bit-flip probabilities $f_m = \frac{2p}{3}$. Given the fact that each constituent code of a CSS scheme is decoded separately, the use of this model simplifies the simulation process because it only requires one of the constituent codes to be executed in most cases. This can be done because the error rate of the CSS code over the complete channel is computed as the sum of the error rates of each constituent code over each separate BSC, and considering the fact that each BSC will have the same flip probability, it will be possible to compute the performance of the scheme over the overall channel by simply obtaining the error rate of one of the constituent codes and summing it to itself.

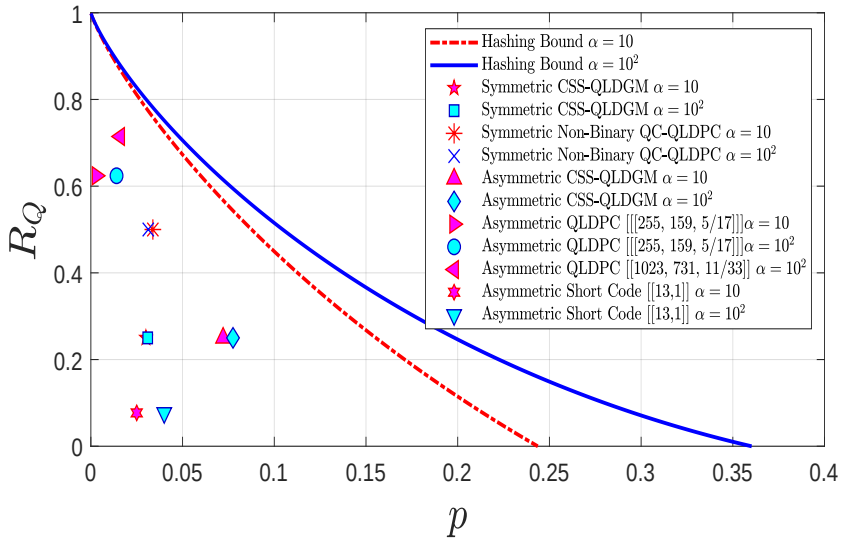


Figure 7.15: *Achievable coding rate at a WER of* 10^{-3} *for various QLDPC codes over Pauli channels with* $\alpha = 10$ *and* $\alpha = 10^2$.

We can use this framework to estimate the performance over the general Pauli channel model by adjusting the flip probabilities of the separate BSCs as $f_m^x = \frac{2p}{\alpha+2}$ and $f_m^z = \frac{p(\alpha+1)}{\alpha+2}$ [201, 206]. This means that each BSC serves as an X and Z error channel, respectively. Against this backdrop, we can compute the individual flip probabilities that each constituent code of the symmetric $R_Q = \frac{1}{2}$ code of [145, 146] would have to function at by substituting the value of p into the expressions that have been given for f_m^x and f_m^z. For instance, over a Pauli channel with $\alpha = 10$, for the same value of f_m at which the code performs with WER $= 10^{-3}$ over the depolarizing channel, its X error decoder will now have to operate at $f_m^x \approx 0.0077$ while the Z operator decoder has to function at $f_m^z \approx 0.0426$. These flip probabilities can then be used to obtain the WER of each constituent code of the non-binary QC-QLDPC CSS code from the results given in [145, 146]. This yields $\text{WER}_x \ll 10^{-5}$ and $\text{WER}_z \approx 10^{-1}$, where WER_x and WER_z denote the WER for the X and Z operator decoders of the CSS code, respectively. Finally, we can add these error probabilities to obtain the

overall WER over the complete asymmetric quantum channel[12], which in this particular case would be $\text{WER}_{\text{non-bin-QC-QLDPC}} = \text{WER}_z + \text{WER}_x \approx 10^{-1}$. Analogously, since $\alpha \geq 10$, p_z will be much larger than p_x and we will be able to compute the depolarizing probability at which the code will have $\text{WER} = 10^{-3}$ over the general Pauli channel by solving for p in $f_m^{\text{dep}} \approx f_m^z = \frac{p(\alpha+1)}{\alpha+2}$, where f_m^{dep} is the flip probability at which the code performs with $\text{WER} = 10^{-3}$ over the i.i.d. X/Z channel model.

This discussion proves that a CSS code designed for a symmetric channel over which the probability distribution for X and Z errors is identical will be unable to yield the same performance over an asymmetric channel. This is reflected in Figure 7.15, where a coding rate $R_Q = \frac{1}{2}$, which is achievable with the codes of [145, 146] over the depolarizing channel for a value of $p \approx 0.0465$, can now only be achieved for $p \approx 0.0338$ and $p \approx 0.0313$ when the corresponding asymmetry coefficient of the general Pauli channel is $\alpha = 10$ and $\alpha = 10^2$, respectively. The same phenomenon was exhibited by the symmetric CSS QLDGM codes in section 7.2.4.1, where performance was shown to be substantially degraded over the general Pauli channel.

As was done previously for the comparison over the depolarizing channel, we can use the distance to the Hashing bound[13], computed as shown in (7.5), to analyze the quality of the strategies shown in Figure 7.15. For instance, for the Pauli channel with asymmetry coefficient $\alpha = 10$, the proposed asymmetric CSS QLDGM scheme exhibits a distance to the Hashing bound of $\delta^{\alpha}_{\text{asym-CSS}} = 3.1$ dB. In the case of the non-binary QC-QLDPC codes of [145, 146] applied to this same channel, the distance to the Hashing bound is $\delta^{\alpha}_{\text{non-bin-QC-QLDPC}} = 4.2$ dB. This distance is $\delta^{\alpha}_{\text{sym-CSS}} = 6.66$ dB for the symmetric CSS QLDGM codes. Clearly, these outcomes showcase the improvements provided by building CSS designs specifically for the Pauli channel model for asymmetry. However, given that other families of QLDPC codes like those of [62, 145, 146] exhibit excellent performance over the depolarizing channel, it is likely that asymmetric adaptations of these codes may surpass the performance we obtain over asymmetric channels with the asymmetric CSS QLDGM codes proposed herein. Nonetheless, given the increased complexity of these error correction strategies, optimiz-

[12]Note that this procedure would be valid to approximate the performance of any symmetric CSS code over a Pauli channel with asymmetry coefficient α.

[13]We denote the distance to the Hashing bound of a Pauli channel with asymmetry coefficient α by δ^{α}. The superscript α is introduced to distinguish this distance measure from δ, the measure used for the depolarizing channel.

ing them for asymmetric quantum channels will be more complex than the schemes proposed in this work.

All in all, the discussion provided in this section along with the results that are portrayed in Figure 7.15 prove that the best symmetric CSS codes of [145, 146] and asymmetric codes that can be found in the literature are outperformed over asymmetric Pauli channels by the codes we have proposed in this chapter.

7.3 CHAPTER SUMMARY

In this chapter we have introduced a technique to design CSS quantum codes based on the use of the generator and parity check matrices of LDGM codes specifically for the general Pauli channel. The proposed methods are based on simple modifications to the upper layer of the decoding graph of a symmetric CSS QLDGM code designed for the depolarizing channel. For the block length used in this article, an asymmetric CSS code has been found for practical Pauli channels with different values of the asymmetry coefficient. Additionally, we have shown how for larger block lengths, the proposed asymmetric CSS codes can be further optimized based on the asymmetry coefficient α by increasing the block length of the code and the value of m_2 according to the guidelines provided in the paper. Over Pauli channels with $\alpha = 10$ and $\alpha = 10^2$, the proposed schemes are closer to the theoretical limit than other existing asymmetric codes and the best codes designed for the depolarizing channel.

CHAPTER 8

QTCs and Quantum Channels

"Somewhere, something incredible is waiting to be known"

Carl Sagan.

This chapter serves as a summary of the main results of other QEC-related publications that the author has been involved in during this PhD thesis. The chapter is divided into four different sections, each one dedicated to a different research topic:

- Section 8.1 studies the design of decoding protocols to improve the peformance of QTCs over realistic Pauli channels (those that include asymmetry). It reviews the primary contents and takeaways of [192]: J. Etxezarreta Martinez, P. Fuentes, P. M. Crespo, and J. Garcia-Frías," Pauli channel online estimation protocol for quantum turbo codes," *IEEE International Conference on Quantum Computing and Engineering (QCE20)*, 2020. doi: 10.1109/QCE49297.2020.00023.
- Section 8.2 discusses the derivation of classically-tractable quantum channel models that can be used to approximate and reenact the effects of decoherence. It provides a succint description of the motivation and results of [97]: J. Etxezarreta Martinez, P. Fuentes, P. M.

Crespo, and J. Garcia-Frias, "Approximating Decoherence Processes for the Design and Simulation of Quantum Error Correction Codes in Classical Computers," *IEEE Access*, vol. 8, pp. 172623-172643, 2020. doi: 10.1109/ACCESS.2020.3025619.

- Section 8.3 introduces the idea of time-varying quantum channel models, which serves as an overview of the work conducted in [191]: J. Etxezarreta Martinez, P. Fuentes, P. M. Crespo, and J. Garcia-Frias, "Time-varying quantum channel models for superconducting qubits," *npj Quantum Inf.*, vol. 7, no. 115, 2021. doi: 10.1038/s41534-021-00448-5.
- Section 8.4 further explores the concept of time-varying quantum channel models by studying the asymptotical error correction limits of these channels. This serves as a summary of the work conducted in [180]: J. Etxezarreta Martinez, P. Fuentes, P. M. Crespo, and J. Garcia-Frias, "Quantum outage probability for time-varying quantum channels," submitted to *Phys. Rev. A*, 2021. arXiv: 2108.13701.

Before moving onward, it must be mentioned that the contents shown herein are meant only as a cursory overview. Readers should refer to Josu Etxezarreta Martinez's PhD dissertation (first-author of this research) or the journal articles themselves for a complete discussion regarding these findings.

8.1 ONLINE ESTIMATION PROTOCOL FOR QTCS OVER PAULI CHANNELS

As was shown previously in Chapter 7, the assumption that information regarding the noise level of a quantum channel is available prior to decoding does not generally hold in real QEC scenarios. In practice, quantum error correction strategies face the phenomenon of channel mismatch, where the noise level of the channel must be estimated prior to decoding. Additionally, we also mentioned how the noise suffered by some quantum devices does not usually exhibit the symmetry described by the depolarizing channel. Instead, general Pauli channels must be employed to provide an accurate representation of the real behaviour of these quantum devices. This means that, although QEC techniques are generally evaluated using the depolarizing channel under the assumption of perfect channel knowledge, in some instances, QEC codes must actually be studied (and achieve

good performance) over asymmetric quantum channels where prior knowledge of channel parameters is unavailable.

We tackle this issue in [192], where we address the channel estimation problem for realistic quantum channels when using QTCs. This involves applying the online estimation protocol for QTCs derived for the depolarizing channel in [31] to decode over Pauli channels that accurately represent the asymmetric behaviour of realistic quantum devices. This online estimation protocol achieves excellent performance without the need for offline channel estimation, which, as we also discussed in Chapter 7, requires additional resources and reduces the rate of the QEC strategy.

In [192], we apply the aforementioned online estimation technique for QTCs to the Pauli channel model for asymmetry shown in section 7.2.1. This is achieved by modifying the online estimator to account for the asymmetry coefficient of the channel and produce estimates of its value. The estimates of the channel asymmetry coefficient can then be used to compute the individual probabilities p_x and p_z and begin the decoding process. As is shown in Figure 8.1, the online estimation technique that accounts for asymmetry attains the same performance as that obtained under a perfect channel knowledge scenario. This means that, as was the case for the depolarizing channel, the online technique achieves excellent performance over asymmetric quantum channels without the need for offline channel estimation.

8.2 APPROXIMATING DECOHERENCE PROCESSES FOR THE DESIGN OF QECCS ON CLASSICAL COMPUTERS

The design and optimization of QEC codes requires the use of an error model, typically in the form of a quantum channel, that accurately represents the decoherence processes that affect quantum information. Based on these error models, appropriate strategies to combat the effects of decoherence can be derived. In order for these QECCs to be applicable in realistic quantum devices, quantum channels should capture the defining characteristics of the physical processes that result in qubit decoherence. From what we have seen throughout this dissertation, we know that the depolarizing channel is the most widespread decoherence model used to evaluate the error correcting abilities of QECC families. This decoherence model is especially useful because it satisfies the Gottesman-Knill theorem

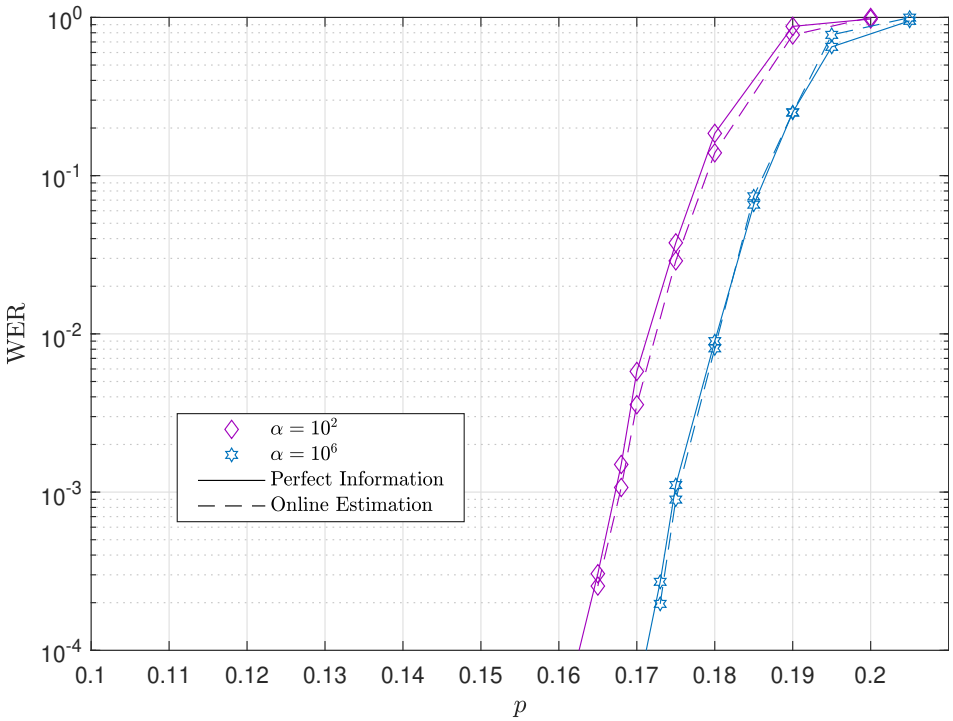


Figure 8.1: *Performance of a QTC over Pauli channels with asymmetry coefficient $\alpha = 10^2$ and $\alpha = 10^6$ under the perfect channel knowledge assumption and when using the online estimation strategy.*

(see section 3.1.4.5), which makes it possible to efficiently simulate it on a classical computer. The reduced qubit-count and limited accessibility of modern quantum computers makes it implausible to simulate decoherence models via quantum means, which implies that having a classically tractable model for decoherence is quite a momentous result. Thus, classical resources remain an invaluable tool for the design of advanced QECCs, even those that will be used beyond the Noisy Intermediate-Scale Quantum (NISQ) era[1] [207].

[1]The NISQ era, a term coined by famous physicist John Preskill, makes reference to the time when quantum computers will be able to perform tasks that are out of the reach of classical computers, but will still be too small (in qubit number) to provide fault-tolerant implementations of quantum algorithms.

The motivation behind our work in [97] is to provide a clear description of how the physical processes that lead to the corruption of quantum information (decoherence processes) can be approximated so that they are tractable using classical resources. For this purpose, our work surveys the techniques that have been presented in the literature to approximate general quantum channels by the widely used Pauli channels. We commence with a review of the decoherence processes that compromise the integrity of quantum information by studying the mathematical description/abstraction of these processes into quantum channels. Then, we apply the technique known as *twirling* [208, 209, 210], which enables us to obtain classically tractable approximations of those quantum channels and allows us to derive the asymmetric Pauli channel and the depolarizing channel. During this process we also prove that any error correction methods designed for the twirled approximated channels will also be valid for the more realistic original channels. Furthermore, we provide an extensive discussion regarding the Pauli channel and its symmetric instance, the depolarizing channel, and show why these approximations are so widely used for QECC construction/simulation with classical resources. Additionally, even though the memoryless version is almost always considered, we also discuss the emergence and impact of memory effects on those channels [211, 212]. Finally, we present the way in which twirled approximations are implemented on a classical computer and discuss how they can be employed to simulate the performance of QECCs.

All in all, the contributions of our work in [97] can be summarized as follows. Primarily, we explain in a clear manner how it is possible to construct and evaluate certain families of quantum error correction codes without the explicit need for a quantum computer (the error models and quantum codes can be entirely simulated in the classical domain). Moreover, we describe how classical computers can be used to simulate the performance of the aforementioned quantum error correcting schemes when they are exposed to the effects of general decoherence models. The key here is that decoherence models can be approximated as Pauli channels, which can be implemented with classical resources, and that the performance of quantum error correcting codes over the approximated channels is equivalent to that obtained for the original decoherence model.

8.3 TIME-VARYING QUANTUM CHANNEL MODELS FOR SUPERCONDUCTING QUBITS

Error models that accurately describe the decoherence processes that corrupt quantum information are a necessity when seeking to construct appropriate quantum error correcting codes. The best known quantum channel models for decoherence are the amplitude damping channel ξ_{AD} and the combined amplitude and phase damping channel ξ_{APD}. However, such channels cannot be efficiently simulated in a classical computer for qubit counts that exceed a small limit. As was mentioned in the previous section, the quantum information theory technique known as twirling allows us to approximate ξ_{AD} and ξ_{APD} by the classically tractable family of Pauli channels. It is important to note that the dynamics of the original quantum channels depend on the qubit relaxation T_1 and dephasing times T_2. Naturally, this dependence will be inherited by the Pauli channel family obtained by twirling the original quantum channels. Throughout the literature, T_1 and T_2 are considered to be fixed parameters, i.e, that they do not fluctuate over time, which implies that the noise dynamics suffered by the qubits in a quantum device are identical for each quantum information processing task.

Unfortunately, the assumption that T_1 and T_2 are time invariant has been disproven in recent experimental studies that analyzed the stability of qubits in superconducting quantum processors [213, 214, 215, 216]. The results pesented in these studies show that the relaxation and dephasing times of superconducting qubits can vary by up to 50% of the mean value of the sample data, which strongly suggests that the dynamics of the decoherence effects change drastically as functions of time. Therefore, if we wish to assess the performance of QEC codes accurately, quantum channel models must somehow incorporate the time-variations suffered by the parameters that define their dynamics.

In our work [191], we amalgamate the findings of [213, 214, 215, 216] with the existing models for quantum noise, which culminates in the proposal of Time-Varying Quantum Channels (TVQCs) $\xi(\omega, t)$ for superconducting qubits. The dynamics of these models are defined by the random processes $T_1(\omega, t)$ and $T_2(\omega, t)$ whose behaviour is described by the decoherence governing parameters of existing superconducting quantum computers. Against this backdrop, we use a metric known as the diamond

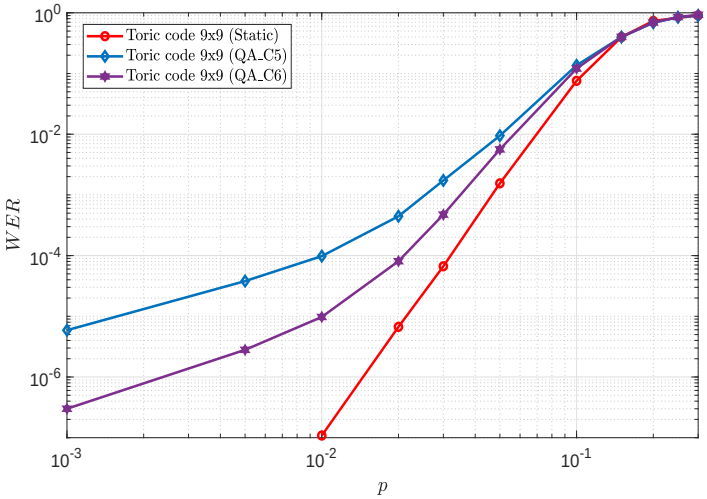
norm distance[2] [217, 218] to assess the difference between the commonly employed static channels and our proposed TVQCs. The results shown in [191] prove that neglecting the fluctuating nature of the relaxation and dephasing times in the construction of noise models provides an unrealistic (overly optimistic) portrayal of decoherence effects. This can also be seen in Figure 8.2, included to provide context, which shows how the performance of QEC codes deteriorates when the time fluctuations of the decoherence parameters are accounted for in the quantum channel model (note how the WER is significantly worse over the time varying channels, QA_C5 and QA_C6, than over the static channel).

In summary, the primary takeaway from our work in [191] is that TVQC models are necessary to construct and appropriately simulate/predict the performance of QECCs. Additionally, despite the fact that the derived time-varying channel models are based on the statistical experimental characterization of the parameters T_1 and T_2 of [213, 214, 215], they are also applicable to any superconducting quantum processor whose decoherence parameters exhibit slow fluctuations. In fact, the model is actually also applicable to any quantum-coherent two-level system that presents similar time dependencies, regardless of its physical implementation.

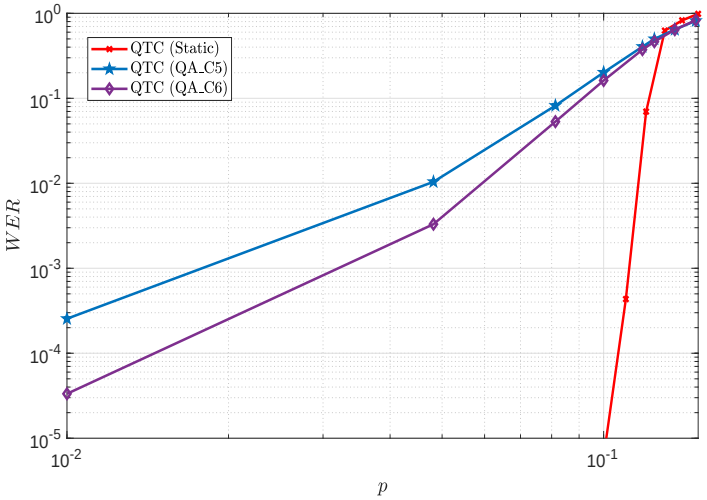
8.4 QUANTUM OUTAGE PROBABILITY FOR TIME-VARYING QUANTUM CHANNELS

TVQCs allow us to account for the experimentally validated time-variance of the relaxation and dephasing times of superconducting qubits and enable us to study and predict the performance of QEC codes with higher precision than with previously existing channel models. However, because the TVQC framework includes the time dependence of T_1 and T_2, the asymptotical error correction limits associated to time-varying channels will be different to the limits of those channels that assume time independence of the aforementioned decoherence defining parameters. In [180], we take on the challenge of studying the asymptotical limits of error correction in the TVQC paradigm. To do so, we draw from the well-known classical scenario of slow/block fading and introduce the concepts of the quantum outage probability and the quantum hashing outage probability of a TVQC.

[2]The diamond norm $||\xi_1 - \xi_2||_\diamond$ provides a measure of how different two quantum channels are, where ξ_1 and ξ_2 denote two quantum channels.



(a)



(b)

Figure 8.2: *Performance of the $d \times d$ Kitaev Toric codes of [32] with $d = 9$ and the QTC of [192] over a static quantum channel and two different time-varying quantum channels.* ***(a)*** *9×9 Toric code.* ***(b)*** *QTC.*

In the context of classical communications, we know from the Shannon channel coding theorem [104] that it will be possible to achieve reliable communications with a code of rate R_c over an arbitrary channel if the realization of the channel capacity $C(\omega)$ is larger than R_c. On the contrary, when for a particular realization of the channel capacity we have $C < R_c$, communication at a low probability of error will not be possible. Thus, we refer to the probability that communications fail when transmitting a codeword with rate R as the outage probability and we define it as

$$p_{\text{out}}^{c}(R_c) = \{\omega \in \Omega : C(\omega) < R_c\}.$$

We can define the quantum outage probability for a TVQC by following this same reasoning. Knowing that the qubit relaxation and dephasing times are modelled as random variables[3], $T_1(\omega)$ and $T_2(\omega)$, realizations of the relaxation and dephasing times will result in a realization of the TVQC in question, and consequently, in a specific value for the quantum channel capacity C_Q. Reminiscent of the classical scenario, if the realization of the decoherence parameters leads to a channel capacity that is lower than the quantum coding rate, R_Q, then the error probability (the QBER or the WER) will not vanish asymptotically with the block length, independently of the selected QEC code. For such realizations, the channel can be said to be in outage. Therefore, we can write

$$p_{\text{out}}^{Q}(R_c) = \{\omega \in \Omega : C_Q(\omega) < R_Q\}.$$

The above expression tells us that, with probability $p_{\text{out}}^{Q}(R_c)$, the capacity of the channel will be lower than the rate of the quantum code and so reliable communication will not be possible. Conversely, with probability $1 - p_{\text{out}}^{Q}(R_c)$ reliable communication over the TVQC will be possible. Thus, the quantum outage probability of a TVQC is defined as the asymptotically achievable error rate of a QEC code with quantum rate R_Q that operates over said TVQC.

[3]They are actually modelled as random processes, i.e, $T_1(\omega, t)$ and $T_2(\omega, t)$, but because the coherence time of a qubit is generally longer than the processing time of a quantum algorithm, realizations of these random processes can be considered to remain constant during each processing round and can thus be modelled as random variables $T_1(\omega)$ and $T_2(\omega)$ [191].

In [104] these ideas are extended and, based on the results of [191, 213], closed-form expressions for known TVQCs, such as the time-varying amplitude damping channel (TVAD), the time-varying amplitude damping Pauli twirl approximated channel (TVADPTA), and the time-varying amplitude damping Clifford twirl approximated channel (TVADCTA), are provided. This work also thoroughly analyzes the quantum outage probability and quantum hashing outage probabilities of the aforementioned TVQCs in different scenarios. The work conducted in [104] concludes by simulating the performance of QTCs over the considered channels and benchmarking them against the derived information theoretic limits. An example of this can be seen in Figure 8.3, which plots the quantum outage probability against the damping parameter γ of the TVAD channel[4]. This figure clearly shows how the probability of the channel being in outage increases with the variability[5] of $T_1(\omega)$.

[4]In the AD channel the damping parameter defines the noise dynamics of the channel similarly to how the depolarizing probability determines the noise level of the depolarizing channel.

[5]By variability we refer to how much realizations of T_1 can differ compared to its mean. This is given by the coefficient of variation $c_v = \frac{\sigma}{\mu}$, where μ and σ, are the mean and variance of the random variable in question.

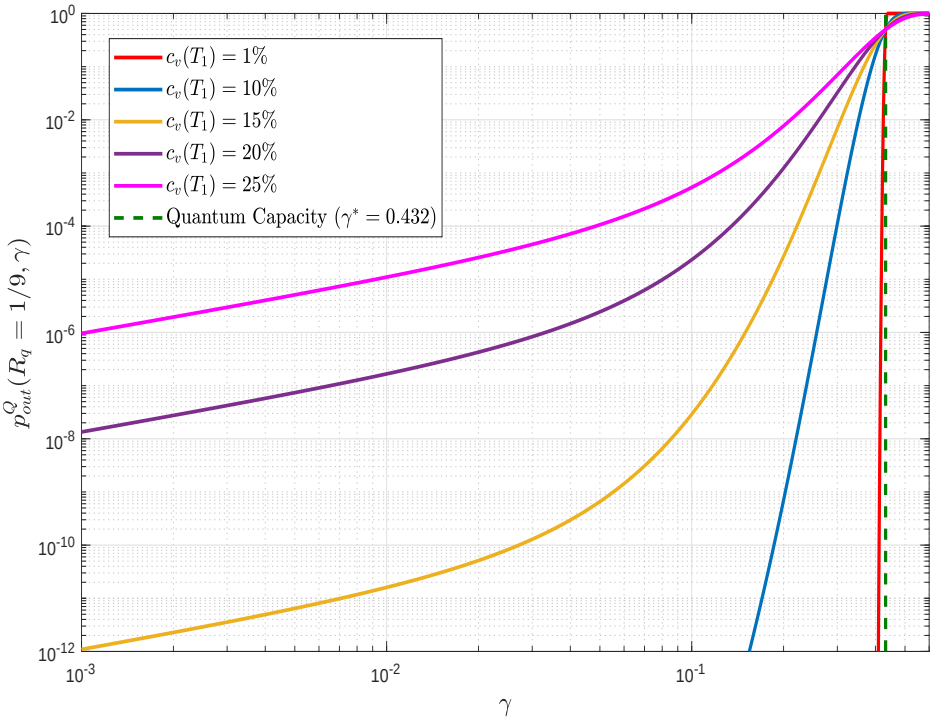


Figure 8.3: *Quantum outage probability of the TVAD channel. TVADs with* $c_v(T1) = \{1, 10, 15, 20, 25\}$ *assuming a rate* $R_Q = 1/9$ *have been considered for the simulations.*

CHAPTER 9

Conclusion and Future Work

"Science never solves a problem without creating ten more"
George Bernard Shaw.

This thesis set out with the objective to ponder and analyze the following two aspects of sparse QEC codes:

- how to improve the performance of CSS and non-CSS QLDPC codes in various different scenarios.
- how to characterize the manifestation of the "quantum-exclusive" phenomenon known as degeneracy and its impact on the performance of sparse quantum codes.

With these goals in mind, the dissertation began with an introduction to the realms of quantum computing, quantum information, and classical error correction. This preliminary part of the thesis commenced in Chapter 2, where we showed how quantum computing can be used to address problems that cannot be solved efficiently on classical machines and we presented various different technologies that are currently being employed to build quantum processors. Then, in Chapter 3, we presented background

concepts on quantum information and classical error correction, such as quantum channels and the factor graph representation of classical codes, that are critical to follow the rest of our discourse.

Following this, we entered the part of the thesis that focuses on understanding degeneracy and its relationship to sparse quantum codes. In Chapter 4 we presented a group theoretic interpretation of stabilizer-based error correction that allowed us to accurately define degeneracy and classify its effects on the decoding algorithm applied to sparse quantum codes. This chapter also included a numerical example that helped to further portray what degeneracy is and how it is involved in the decoding process of stabilizer codes. Based on this new perspective, in Chapter 5 we studied the ways in which the performance of sparse quantum codes has been evaluated in the literature, and we showed how this can sometimes lead to an overestimation of the error rate. Furthermore, we discussed previously existing methods to compute the logical error rate of sparse quantum codes and we also proposed our own classically-inspired strategy for this same purpose. We closed this chapter by discussing the most relevant improved decoding strategies for sparse quantum codes that have appeared in the literature.

After Chapter 5, we entered the second major part of the thesis, which concerned itself with the analysis and optimization of the performance of LDGM-based quantum codes in a variety of different scenarios. We started with the derivation of a non-CSS construction in Chapter 6, where we showed how it is possible to improve the performance of known CSS LDGM-based codes by rearranging the structure of their corresponding factor graphs. We also showed how the proposed non-CSS codes outperformed other existing codes of comparable complexity. We followed this up by studying the impact that the channel mismatch phenomenon has on this family of non-CSS LDGM-based codes in Chapter 7. This analysis showed that having no prior knowledge of the channel led to a worsening of the performance of our codes. To face this, we derived an online estimation strategy similar to one that was previously used for QTCs. We finished this chapter by studying the performance of CSS LDGM-based codes in the context of asymmetric quantum channels, which have been shown to represent specific quantum devices with more accuracy than the depolarizing channel. In the final chapter of the dissertation, Chapter 8, we provided a brief overview of other QEC-related works that the author has been involved in during his time as a PhD student.

All in all, in this thesis we have seen how degeneracy is intricately related to the performance of quantum codes and we have studied and improved the performance of a specific family of sparse quantum codes in different communication scenarios. At the same time, we have also seen how there is ample room for growth, progress, and (obviously) work, in both of these areas. Consequently, we close this chapter (and the dissertation itself) by entertaining some of the questions and potential research topics that our work herein has left unanswered.

- **Design of degenerate sparse quantum codes.**

Having shown that degenerate errors do not actually alter the codespace of a stabilizer code and that they do not result in decoding mistakes, it is clear that degeneracy can have a positive impact on the performance of sparse quantum codes. Thus, it is likely that constructing quantum codes with a maximized probability of suffering degenerate errors will lead to improvements in performance. An added benefit of such a strategy is the fact that the performance gain would come at no cost to the decoding complexity, which cannot be said for other performance improvement methods. In the literature, most of the research related to the exploitation of degeneracy has focused on the decoder. The best example of this can be found in [154] were the performance of particular sparse quantum codes is improved by building a decoder that can take advantage of degeneracy. However, this decoder-focused approach differs significantly from our suggestion of specifically designing codes to be degenerate. For this reason, it may be difficult to find inspiration in the literature and define a starting point for our idea. Based on intuition, one could begin by studying the coset structure of a short stabilizer code using the degenerate error detecting methods discussed in Chapter 5 and then attempting to re-organize it so that the lowest weight operators all fall within the coset with highest probability. If successful, we could then try to generalize this approach to larger codes.

- **Improvement and design of modified decoding strategies.**

The idea of improved decoding strategies for sparse quantum codes has been mentioned numerous times throughout this thesis. In Chapter 5 we saw how these methods can be used to reduce the number of end-to-end errors with different syndromes, albeit at the cost of an increase in the decoding complexity. However, some of these schemes can only be applied to CSS quantum codes. Therefore, it is likely that the adaptation and application of these strategies to the non-CSS codes derived in Chapter 6 will lead

to performance improvements. Furthermore, because the construction of a "degenerate" decoder for sparse quantum codes remains an open problem in QEC, another potential line of work would involve seeking ways in which to modify the SPA decoding algorithm of sparse quantum codes to account for the degeneracy phenomenon.

• Construction & analysis of QEC codes on real quantum computers.

A topic that we have rarely mentioned in this dissertation is the construction of practical error correction schemes on existing quantum computers. This is mostly due to the fact that sparse quantum codes generally require large block lengths (over 1000 qubits) to provide noticeable improvements in performance; a prohibitive requirement, since currently existing quantum computers barely exceed a count of 100 physical qubits. In other words, practical QEC strategies for the present must function for significantly smaller block lengths than those we have considered here. This opens up various different research options that remain somewhat unexplored in the literature. One possibility would be to study and optimize shorter block length sparse code constructions and then seek to derive stabilizer encoding circuits from their corresponding QPCMs. Despite the suboptimal performance of shorter block length sparse quantum codes, deriving a valid method to obtain the encoding circuit from the corresponding QPCM would bridge an important gap between theoretical and practical QEC. QEC codes are physically implemented using quantum circuits and a strategy capable of deriving them from QPCMs represents an important advancement for both present and future QEC methods. Following this, and conditioned by the number of available qubits, these encoding circuits could be simulated on cloud-accessible quantum computers such as IBM-Q. Another possibility would be to look at these same problems from the perspective of different quantum code families, like surface codes, instead of sparse quantum codes.

• Sparse quantum codes over TVQCs.

In Chapter 8 we saw that TVQCs provide a more accurate portrayal of the decoherence-induced noise experienced by superconducting qubits. Based on what we have seen in this dissertation, it would be interesting to benchmark the performance of CSS and non-CSS QLDGM codes over these channels and to study them based on the view provided by the channel outage probability.

Appendices

APPENDIX A

Syndrome-based Decoding of LDPC codes

Sum-product based decoding of an LDPC code can be easily understood by describing how the procedure unfolds over the factor graph associated to the PCM of the LDPC code under consideration. Assume that the receiver has obtained the syndrome $\mathbf{s} = \mathbf{H}_c\mathbf{x}^\top$, where $\mathbf{H}_c$ is the (sparse) PCM of the LDPC code and $\mathbf{x}$ is the channel output. Aside from the syndrome, an SPA-based decoder requires information regarding the communication channel. It is commonplace in many classical and quantum coding scenarios to assume that the receiver knows the probability distribution of the channel, P($\mathbf{e}$). In reality, practical communication systems must somehow perform channel estimation to estimate the noise level of the channel. This has been studied in both the classical domain [223, 224, 225] and in the quantum paradigm [31, 192, 185, 186, 160], and we also discuss it in depth in Chapter 7. For the sake of simplicity, in what follows we will simply assume that the probability distribution of the channel is known to the receiver. Under this premise, we can now address how the sum product algorithm is executed over the corresponding factor graph. As mentioned in Chapter 3, the SPA attempts to estimate the symbol-wise most likely channel error by computing the solution to

$$\begin{aligned} \hat{\mathrm{e}}_j^{\mathrm{SW}} &= \arg\max_{\mathrm{e}} P(\mathrm{e}_j = \mathrm{e}|\mathbf{z}) \\ &= \arg\max_{\mathrm{e}_j \in \{0,1\}} \sum_{\mathrm{e}_1,\ldots,\mathrm{e}_{j-1},\mathrm{e}_{j+1},\ldots,\mathrm{e}_N} P(\mathrm{e}_1,\ldots,\mathrm{e}_N|\mathbf{z}). \end{aligned} \tag{A.1}$$

To do so, the algorithm operates by exchanging messages between the variable nodes and the parity check nodes of the factor graph associated to the PCM of the LDPC code in question. Assuming that the graph has no cycles, the solution provided by the SPA agrees with the solution of (A.1).

In order to simplify the description of the SPA, assume that the selected LDPC code is binary and that communications take place over a Binary Symmetric Channel (BSC) with crossover probability[1] q. In this context, the sum product algorithm can be summarized into the following set of sequential steps:

1. **Initialization:** In order to begin, the algorithm must be initialized with channel information. This is done by computing the message pair $(m^0_{\mathrm{e}_i\to\mathrm{c}_j}, m^1_{\mathrm{e}_i\to\mathrm{c}_j})$ for each variable node on the factor graph, where e_i denotes the i-th variable node[2], c_j denotes the j-th variable node, and the subscript $\mathrm{e}_i \to \mathrm{c}_j$ denotes the message transmitted from variable node e_i to check node c_j. In this notation, the superscripts represent the value of the i-th component of the error sequence, i.e., a message $m^x_{\mathrm{e}_i\to\mathrm{c}_j}$ can be understood as the "belief" that variable node e_i sends to parity check node c_j regarding the certainty it has of being equal to $\mathrm{x} = \{0, 1\}$. During this initialization phase the receiver only knows the distribution probability of the channel, and so these beliefs are simply equal to the BSC flip probabilities:

$$\begin{aligned} m^0_{\mathrm{e}_i\to\mathrm{c}_j} &= P_{\mathrm{BSC}}(\mathrm{e}_i = 0) = 1 - q, \\ m^1_{\mathrm{e}_i\to\mathrm{c}_j} &= P_{\mathrm{BSC}}(\mathrm{e}_i = 1) = q. \end{aligned} \tag{A.2}$$

   As code block lengths increase, the computation of these message pairs for each node of the factor graph can become increasingly cumbersome. For this reason, it is common to use Log-Likelihood Ratios (*llrs*). By defining channel *llr*s during this initialization process, each variable node need only compute and send a single message to the check nodes it is connected to. As a result, the algorithm computes

[1] A bit transmitted over a BSC with crossover probability q will be flipped with probability q and will remain unchanged with probability $1 - q$.

[2] Since each variable node is associated to a component of the error vector, we extend our notation (regular lower case romans) to denote the variable nodes.

and provides each variable node e_i with its corresponding channel *llr*

$$l_{\mathrm{ch}}(\mathrm{e}_i) = \log\left(\frac{m^0_{\mathrm{e}_i\to\mathrm{c}_j}}{m^1_{\mathrm{e}_i\to\mathrm{c}_j}}\right) = \log\left(\frac{1-q}{q}\right).$$

Subsequently, each variable node transmits the message $\mu^1_{\mathrm{e}_i\to\mathrm{c}_j} = l_{\mathrm{ch}}(\mathrm{e}_i)$, to all of the parity check nodes it is attached to. The superscript now denotes that this exchange occurs in the first algorithm iteration, which is logical given that it takes place during the initialization step. This process is shown in Figure A.1.

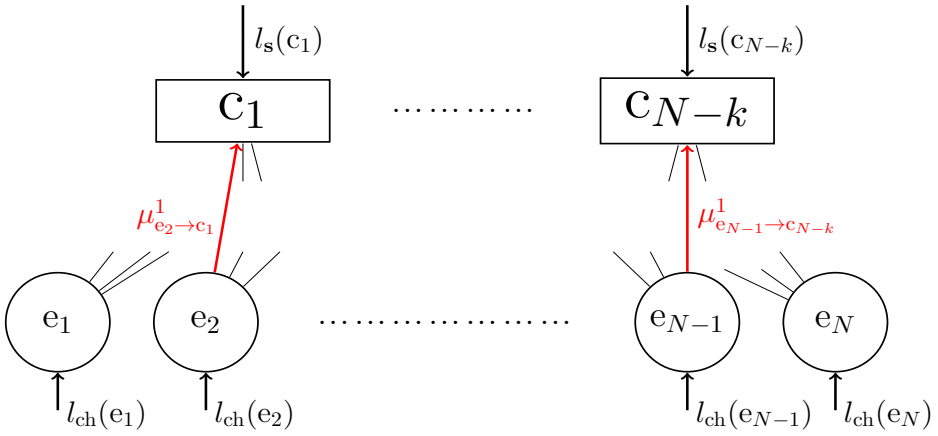


Figure A.1: *Initialization of the sum-product algorithm and transmission of the first messages from variable nodes to parity check nodes.*

2. **Parity check node to variable node messaging:** Aside from the channel *llr*s that they receive from variable nodes, each parity check node c_i is also associated to a component s_i of the obtained syndrome $\mathbf{s} = [\mathrm{s}_1 \dots \mathrm{s}_{N-k}]^\top \in \mathbb{F}_2^{N-k}$. The procedure is similar to the one employed for the channel messages: the binary value $\mathrm{s}_i \in [0, 1]$ is transformed into an *llr* as

$$l_{\mathbf{s}}(\mathrm{c}_i) = \log\left(\frac{P(\mathrm{s}_i = 0)}{P(\mathrm{s}_i = 1)}\right)$$

where $l_{\mathbf{s}}(\mathrm{c}_i) = -\infty$ if $\mathrm{s}_i = 1$ or $l_{\mathbf{s}}(\mathrm{c}_i) = \infty$ if $\mathrm{s}_i = 0$. The association of syndrome *llr*s to parity check nodes is also shown in Figure

A.1. In practical implementations of the algorithm, these syndrome likelihoods are chosen as finite approximations in order to guarantee numerical stability: $l_{\mathbf{s}}(\mathrm{c}_i) \approx -\infty$ and $l_{\mathbf{s}}(\mathrm{c}_i) \approx \infty$. Parity check nodes combine this syndrome knowledge with the information they receive from variable nodes and compute new messages which they then relay back to the variable nodes. These messages are computed according to the SPA tanh rule [47, 48, 226], and consider all incoming variable node messages except those received over the edge they are to be transmitted along, as required by the SPA message update rule [47, 48]. Essentially, this comes down to computing

$$\mu^1_{\mathrm{c}_i \to \mathrm{e}_j} = 2\mathrm{atanh} \prod_{k=1}^{\Xi-1} \tanh\left(\frac{\mu^1_{\mathrm{e}_k \to \mathrm{c}_i}}{2}\right) \tanh\left(\frac{l_{\mathbf{s}}(\mathrm{c}_i)}{2}\right), \tag{A.3}$$

where Ξ represents the number of edges of each parity check node. The product considers up to $\Xi - 1$ messages since the message $\mu^1_{\mathrm{e}_j \to \mathrm{c}_i}$ received on the edge $\mu^1_{\mathrm{c}_i \to \mathrm{e}_j}$ will be transmitted over, is not considered. Once more, the superscript 1 denotes that these messages are exchanged in the first algorithm iteration or decoding round. This procedure is portrayed over a factor graph in Figure A.2.

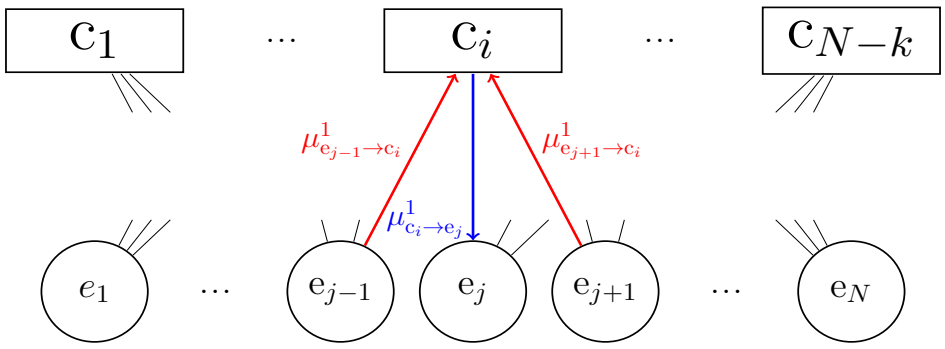


Figure A.2: *Message exchange between parity check nodes and variable nodes during the first decoding iteration.*

3. **Computation of the marginal probabilities:** This stage of the decoding process is reached once the message pair $(\mu^1_{\mathrm{e}_j \to \mathrm{c}_i}, \mu^1_{\mathrm{c}_i \to \mathrm{e}_j})$ has been exchanged over every edge of the factor graph, where $i = 1, \ldots, N-k$ and $j = 1, \ldots, N$. The exchange of these two messages

over the factor graph edges also symbolizes the conclusion of the first sum product decoding iteration. At this point, the algorithm must determine whether it has estimated the correct error sequence or if subsequent decoding iterations will be necessary. In order to produce an estimate of the error sequence, the decoder needs to obtain the marginal probabilities $P(\mathrm{e}_j|\mathbf{s})$, where we have assumed $\mathrm{e}_j \in \mathbb{F}_2$ and $j = 1, \ldots, N$. These marginal probabilities are calculated by each variable node e_j, in *llr* form and at the end of each decoding iteration[3] t, via the following computation:

$$l_{\mathrm{ap}}^{t}(\mathrm{e}_j) = l_{\mathrm{ch}}(\mathrm{e}_j) + \sum_{k=1}^{\sigma} \mu_{\mathrm{c}_k \to \mathrm{e}_j}^{t},$$

where σ denotes the degree of the variable nodes, i.e., the summation considers all incoming check node messages to each variable node.

4. **Stop/Continue criterion:** Once every variable node has obtained its corresponding *a posteriori* log-likelihood ratio $l_{\mathrm{ap}}^{t}(\mathrm{e}_j)$, the symbol-wise most likely error sequence $\hat{\mathbf{e}}^{\mathrm{SW},t}$ can be obtained. Recall that $\hat{\mathbf{e}}^{\mathrm{SW},t} = [\hat{\mathrm{e}}_1^{\mathrm{SW},t} \ldots \hat{\mathrm{e}}_N^{\mathrm{SW},t}]$, where $\hat{\mathrm{e}}_j^{\mathrm{BW}}$ is given by (A.1), ergo $\hat{\mathrm{e}}_j^{\mathrm{SW},t}$ is derived by making a hard decision[4] on $l_{\mathrm{ap}}^{t}(\mathrm{e}_j)$. Once $\hat{\mathbf{e}}^{\mathrm{SW},t}$ has been calculated, the decoder computes the product $\hat{\mathbf{s}}_t = (\hat{\mathbf{e}}^{\mathrm{SW},t})\mathbf{H}_c^{\top}$, which yields the syndrome associated to the most likely error sequence. $\hat{\mathbf{s}}_t$ is critical in dictating what is left of the decoding process: if $\hat{\mathbf{s}}_t = \mathbf{s}$, the correct solution has been found and the procedure halts. However, if $\hat{\mathbf{s}}_t \neq \mathbf{s}$ another decoding round will be executed. This procedure is repeated until $\hat{\mathbf{s}}_t = \mathbf{s}$ or a maximum number of decoding iterations T is reached.

5. **Node messaging in posterior iterations:** In order to complete this description of SPA decoding, the manner in which messages are exchanged over the factor graph after the first decoding iteration ($t = 1$) must be explained. Given that the computation of marginal probabilities and the stop/continue criterion have been discussed in terms of an arbitrary decoding iteration t, the same must be done for the messaging between check nodes and variable nodes, and vice-versa. If we assume that in the previous decoding round $t - 1$ an

[3] Subsequent algorithm steps operate identically for every decoding iteration, hence it is logical to describe them for an arbitrary iteration.

[4] This is as simple as estimating $\hat{\mathrm{e}}_j^{\mathrm{SW}} = 0$ if $l_{\mathrm{ap}}^{t}(\mathrm{e}_j) \geq 0$ and $\hat{\mathrm{e}}_j^{\mathrm{SW}} = 1$ otherwise.

incorrect solution was obtained, then during the subsequent iteration t, each variable node e_j will compute and transmit the following message over each of its edges:

$$\mu^{t}_{\mathrm{e}_j \to \mathrm{c}_i} = l_{\mathrm{ch}}(\mathrm{e}_i) + \sum_{k=1}^{\sigma-1} \mu^{t-1}_{\mathrm{c}_k \to \mathrm{e}_j},$$

where σ denotes the degree of the variable nodes, and so messages are computed according to the sum-product update rule: disregarding the message received in the previous iteration through the edge over which transmission of $\mu^{t}_{\mathrm{e}_j \to \mathrm{c}_i}$ is to occur. This is shown in Figure A.3.

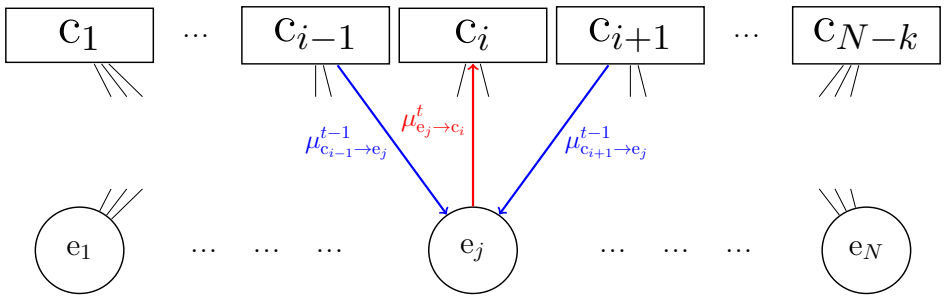


Figure A.3: *Message exchange between variable nodes and parity check nodes during decoding iteration t.*

In terms of messages transmitted from parity check nodes to variable nodes during iteration t, the messages are computed according to the expression shown in (A.3). This exchange of messages is shown in Figure A.2. Notice that said figure and expression (A.3) are a representation of the first decoding iteration: $t = 1$. Hence for subsequent iterations messages should actually be denoted by the superscript t.

APPENDIX B

Improved Decoding Strategies for QLDPC codes

In what follows we provide a detailed summary of the most relevant modified/improved QLDPC decoding strategies that have appeared in the literature.

B.1 CORRELATION EXPLOITING DECODER

The correlation exploiting decoder of [22] implements the SPA modifications originally suggested in [18] and represents one of the earliest attempts of an improved decoding strategy for CSS QLDPC codes. Although they are discussed in Chapter 6, in case readers have come directly from Chapter 5 we will also present CSS codes here. CSS codes are a subclass of stabilizer codes that are built from existing classical codes in such a manner that the fulfilment of the symplectic criterion is guaranteed. The QPCM of a generic CSS code is given by

$$\mathbf{H}_Q = (\mathrm{H}_x|\mathrm{H}_z) = \begin{pmatrix} \mathrm{H}_x^{'} & 0 \\ 0 & \mathrm{H}_z^{'} \end{pmatrix}, \tag{B.1}$$

where $\mathrm{H}_x = \begin{pmatrix} \mathrm{H}_x^{'} \\ 0 \end{pmatrix}$ and $\mathrm{H}_z = \begin{pmatrix} 0 \\ \mathrm{H}_z^{'} \end{pmatrix}$. In this construction, $\mathrm{H}_x^{'}$ and $\mathrm{H}_z^{'}$ are the parity check matrices of two classical LDPC codes C_1 and C_2,

respectively, where each matrix is used to correct either bit-flips or phase-flips. The classical codes are chosen so that $C_2^{\perp} \subseteq C_1$, where $C_2^{\perp}$ is the dual of the classical LDPC code C_2. This design constraint, generally referred to as the *CSS condition*, reduces the expression of the symplectic criterion shown in (4.24) to $\mathrm{H}_z^{'}\mathrm{H}_x^{'T} = 0$. Due to the particular structure of a CSS QPCM (B.1), CSS codes can and are almost always decoded by means of two separate binary BP decoders. This is made possible because $\mathbf{w} = \mathbf{H}_Q \odot \mathbf{e} = \mathrm{H}_z\mathbf{e}_x^{\top} \oplus \mathrm{H}_x\mathbf{e}_z^{\top} = [\mathrm{H}_z'\mathbf{e}_x^{\top}\ \mathrm{H}_x'\mathbf{e}_z^{\top}] = [\mathbf{w}_x\ \mathbf{w}_z]$, where $\mathbf{H}_C$ is given in (B.1), $\mathbf{w}_x = \mathrm{H}_z'\mathbf{e}_x^{\top}$ and $\mathbf{w}_z = \mathrm{H}_x'\mathbf{e}_z^{\top}$, respectively. Thus, the syndrome that is obtained when using a CSS code is of the form $\mathbf{w} = [\mathbf{w}_x\ \mathbf{w}_z]$, i.e, half of it contains information strictly about bit flips and the other half only provides knowledge regarding phase flips. In this way, two separate decoders can be used simultaneously to produce the error estimates $\hat{\mathbf{e}}_x$ and $\hat{\mathbf{e}}_z$, and so obtain $\hat{\mathbf{w}}_x = \mathrm{H}_z'\hat{\mathbf{e}}_x^{\top}$ and $\hat{\mathbf{w}}_z = \mathrm{H}_x'\hat{\mathbf{e}}_z^{\top}$, where these are estimates of $\mathbf{w}_x$ and $\mathbf{w}_z$, respectively.

Despite being convenient, a decoding stratagem based on two independent binary BP decoders will ignore the correlations that exist between X and Z errors over the depolarizing channel due to its separate processing of bit and phase flips. The correlation exploiting decoder circumvents this issue by iteratively exchanging messages between the X and Z operator decoders. After each iteration, the a priori probability of each X or Z variable node[1] q is updated based on the decoding outcome of the opposite decoder, i.e, the variable nodes associated to the X operators, which we will denote as $\mathbf{e}_x$, are updated based on the results obtained by the Z operator decoder and viceversa. More explicitly, during each decoding iteration t, the X decoder transmits the message $p^t(\mathbf{E}_{z,q} = 1)$ to the Z decoder and the Z decoder relays the message $p^t(\mathbf{E}_{x,q} = 1)$ to the X decoder. These messages are given by

$$\begin{aligned} &p^t(\mathbf{E}_{z,q} = 1) = p^t(\mathbf{E}_{x,q} = 1)p(\mathbf{E}_{z,q} = 1|\mathbf{E}_{x,q} = 1) \\ &+ (1 - p^t(\mathbf{E}_{x,q} = 1))p_q(\mathbf{E}_{z,q} = 1|\mathbf{E}_{x,q} = 0), \\ &p^t(\mathbf{E}_{x,q} = 1) = p_q^t(\mathbf{E}_{z,q} = 1)p(\mathbf{E}_{x,q} = 1|\mathbf{E}_{z,q} = 1) \\ &+ (1 - p^t(\mathbf{E}_{z,q} = 1))p(\mathbf{E}_{x,q} = 1|\mathbf{E}_{z,q} = 0), \end{aligned} \tag{B.2}$$

where $p^t(\mathbf{E}_{x,q} = 1)$ and $p^t(\mathbf{E}_{z,q} = 1)$ denote the probability of an X or Z error taking place on the q-th variable node of each corresponding decoder

[1] In this case, variable nodes represent qubits.

Table B.1: *Joint Distribution of X and Z operators over the depolarizing channel. p is the depolarizing probability of the channel.*

$P(\mathbf{E}_x, \mathbf{E}_z)$	$\mathbf{E}_x = 0$	$\mathbf{E}_x = 1$
$\mathbf{E}_z = 0$	$1 - p$	$\frac{p}{3}$
$\mathbf{E}_z = 1$	$\frac{p}{3}$	$\frac{p}{3}$

during that same iteration t, and the conditional probabilities are obtained from the joint distribution of X and Z operators over the depolarizing channel shown in table B.1. The messages $p^t(\mathbf{E}_{z,q} = 1)$ and $p^t(\mathbf{E}_{x,q} = 1)$ are used as the a priori probabilities for the q-th Z and X variable nodes in the $(t+1)$-th decoding round. In [22], this correlation exploiting decoder is shown to yield small performance improvements, foreshadowing the potential that degeneracy-exploiting decoding strategies have to improve the performance of QLDPC codes.

B.2 FREEZING DECODER

The "freezing" technique is the first and most simple method proposed in [55]. The procedure works by selecting a random frustrated check node from the original factor graph and tampering with the prior probabilities of one of the variable nodes connected to it. Frustrated check nodes are those whose associated real syndrome component w_c does not match with the same estimated syndrome component $\hat{\mathrm{w}}_c$, where $\mathbf{w} = [\mathrm{w}_1, \ldots, \mathrm{w}_{N-k}]$, $\hat{\mathbf{w}}$ is obtained from the symplectic product $\mathbf{H}_Q \odot \hat{\mathbf{e}}$, $\mathbf{H}_Q$ is the QPCM of the code, and $\hat{\mathbf{e}}$ denotes the symplectic representation of the estimate of the error pattern produced by a standard BP decoder.

Once a frustrated check node has been chosen, the prior probability $p(\mathbf{E}_q)$ or channel llr $l_{\mathrm{ch}}(\mathbf{E}_q)$ of one of the variable nodes $\mathbf{e}_q$ connected to the aforementioned check node is "frozen" to a value δ_q. The decoder is rerun based on this new prior probability for a fixed number of iterations. If the correct syndrome estimate is found during this second decoding phase then the technique has been successful. If not, then another of the variable nodes connected to the frustrated check node is "frozen", the previous check node is restored and decoding is reattempted. If one of these combinations results in a non-frustrated check node but not the appropriate syndrome,

another frustrated check node is selected, a variable node connected to it is frozen at random, and the decoding process is re-run. Given the myriads of frozen and non-frozen node combinations that are possible with this technique, if the correct syndrome estimate is not found in the first few decoding reattempts, the increase in decoding complexity can become quite substantial.

B.3 RANDOM PERTURBATION DECODER

This technique is similar to the "freezing" decoder in that it also works with the prior probabilities of qubits related to frustrated check nodes. It consists in identifying all of the frustrated check nodes of the factor graph and applying the following perturbations (up to normalization) to the prior probabilities of each variable node q connected to these check nodes:

$$\begin{aligned} &p(\mathbf{E}_q = \mathrm{I}) \to p(\mathbf{E}_q = I) \\ &p(\mathbf{E}_q = \mathrm{X}) \to (1+\delta_x)p(\mathbf{E}_q = \mathrm{X}) \\ &p(\mathbf{E}_q = \mathrm{Y}) \to (1+\delta_y)p(\mathbf{E}_q = \mathrm{Y}) \\ &p(\mathbf{E}_q = I) \to (1+\delta_z)p(\mathbf{E}_q = \mathrm{Z}), \end{aligned} \tag{B.3}$$

where δ_x, δ_y, and δ_z are random variables uniformly distributed over the range $[0, \delta]$ for a fixed δ. Notice that this technique is strictly designed for a GF(4) decoder. A similar strategy for binary decoders, known as the adjusted decoder, is shown later on. The primary goal of this method is to resolve symmetric degeneracy errors (see section 4.3.2) by perturbing the prior probabilities used by the qubit-wise BP decoder. Symmetric degeneracy errors occur because degenerate errors of equal weight are completely symmetric under qubit-wise BP decoding, and so the only way to resolve them is by breaking this symmetry. This requires the perturbations to be random, since an equal increment in the probabilities would cause the issue to persist. In [55], a random perturbation decoder is shown to be successful in solving instances in which symmetric degeneracy errors have taken place.

B.4 COLLISION DECODER

Decoding based on collisions is the last technique introduced in [55]. This strategy is used in conjunction with either of the previously discussed

methodologies (freezing or random perturbation) to have a more structured approach to fixing symmetric degeneracy errors. Once the traditional BP decoder has failed, the collision decoder begins by identifying a pair of "colliding" check nodes, which are two unsatisfied check nodes that share some variable nodes. Assuming that colliding pairs of check nodes occur as a result of errors on their shared variable nodes, the decoder then chooses to either apply random perturbations to those shared nodes or to freeze one of them.

B.5 ENHANCED FEEDBACK DECODER

The authors of [55] mention that despite the performance improvements provided by their proposed decoders, all of the errors that arise in their simulations are still end-to-end different syndrome errors (no degenerate or identical syndrome errors occur). This means that all of the errors that happen when using these modified decoding strategies can be attributed to the decoders themselves, which implies that further improvements can be made to these modified decoding strategies and a higher number of those end-to-end errors with different syndromes might possibly be corrected. This is done in [149], where the authors, following the observations made in [55], present an enhanced feedback BP iterative decoding algorithm that provides useful information to the BP decoder based on exploiting not only the syndrome but also the stabilizer elements themselves.

As do most of the other modified decoding approaches, this method uses a conventional BP decoder as its default decoding tool until an end-to-end different syndrome error is found. Once this happens, a frustrated check node c_j and a variable node q that share an edge on the factor graph are chosen at random. If $\mathrm{w}_j = 1$ and $\hat{\mathrm{w}}_j = 0$, where w_j represents the j-th component of the error syndrome $\mathbf{w} = [\mathrm{w}_1, \ldots, \mathrm{w}_{N-k}]$ and $\hat{\mathrm{w}}_j$ represents the j-th component of the estimated error syndrome $\hat{\mathbf{w}} = [\hat{\mathrm{w}}_1, \ldots, \hat{\mathrm{w}}_{N-k}]$, it is clear that the real error $\mathbf{E}$ anticommutes with the j-th stabilizer generator $\mathbf{S}_j$ while the estimated error $\hat{\mathbf{E}}$ commutes with $\mathbf{S}_j$. The enhanced feedback decoder uses this information to modify the prior probabilities of the variable node q (as it connects to the check node c_j associated to the syndrome component w_j) in order to make an anticommuting error more likely than the trivial error that the decoder generally leans towards (the decoder is biased towards estimating the identity operator). Thus, in the case that $\mathrm{w}_j = 1$ and $\hat{\mathrm{w}}_j = 0$, the prior probabilities are changed to

$$p(\mathbf{E}_q = \gamma) \rightarrow \begin{cases} \frac{p}{2} & \text{if } \gamma = I \text{ or } \gamma = \mathbf{S}_j^q, \\ 1 - \frac{p}{2} & \text{otherwise,} \end{cases} \tag{B.4}$$

where $\mathbf{S}_j^q$ is the q-th component of the j-th stabilizer generator and p is the depolarizing probability of the channel. If the inverse scenario is encountered, i.e, $\mathrm{w}_j = 0$ and $\hat{\mathrm{w}}_j = 1$, then the prior probabilities are changed to

$$p(\mathbf{E}_q = \gamma) \rightarrow \begin{cases} 1 - \frac{p}{2} & \text{if } \gamma = I \text{ or } \gamma = \mathbf{S}_j^q, \\ \frac{p}{2} & \text{otherwise.} \end{cases} \tag{B.5}$$

For instance, if $\mathbf{S}_j^q = X$ when $\mathrm{w}_j = 1$ and $\hat{\mathrm{w}}_j = 0$, then we would have the adjusted prior probabilities $p(\mathbf{E}_q = I) = P(\mathbf{E}_q = X) = \frac{p}{2}$ and $p(\mathbf{E}_q = Z) = P(\mathbf{E}_q = Y) = 1 - \frac{p}{2}$. Whereas if $\mathrm{w}_j = 0$ and $\hat{\mathrm{w}}_j = 1$ we would have $p(\mathbf{E}_q = I) = P(\mathbf{E}_q = X) = 1 - \frac{p}{2}$ and $p(\mathbf{E}_q = Z) = P(\mathbf{E}_q = Y) = \frac{p}{2}$.

After the prior probabilities of the variable node are adjusted, decoding is reattempted for a fixed number of iterations. If the algorithm halts during this process, i.e, $\mathbf{w} = \hat{\mathbf{w}}$, then the correct syndrome has been found. If the check node c_j is still frustrated after this process then the prior probabilities of variable node i are restored and the prior probabilities of a different variable node i' connected to c_j are modified following equations (B.4) and (B.5). This procedure runs until c_j is no longer frustrated or until decoding based on modified prior probabilities has been attempted with all the variable nodes connected to c_j. If c_j is no longer frustrated and $\mathbf{w} = \hat{\mathbf{w}}$ then decoding is complete. However, if c_j is no longer frustrated but $\mathbf{w} \neq \hat{\mathbf{w}}$, or all the qubits that connect to c_j have been exhausted, then a different frustrated check node c'_j is selected and the process begins anew. It is obvious that the complexity increase resulting from this technique will be heavily dependent on the efficacy of the prior probability modifications, i.e, how many frustrated check nodes must it tamper with prior to the correct syndrome being found. In [149], the enhanced feedback decoder is shown to significantly outperform the random perturbation decoder of [55] in terms of both QBER performance and decoding efficiency. In fact, when compared to a standard BP decoder, the work of [149] shows that

the enhanced feedback technique is approximately 10 times better[2] while only requiring a 30% increase in the number of decoding iterations.

B.6 SUPERNODE DECODER

In [20], a modified GF(4) decoder for dual-containing CSS codes, also referred to as homogeneous CSS codes, is proposed. A CSS code is said to be dual-containing if $\mathrm{H}_z^{'} = \mathrm{H}_x^{'}$. These codes are useful because $\mathrm{H}_z^{'}\mathrm{H}_x^{'T} = \mathrm{H}_z^{'}\mathrm{H}_z^{'T} = 0$ is always fulfilled. The main drawback of these dual-containing CSS codes is that their performance under SPA-based decoding is hindered by their large amount of length-4 cycles. To make matters worse, these cycles become even more prevalent when the error correction scheme is quaternary. Quaternary quantum codes are built based on the Pauli-to-GF(4) isomorphism, which similarly to the mapping shown in (4.6), serves to map the single qubit Pauli operators to equivalent 4-ary symbols as

$$I \to 0,\ Z \to 1,\ X \to \omega,\ Y \to \bar{\omega}. \tag{B.6}$$

Using this mapping, a quaternary $m \times N$ CSS QPCM $\mathbf{H}_Q$ can be constructed as:

$$\mathbf{H}_Q = \begin{pmatrix} \mathrm{H}_x' \\ \omega\mathrm{H}_z' \end{pmatrix}, \tag{B.7}$$

where $\mathrm{H}_z^{'}$, and $\mathrm{H}_x^{'}$ are the same as in (B.1). Herein lies the reason why the length-4 cycle issue is further aggravated for homogeneous quaternary CSS codes. Since $\mathrm{H}_z^{'} = \mathrm{H}_x^{'}$, the same two classical PCMs are used to build $\mathbf{H}_Q$, which forces the i-th and $(i + \frac{m}{2})$-th rows to completely overlap and leads to an increase in the number of the aforementioned cycles.

Fortunately, the supernode decoder is capable of eliminating these specific cycles (those that are generated because of the homogeneous CSS structure of $\mathbf{H}_Q$). It achieves this by means of a modified Tanner graph where the i-th and $(i+\frac{m}{2})$-th check nodes become the i-th supernode, which automatically halves the number of length-4 cycles of the original factor graph. Naturally, such a change requires that the message exchange between the resulting supernodes and the original variable nodes be updated so that

[2]It is also worth noting that the authors of [149] assess their decoding performance using the physical error rate; they do not account for the presence of degeneracy. This means that the enhanced feedback decoder may actually exhibit better performance than is reflected in [149].

both of the original checks, c_i and $\mathrm{c}_{i+\frac{m}{2}}$, are still satisfied. Since the i-th and $(i+\frac{m}{2})$-th rows of $\mathbf{H}_Q$ fulfill $\mathrm{H}_i = \omega \mathrm{H}_{i+\frac{m}{2}}$, the i-th and $(i+\frac{m}{2})$-th syndrome components will be identically related, $\mathrm{w}_i = \omega \mathrm{w}_{i+\frac{m}{2}}$. Thus, the computation of the check-to-variable node messages changes only in that an updated syndrome $\bar{\mathrm{w}}_i = \mathrm{w}_i + \omega \mathrm{w}_{i+\frac{m}{2}} \in \mathrm{GF}(4)$ is used, where $i = 1, \ldots, \frac{m}{2}$. Essentially, the factor graph is reduced in size without the computation of the check node messages becoming more complex, which implies that the supernode decoder also represents an improvement in terms of decoding complexity when compared to the previously discussed heuristic methods.

In [20], this supernode decoder is shown to exhibit superior WER performance and lower decoding complexity when compared to existing state-of-the-art decoding techniques. Additionally, given that some short cycles are still present in the supernode factor graph (those that arise because of the symplectic criterion), the authors of said work also apply the Uniformly-Reweighted Belief Propagation (URW-BP) strategy of [219, 220] to their supernode decoder in order to alleviate the impact of the remaining short cycles. The authors also mention that the supernode decoder may be seamlessly amalgamated with other heuristic methods to improve performance even further. This is done in [135], where a set of novel improved decoding strategies are proposed. The reader is referred to [20, 118, 220, 221] for a rigorous discussion on the specifics of quaternary decoding and its increased nuance.

B.7 ADJUSTED DECODER

The adjusted decoder is the first improved decoding technique that is proposed in [135]. A similar decoder is derived and shown to improve the performance of surface codes in [222]. Reminiscent of the correlation exploiting decoder, the goal of the adjusted decoder is to reintroduce the correlations between X and Z errors that are present in the depolarizing channel but are ignored when decoding a CSS code with a generic binary CSS decoder.

The adjusted decoder tackles this issue by adding the correlations neglected by the separate binary decoders via "adjusting" the a priori channel probabilities of specific variable nodes of the factor graph. As with most of the previously discussed heuristic methods, the procedure begins by attempting to decode with a generic binary CSS decoder comprised of two

separate decoders, one to decode bit-flips and the other to decode phase-flips. If decoding is successful or if both estimates $\hat{\mathbf{w}}_x \neq \mathbf{w}_x$ and $\hat{\mathbf{w}}_z \neq \mathbf{w}_z$, then the process halts. However, if one of the estimates is correct, then decoding is reattempted for the incorrect component based on a set of adjusted probabilities. For instance, if $\hat{\mathbf{w}}_x = \mathbf{w}_x$ but $\hat{\mathbf{w}}_z \neq \mathbf{w}_z$, then the a priori probabilities of the Z operator variable nodes are modified to:

$$p(\mathbf{e}_{z,q} = 1) \rightarrow \begin{cases} \frac{p_y}{p_x+p_y} & \text{if } \hat{\mathbf{E}}_{x,q} = 1, \\ \frac{p_z}{1-(p_x+p_y)} & \text{otherwise,} \end{cases} \tag{B.8}$$

where the subscript q denotes the q-th variable node of the Z decoder and $\mathbf{e}$ denotes the symplectic representation of the error operator in question. If instead $\hat{\mathbf{w}}_z = \mathbf{w}_z$ but $\hat{\mathbf{w}}_x \neq \mathbf{w}_x$, then the adjustment is the following:

$$p(\mathbf{e}_{x,q} = 1) \rightarrow \begin{cases} \frac{p_y}{p_y+p_z} & \text{if } \hat{\mathbf{E}}_{z,q} = 1, \\ \frac{p_x}{1-(p_y+p_z)} & \text{otherwise,} \end{cases} \tag{B.9}$$

where the subscript q denotes the q-th variable node of the X decoder.

B.8 AUGMENTED DECODER

The second modified decoding technique proposed in [135] is the augmented decoder. It is based on the decoder proposed in [228] for classical binary LDPC codes and serves to improve the performance of both binary and quaternary BP decoders. As with most of the previously discussed methodologies, decoding is initially attempted with a standard BP decoder. If this is unsuccessful, decoding is reattempted with an "augmented" PCM

$$\mathbf{H}_A = \begin{pmatrix} \mathbf{H}_Q \\ \mathbf{H}_\delta \end{pmatrix},$$

where $\mathbf{H}_\delta$ denotes a subset of rows chosen at random from the original QPCM $\mathbf{H}_Q$. The size of this subset is determined by the parameter δ, which is known as the augmentation density. Decoding based on $\mathbf{H}_A$ also requires that the syndrome be augmented accordingly, $\mathbf{w}_A = [\mathbf{w}\ \mathbf{w}_\delta]$, where $\mathbf{w}$ is the measured syndrome and $\mathbf{w}_\delta$ represents the syndrome values associated

to the rows of $\mathbf{H}_Q$ that comprise $\mathbf{H}_\delta$. If the initial decoding reattempt based on $\mathbf{H}_A$ is not successful, the process is repeated until either $\hat{\mathbf{w}} = \mathbf{w}$ or a predefined maximum number of attempts is reached. In the particular case of binary CSS decoders, both the X decoder PCM and the Z decoder PCM, as well as the syndromes $\mathbf{w}_x$ and $\mathbf{w}_z$, have to be augmented.

Since decoding directly over the augmented matrix $\mathbf{H}_A$ conduces to an increase in complexity (larger matrices and syndromes are required), an equivalent decoding alternative capable of mitigating the impact of the augmentation strategy on the complexity of the scheme is also proposed in [135]. Instead of using the augmented matrix, this method operates by modifying the marginal probabilities of the SPA based on a binary function whose entries represent if a check node from the factor graph has been duplicated when performing the augmentation. In terms of performance, augmented GF(4) decoders and augmented supernode decoders have been shown to outperform random perturbation and enhanced feedback decoders when used to decode dual-containing CSS codes [135]. When applied to non-dual-containing CSS codes and non-CSS codes, the augmented decoder was shown to perform similarly to random perturbation and enhanced feedback decoders.

B.9 COMBINED DECODER

The last technique proposed in [135] is a CSS decoder that combines both the adjusted and augmented decoding methods. Initially, a standard binary CSS decoder is executed. If both $\hat{\mathbf{w}}_x$ and $\hat{\mathbf{w}}_z$ do not match with the measured syndromes, then decoding is reattempted for the X operators using the augmentation technique. If this is unsuccessful after a fixed number of attempts, the augmented decoder is used for the Z operators. If both of the estimated syndromes still do not match, then the procedure is halted. However, if we obtain $\hat{\mathbf{w}}_x = \mathbf{w}_x$ or $\hat{\mathbf{w}}_z = \mathbf{w}_z$, either from the initial standard binary decoding or due to one of the augmented modifications, then decoding is reattempted for the remaining failed component by means of adjusted a priori probabilities. If this is unsuccessful, the last resort is to reattempt decoding for the failed component using the augmentation technique and maintaining the adjusted probabilities. In [135], the combined decoder is shown to outperform random perturbation and enhanced feedback decoding techniques when applied to dual-containing CSS codes. When used to decode non-dual containing CSS codes, the com-

bined decoder outperforms a regular GF(4) decoder but performs slightly worse than quaternary modified decoders (enhanced feedback, augmented and random perturbation).

B.10 ORDERED STATISTICS DECODER

The Ordered Statistics Decoding (OSD) technique of [229] is a well known classical decoding strategy that improves performance at the expense of the decoding complexity. In [151], the OSD algorithm is applied to the quantum domain as a post-processing methodology to improve performance whenever the traditional BP decoder fails. The algorithm uses the soft-decisions made by the traditional BP decoder to re-arrange the qubits in descending order of their total probability of error. This serves to sort the qubits in terms of their reliability, which is then used to make hard decisions on the qubits for which the traditional BP decoder has the most certainty. Next, assuming that the decoder makes correct hard decisions on $N - \mu$ qubits, the OSD-decoder flips the μ most unreliable qubits of the traditional BP error estimate in order to find an error estimate that produces a matching syndrome. If multiple correction operators are found, the error sequence of minimum weight is chosen as the correct estimate of the error pattern. In some cases, the initial number of unreliable qubits that is used by the OSD algorithm is insufficient to produce an acceptable estimate of the error. If so, the qubits are reprocessed by flipping an increased number of unreliable qubits until an appropriate estimate is found or until a predefined number of processing rounds is reached. If one processing round is sufficient to find a matching syndrome the algorithm is known as order-0 or OSD-0. If subsequent processing rounds are required, the algorithm is referred to as OSD-i or order i OSD, where i denotes the number of processing rounds.

Given its particular structure and provided that sufficient decoding time is permitted, the OSD decoder will always recover an error pattern that maps to the correct syndrome. This stands in stark contrast to the generic SPA decoder and all of the aforementioned improved decoding techniques, which may sometimes yield an incorrect syndrome regardless of the allowed decoding time. In [151, 152] the OSD decoder is shown to vastly improve the performance of a generic SPA decoder, regardless of the type of employed QLDPC code. In fact, the OSD decoding algorithm is compatible with non-CSS decoding, which if applied to existing non-CSS decoders, would serve to

further improve their performance [23, 63]. Also in [151], the order 0 version of the algorithm is shown to surpass the enhanced feedback, augmented, and random perturbation decoders.

B.11 REFINED BELIEF PROPAGATION DECODING

Recently, in [153], methods to improve the performance and simplify the complexity of a GF(4) decoder for QLDPC codes have been proposed. Despite the fact that this decoder is not specifically designed to correct end-to-end errors with different syndromes, it is relevant to our discussion. First off, it represents a leap forward for quantum quaternary decoding, which in terms of performance is a better decoding strategy to handle end-to-end errors with different syndromes than the binary BP decoder. The authors of [153] show that their refined BP quaternary decoder has the same decoding complexity as a standard binary BP decoder. This symbolizes a major reduction in the complexity of quaternary decoding, since the decoding algorithm of a generic GF(4) decoder runs 16 times slower than the binary equivalent when decoding a quantum stabilizer code. Another reason that makes the decoding strategy of [153] relevant is that this modified GF(4) decoder is compatible with all of the previously discussed improved decoding methods, which implies that performance of QLDPC codes could become orders of magnitude better while only paying a small price in the form of a more complex decoder.

The reduction in the decoding complexity of the refined BP decoder is achieved by passing single-valued messages over the factor graph instead of the multivalued messages that are typically exchanged in quaternary decoding. The use of single-valued messages in a GF(4) decoder is made possible by the insight that check node messages are more indicative of the commutation status of an error component and the corresponding stabilizer generator than the actual type of Pauli operator acting on said component. This stems from the fact that a quantum syndrome, whose components are directly related to the check nodes of a QLDPC factor graph, is a binary vector that represents the commutation status of each error component with the stabilizer generators.

Aside from the use of single-valued messages, two additional useful strategies to improve the performance of quantum BP decoders while keeping their complexity low are detailed in [153]. The first one involves modi-

fying the message passing schedule of the decoder from a parallel message passing schedule to a serial one. The most common implementation of the SPA or BP algorithm is based on a parallel schedule, where messages of the same kind are exchanged simultaneously over the factor graph during each iteration, i.e, all the variable node messages are transmitted to the check nodes at the same time instant t_1 and all the check node messages are transmitted to the variable nodes at the same time instant t_2. A less common but possibly better messaging schedule is the serial implementation of the BP algorithm. In [230, 231], it is shown to improve the convergence behavior when the underlying Tanner graph has many short cycles at no increase in decoding complexity. If messages are updated in this manner, the process operates in a top-down sequential fashion, where the first variable node receives messages from its neighbouring check nodes, to who it then replies with its own messages, who then move on to the next variable node, and so on.

The second strategy involves adjusting the magnitude of the messages exchanged over the factor graph, since they can easily become overestimated [232, 233, 234]. Such an adjustment can be performed via message normalization or offset, which can be applied seamlessly to the refined BP decoder given the complexity reduction that is attained by migrating to single-valued message exchanges.

The results shown in [153] show that performance can be significantly improved by using a serial BP algorithm and normalization techniques. The authors of this work mention that further improvements may be possible by improved exploitation of quantum degeneracy, either by stacking their decoder with any of the heuristic methods we have explained if complexity increments are affordable[3], or by devising new techniques. This is discussed at length in [154].

[3]Decoding efficiency is primordial in the quantum paradigm, since the coherence of quantum states decays rapidly.

APPENDIX C

Monte Carlo Simulations

Monte Carlo simulations, also known as the Monte Carlo Method or multiple probability simulations, are a mathematical technique that can be used to model the probability of different outcomes in a process that cannot easily be predicted due to the intervention of random variables. It is a technique used to understand the impact of risk and uncertainty in prediction and forecasting models. A simple example of a Monte Carlo Simulation is to consider calculating the probability of rolling two standard dice. There are 36 combinations of dice rolls. Based on this, you can manually compute the probability of a particular outcome. Using a Monte Carlo Simulation, you can simulate rolling the dice 10,000 times (or more) to achieve more accurate predictions.

In this dissertation we employ the Monte Carlo Method to assess the performance of various families of quantum codes. The method provides us with guidelines that ensure that the simulations we conduct are accurate, which ultimately allows us to draw valid conclusions from the results these simulations provide. Although they vary depending on the quantum channel model under consideration, all our simulations operate based on the following principle:

1. First, an N-qubit operator $\mathbf{A} \in \overline{\mathcal{G}}_N$ is generated according to the probability distribution of the particular quantum channel that is being considered. This probability distribution will be different for each

of the quantum channels (i.i.d. X/Z channel, depolarizing channel, asymmetric channel) discussed in this dissertation.

2. Next, the syndrome $\mathbf{w}$ associated to the operator $\mathbf{A}$ is computed and fed to the decoder so that the decoding process can begin.

3. Once the decoding process has finished, the decoding estimate $\hat{\mathbf{A}}$ and the real operator $\mathbf{A}$ are compared.

4. After this comparison, a new simulation round begins by going back to step 1 and generating another N-qubit operator.

In terms of the comparison that is performed in step 3 different performance assessment metrics or figures of merit can be obtained. Throughout this dissertation we employ the WER and the QBER as the figures of merit. The WER represents the probability that at least one qubit of the estimated operator is incorrect, i.e, it compares the entire operator instead of its constituent single-qubit operators. Thus, the WER will consider that the scheme has been successful whenever $\hat{\mathbf{A}} \star \mathbf{A} \in \overline{\mathcal{Z}}(\overline{\mathcal{S}})$ and that an error has occurred whenever this does not hold[1]. On the other hand, the QBER compares the individual single-qubit operators that make up $\mathbf{A}$ and $\hat{\mathbf{A}}$, so it represents the fraction of qubits that experience an error. This means that it is computed by checking if $\mathrm{A}_i = \hat{\mathrm{A}}_i, \forall i = 1, \ldots, N$, where a qubit error occurs whenever the equality does not hold.

To ensure that the WER and QBER results of our simulations are accurate, we invoke the rules of Monte Carlo simulations. This requires that we follow the guideline shown below to select the number of necessary simulation rounds[2] $\mathrm{N}_{\mathrm{rounds}}$ [162]:

$$\mathrm{N}_{\mathrm{rounds}} = \frac{100}{\mathrm{WER}}. \tag{C.1}$$

If we assume that the observed error events are independent, the above rule of thumb yields the following 95 % confidence interval:

[1] Recall that comparison up to a stabilizer element computes the logical error rate, which is the appropriate way of assessing the performance of degenerate quantum codes.

[2] Simulation rounds are also sometimes referred to as blocks due to the similarity with classical communications in which each simulation round represents the transmission of a block of information bits.

$$P(0.8\tilde{\text{WER}} \leq \text{WER} \leq 1.25\tilde{\text{WER}}) = 0.95,$$

where $\tilde{\text{WER}}$ represents the empirically estimated value of the WER.

For example, in order to have high confidence (with a 95% confidence level) regarding the performance of an arbitrary quantum code at a WER of 10^{-3}, this rule tells us that we would need to conduct $\text{N}_{\text{rounds}} = 100000$ simulation rounds. In summary, conducting our simulations based on the principle of (C.1) ensures that the corresponding results will be statistically representative and that they will accurately portray the performance of the quantum codes that are being considered.

References

[1] R. P. Feynman, "Simulating physics with computers," *Int. J. Theor. Phys.*, vol. 21, pp. 467-488, 1982. doi: 10.1007/BF02650179.

[2] S. Imre and F. Balazs, "Quantum Computing and Communications: An Engineering Approach," *John Wiley & Sons*, 2005. ISBN:978-0-470-86902-4.

[3] 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. doi: https://doi.org/10.1098/rspa.1992.0167.

[4] P. W. Shor, "Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer," *SIAM Journal on Computing*, vol. 26, no. 5, pp. 1484-1509, 1997. doi: 10.1137/S0097539795293172.

[5] L. K. Grover, "A Fast Quantum Mechanical Algorithm for Database Search," *Annual ACM Symp. on Theory of Comp.*, pp. 212-219, 1996. arXiv:quant-ph/9605043.

[6] L. K. Grover, "From Schrodingers equation to the quantum search algorithm," *Pramana - J. Phys.*, vol. 56, pp. 333-348, 2001. doi: 10.1007/s12043-001-0128-3.

[7] M. Fitzi, N. Gisin, and U. Maurer, "Quantum Solution to the Byzantine Agreement Problem," *Phys. Rev. Lett.*, vol. 87, no. 21, pp. 217901, 2001. doi: 10.1103/PhysRevLett.87.217901.

[8] 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. doi: 10.1109/ACCESS.2014.2327596.

[9] 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. doi: 10.1109/ACCESS.2015.2478793.

[10] P. Botsinis, D. Alanis, Z. Babar, S. X. Ng, and L. Hanzo, "Non-coherent quantum multiple symbol differential detection for wireless systems," *IEEE Access*, vol. 3, pp. 569-598, 2015. doi: 10.1109/ACCESS.2015.2432015.

[11] C. H. Bennett, and G. Brassard, "Quantum cryptography: Public key distribution and coin tossing ," *Theoretical Computer Science*, vol. 560, pp. 7-11, 2014. doi: 10.1016/j.tcs.2014.05.025.

[12] A. Ekert, "Quantum cryptography based on Bells theorem," *Phys. Rev. Lett.*, vol. 67, no. 6, pp. 661-663, 1991. doi: 10.1103/PhysRevLett.67.661.

[13] P. W. Shor, "Scheme for reducing decoherence in quantum computer memory," *Phys. Rev. A*, vol. 52, no. 4, pp. R2493-R2496, 1995. doi: 10.1103/PhysRevA.52.R2493.

[14] M. Schlosshauer, "Decoherence, the measurement problem, and interpretations of quantum mechanics," *Rev. Mod. Phys.*, vol. 76, pp. 1267, 2005. doi: 10.1103/RevModPhys.76.1267.

[15] M. Schlosshauer, "Quantum decoherence," *Physics Reports*, vol. 831, pp. 1-57, 2019. doi: 10.1016/j.physrep.2019.10.001.

[16] D. Gottesman, "Stabilizer codes and quantum error correction," *PhD dissertation*, California Inst. Tech., Pasadena, CA, USA, 1997. arXiv:quant-ph/9705052.

[17] A. M. Steane, "Quantum Reed Muller codes," *IEEE Trans. Inf. Theory*, vol. 45, no. 4, pp. 1701-1703, 1999. doi: 10.1109/18.771249.

[18] D. J. C. MacKay, G. Mitchinson, and P. L. McFadden, "Sparse-graph codes for quantum error correction," *IEEE Trans. Inf. Theory*, vol. 50, no. 10, pp. 2315-2330, 2004. doi: 10.1109/TIT.2004.834737.

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

[20] Z. Babar, P. Botsinis, D. Alanis, S. X. Ng, and L. Hanzo, "Fifteen Years of Quantum LDPC Coding and Improved Decoding Strategies," *IEEE Access*, vol. 3, pp. 2492-2519, 2015. doi. 10.1109/ACCESS.2015.2503267.

[21] H. Lou, and J. Garcia-Frias "Quantum error-correction using codes with low-density generator matrix," *IEEE 6th Workshop on Signal Processing Advances in Wireless Communications*, 2005. doi: 10.1109/SPAWC.2005.1506298.

[22] H. Lou and J. Garcia-Frias, "On the Application of Error-Correcting Codes with Low Density Generator Matrix over Different Quantum Channels," *4th International Symposium on Turbo Codes & Related Topics*, 2006. online: https://ieeexplore.ieee.org/document/5755950.

[23] P. Fuentes, J. Etxezarreta Martinez, P. M. Crespo, and J. Garcia-Frias, "Approach for the construction of non-Calderbank-Steane-Shor low-density-generator-matrix based quantum codes," *Phys. Rev. A*, vol. 102, pp. 012423, 2020. doi: 10.1103/PhysRevA.102.012423.

[24] P. Fuentes, J. Etxezarreta Martinez, P. M. Crespo, and J. Garcia-Frias, "Design of low-density-generator-matrix–based quantum codes for asymmetric quantum channels," *Phys. Rev. A*, vol. 103, pp. 022617, 2021. doi: 10.1103/PhysRevA.103.022617.

[25] H. Ollivier, and J. P. Tillich, "Description of a quantum convolutional code," *Phys. Rev. Lett.*, vol. 91, 2003. doi: 10.1103/PhysRevLett.91.177902.

[26] M. Grassl and M. Rotteler, "Non-catastrophic en- coders and encoder inverses for quantum convolutional codes," *Proc. ISIT IEEE*, pp. 1019, 2006. doi: 10.1109/ISIT.2006.261956.

[27] J. G. D. Forney, M. Grassl, and S. Guha, "Con- volutional and tail-biting quantum error-correcting codes," *IEEE Trans. on Inf. Theory*, vol. 53, no. 3, pp. 865-880, 2007. doi: 10.1109/TIT.2006.890698.

[28] D. Poulin, J. P. Tillich, and H. Ollivier, "Quantum Serial Turbo Codes," *IEEE Trans. Inf. Theory*, vol. 55, no. 6, pp. 2776-2798, 2009. doi: 10.1109/TIT.2009.2018339.

[29] M. M. Wilde, M. Hsieh, and Z. Babar, "Entanglement-Assisted Quantum Turbo Codes," *IEEE Trans. Inf. Theory*, vol. 60, no. 2, pp. 1203-1222, 2014. doi: 10.1109/TIT.2013.2292052.

[30] 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. 633, 2019. doi: 10.3390/e21070633.

[31] J. Etxezarreta 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, 2019. doi: 10.3390/e21121133.

[32] A. Y. Kitaev, "Quantum computations: Algorithms and error correction," *Russ. Math. Surveys*, vol. 52, no. 6, pp. 1191-1249, 1997. doi: 10.1070/rm1997v052n06abeh002155.

[33] E. Dennis, A. Kitaev, A. Landahl, and J. Preskill, "Topological quantum memory," *J. Math. Phys.*, vol. 43, pp. 4452-4505, 2002. doi: 10.1063/1.1499754.

[34] A. Y. Kitaev, "Fault-tolerant quantum computation by anyons," *Ann. Phys.*, vol. 303, pp. 2, 2003. doi: 10.1016/S0003-4916(02)00018-0.

[35] A. G. Fowler, M. Mariantoni, J. M. Martinis, A. N. Cleland, "Surface codes: Towards practical large-scale quantum computation," *Phys. Rev. A*, vol. 86, pp. 032324, 2012. doi: 10.1103/PhysRevA.86.032324.

[36] D. Aharonov, and M. Ben-Or, "Fault-tolerant quantum computation with constant error," *Proceedings of the twenty-ninth annual ACM symposium on Theory of computing*, pp. 176–188, 1997. online: arXiv:quant-ph/9611025.

[37] E. Knill, R. Laflamme, and W. H. Zurek, "Resilient quantum computation: Error Models and Thresholds," *Proc. Roy. Soc. A*, vol. 454, pp. 342, 1998. doi: 10.1098/rspa.1998.0166.

[38] R. Laflamme, C. Miquel, J. P. Paz, and W. H. Zurek, "Perfect quantum error correction code," *Phys. Rev. Lett.*, vol. 77, pp. 198, 1996. doi: 10.1103/PhysRevLett.77.198.

[39] J. Preskill, "Fault-tolerant quantum computation," *Introduction to quantum computation*, 1999. arXiv:quant-ph/9712048.

[40] J. 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. doi: 10.1109/TIT.2013.2292061.

[41] A. Leverrier, J. Tillich and G. Zémor, "Quantum Expander Codes," *2015 IEEE 56th Annual Symposium on Foundations of Computer Science*, pp. 810-824, 2015. doi: 10.1109/FOCS.2015.55.

[42] M. B. Hastings, J. Haah, and R. O'Donnell, "Fiber Bundle Codes: Breaking the $N^{\frac{1}{2}}$ polylog(N) Barrier for Quantum LDPC Codes," 2020. arXiv:2009.03921.

[43] P. Panteleev and G. Kalachev, "Quantum LDPC Codes with Almost Linear Minimum Distance," 2020. arXiv:2012.04068.

[44] N. P. Breuckmann, J. N. Eberhardt, "Balanced Product Quantum Codes," 2020. arXiv:2012.09271.

[45] N. P. Breuckmann, J. N. Eberhardt, "Homological product codes," *Proceedings of the forty-sixth annual ACM symposium on Theory of computing (STOC '14)*, pp. 273–282., 2014. doi: 10.1145/2591796.2591870.

[46] C. Berrou, A. Glavieux, and P. Thitimajshima, "Near Shannon limit error-correcting coding and decoding: Turbo-codes," *Proceedings of ICC '93 - IEEE International Conference on Communications*, vol. 2, pp. 1064-1070, 1993. doi: 10.1109/ICC.1993.397441.

[47] 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. doi: 10.1109/18.910572.

[48] J. Pearl, "Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference," *Morgan Kauffman*, 1988. online: https://www.sciencedirect.com/book/9780080514895/probabilistic-reasoning-in-intelligent-systems.

[49] G. D. Forney, "The Viterbi Algorithm," *Proceedings of the IEEE*, vol.61, no. 3, pp. 268-278 1973. doi: 10.1109/PROC.1973.9030

[50] N.Wiberg, "Codes and decoding on general graphs," *PhD dissertation*, 1996. online: https://www.semanticscholar.org/paper/Codes-and-Decoding-on-General-Graphs-Wiberg/eb44d50bce92b4ce2c0ea53bd8ede95f628ee3cb.

[51] M. B. Hastings, "Quantum belief propagation," *Phys. Rev. B*, vol.76, pp. 201102, 2007. doi: 10.1103/PhysRevB.76.201102.

[52] M. Leifer and D. Poulin, "Quantum graphical models and belief propagation," *Annals of Physics*, vol.323, no. 8, pp. 1899-1946, 2007. doi: 10.1016/j.aop.2007.10.001

[53] 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, 1998. doi: 10.1103/PhysRevA.57.830.

[54] G. Smith and J. A. Smolin, "Degenerate Quantum Codes for Pauli Channels," *Phys. Rev. Lett.*, vol.98, pp. 030501, 2007. doi: 10.1103/PhysRevLett.98.030501.

[55] D. Poulin and Y. Chung, "On the iterative decoding of sparse quantum codes," *QIC*, vol.8, no.10 pp. 987, 2008. doi: 10.5555/2016985.2016993.

[56] M. H. Hsieh and F. Le Gall, "NP-hardness of decoding quantum error-correction codes," *Phys. Rev. A*, vol. 83, pp. 052331, 2010. doi: 10.1103/PhysRevA.83.052331.

[57] A. R. Calderbank and P. W. Shor, "Good quantum error correcting codes exist," *Physical Review A*, vol. 54, pp. 1098-1105, 1996. doi: 10.1103/PhysRevA.54.1098.

[58] A. Steane, Proceedings of The Royal Society A Mathematical, "Multiple-particle interference and quantum error correction," *Physical and Engineering Sciences*, vol. 452, no. 1954, 1996. doi: 10.1098/rspa.1996.0136.

[59] M-H. Hsieh, I. Devetak, and T. Brun, "General entanglement-assisted quantum error-correcting codes," *Physical Review A*, vol. 76, pp. 062313, 2007. doi: 10.1103/PhysRevA.76.062313.

[60] M-H. Hsieh, T. Brun, and I. Devetak, "Entanglement-assisted quantum quasicyclic low-density parity-check codes," *Physical Review A*, vol. 79, pp. 032340, 2009. doi: DOI:10.1103/PhysRevA.79.032340.

[61] Y. Fujiwara and V. D. Tonchev, "A Characterization of Entanglement-Assisted Quantum Low-Density Parity-Check Codes," *IEEE Transactions on Information Theory*, vol. 59, no. 6, pp. 3347-3353. doi: 10.1109/TIT.2013.2247461.

[62] D. Maurice, J.-P. Tillich, and I. Andriyanova, "A family of quantum codes with performances close to the hashing bound under iterative

decoding," *IEEE International Symposium on Information Theory*, pp. 907-911, 2013. doi: 10.1109/ISIT.2013.6620358.2013.

[63] T. Camara, H. Ollivier, and J.-P. Tillich, "A class of quantum LDPC codes: construction and performances under iterative decoding," *IEEE International Symposium on Information Theory*, pp. 811-815, 2007. doi: 10.1109/ISIT.2007.4557324.

[64] D. Maurice, J.-P. Tillich, and I. Andriyanova, "Spatially coupled quantum LDPC codes," *IEEE Information Theory Workshop*, 2012. doi: 10.1109/ITW.2012.6404686.

[65] A. R. Calderbank and P. W. Shor, "Good quantum error correcting codes exist," *Phys. Rev. A*, vol. 54, pp. 1098-1105, 1996. doi: 10.1103/PhysRevA.54.1098.

[66] A. R. Calderbank, E. M. Rains, P. W. Shor, and N. J. A. Sloane, "Quantum error correction via codes over GF(4)," *IEEE Trans. Inf. Theory*, vol. 44, pp.1369–1387, 1998. doi: 10.1109/18.681315.

[67] B. Yoshida and I. L. Chuang, "Framework for classifying logical operators in stabilizer codes," *Phys. Rev. A*, vol. 81, pp. 052302, 2010. doi: 10.1103/PhysRevA.81.052302.

[68] E. Pelchat and D. Poulin, "Degenerate Viterbi Decoding," *IEEE Trans. on Information Theory*, vol. 59, no. 6, pp. 3915 - 3921, 2013. doi: 10.1109/TIT.2013.2246815.

[69] S. Varsamopoulos, B. Criger, and K. Bertels, "Decoding small surface codes with feedforward neural networks," *Quantum Science and Technology*, vol. 3, no. 1, pp. 015004, 2017. doi: 10.1088/2058-9565/aa955a.

[70] P. Iyer and D. Poulin, "Hardness of decoding quantum stabilizer codes," *IEEE Trans. on Information Theory*, vol. 61, no. 9, pp. 5209-5223, 2015. doi: 10.1109/TIT.2015.2422294.

[71] D. Poulin, "Optimal and Efficient Decoding of Concatenated Quantum Block Codes," *Phys. Rev. A*, vol. 74, pp. 052333, 2006. doi: 10.1103/PhysRevA.74.052333.

[72] D. K. Tuckett, A. S. Darmawan, T. C. Chubb, S. Bravyi, S. D. Bartlett, and S. T. Flammia, "Tailoring Surface Codes for Highly Biased Noise," *Phys. Rev. X*, vol. 9, no. 4, pp. 041031, 2019. doi: 10.1103/PhysRevX.9.041031.

[73] D. K. Tuckett, "Tailoring surface codes: Improvements in quantum error correction with biased noise," *PhD Dissertation*, University of Sydney, 2020.

[74] E. Sabo, A. B. Aloshious, and K. R. Brown, "Trellis Decoding For Qudit Stabilizer Codes And Its Application To Qubit Topological Codes," arXiv:2106.08251, 2021.

[75] M. A. Nielsen, and I. Chuang, "Quantum Computation and Quantum Information: 10th Anniversary Edition," *Cambridge: Cambridge University Press*, 2011. doi: 10.1017/CBO9780511976667.

[76] A. E. Gamal and Y-H. Kim. "Network Information Theory," Cambridge University Press, 2012. USA. arXiv: 1001.3404.

[77] A. M. Turing, "On Computable Numbers, with an Application to the Entscheidungsproblem," *Proc. Lond. Math. Soc.*, vol. 2, no. 42, pp. 230, 1937. doi: 10.1112/plms/s2-42.1.230.

[78] R. L. Rivest, A. Shamir, and L. Adleman, "A method for obtaining digital signatures and public-key cryptosystems," *Communications of the ACM*, vol. 21, no. 2, pp. 120–126, 1978. doi: 10.1145/359340.359342.

[79] J. W. Bos, J. A. Halderman, N. Heninger, J. Moore, M. Naehrig, and E. Wustrow E, "Elliptic Curve Cryptography in Practice," *International Conference on Financial Cryptography and Data Security*, pp. 157-175, Springer, Berlin, 2014. doi: 10.1007/978-3-662-45472-5_11.

[80] Satoshi Nakamoto, "Bitcoin : A Peer-to-Peer Electronic Cash System," 2009. online: https://bitcoin.org/bitcoin.pdf.

[81] Google Quantum AI, "Exponential suppression of bit or phase errors with cyclic error correction," *Nature*, vol. 595, pp. 383–387, 2021. doi: 10.1038/s41586-021-03588-y.

[82] J. M. Gambetta, J. M. Chow, and M. Steffen, "Building logical qubits in a superconducting quantum computing system," *npj Quantum Inf*, vol. 3, no. 2, 2017. doi: 10.1038/s41534-016-0004-0.

[83] L. Henriet, L. Beguin, A. Signoles, T. Lahaye, A. Browaeys, G. O. Reymond, C. Jurczak, "Quantum computing with neutral atoms," *Quantum*, vol. 4, no. 327, 2020. doi: 10.22331/q-2020-09-21-327.

[84] S. Weidt, J. Randall, S. C. Webster, K. Lake, A. E. Webb, I. Cohen, T. Navickas, B. Lekitsch, A. Retzker, and W. K. Hensinger, "Trapped-Ion Quantum Logic with Global Radiation Fields," *Phys. Rev. Lett.*, vol. 117, no. 22, pp. 220501, 2016. doi: 10.1103/PhysRevLett.117.220501.

[85] D, Niemietz, P. Farrera, S. Langenfeld, and G. Rempe, "Nondestructive detection of photonic qubits," *Nature*, vol. 591, pp. 570–574, 2021. doi: 10.1038/s41586-021-03290-z.

[86] F. Jelezko, J. Wrachtrup, "Single defect centres in diamond: A review," *Special Issue: Selected Topics in Physics and Applications of CVD Diamond*, vol. 203, no. 13, pp. 3207-3225, 2006. doi: 10.1002/pssa.200671403.

[87] P. A. Dirac, "The principles of Quantum mechanics," *Oxford University Press*, 1982.

[88] H. K. Lo, S. Popescu, and T. Spiller, "Introduction to Quantum Computation and Information," *World Scientific*, 1998. doi: https://doi.org/10.1142/3724.

[89] M. Born, "The Born-Einstein letters," Walker, 1971.

[90] R. Bedington, J.M. Arrazola, and A. Ling, "Progress in satellite quantum key distribution," *npj Quantum Inf*, vol. 3, no. 30, 2017. doi:10.1038/s41534-017-0031-5.

[91] R. Renner, "Security of Quantum Key Distribution," *PhD dissertation*, Swiss Federal Institute of Tech., Zurich, Switzerland, 2005. arXiv:quant-ph/0512258.

[92] A. Cabello, "Quantum Key Distribution in the Holevo Limit," *Phys. Rev. Letters*, vol. 85, no. 26, pp. 5635, 2000. doi: 10.1103/PhysRevLett.85.5635.

[93] C. H. Bennett, G. Brassard, C. Crépeau, R. Jozsa, A. Peres, and W. K. Wootters, "Teleporting an unknown quantum state via dual classical and Einstein-Podolsky-Rosen channels," *Phys. Rev. Letters*, vol. 70, no. 13, pp. 1895-1899, 1993. doi: 10.1103/PhysRevLett.70.1895.

[94] D. Bouwmeester, J. W. Pan, K. Mattle, M. Eibl, H. Weinfurter, and A. Zeilinger, "Experimental quantum teleportation," *Nature*, vol. 390, pp. 575–579, 1997. doi:10.1038/37539.

[95] C. H. Bennett S. J. Wiesner, "Communication via one- and two-particle operators on Einstein-Podolsky-Rosen states," *Phys. Rev. Letters*, vol. 69, no. 20, pp. 2881-2884, 1993. doi: 10.1103/PhysRevLett.69.2881.

[96] J. Barreiro, TC. Wei, and P. Kwiat, "Beating the channel capacity limit for linear photonic superdense coding," *Nature Phys*, vol. 4, pp. 282–286, 2008. doi: 10.1038/nphys919.

[97] J. Etxezarreta Martinez, P. Fuentes, P. M. Crespo, and J. Garcia-Frias, "Approximating Decoherence Processes for the Design and Simulation of Quantum Error Correction Codes in Classical Computers," *IEEE Access*, vol. 8, pp. 172623-172643, 2020. doi: 10.1109/ACCESS.2020.3025619.

[98] P. Fuentes, J. Etxezarreta Martinez, P. M. Crespo, and J. Garcia-Frias, "Degeneracy and its impact on the decoding of sparse quantum codes," *IEEE Access* vol. 9, pp. 89093-89119, 2021. doi: 10.1109/ACCESS.2021.3089829.

[99] R.Penrose, "Application of negative dimensional tensors," *Combinatorial Mathematics and Its Applications*, Academic Press, pp. 221-224, 1971. online: http://homepages.math.uic.edu/ kauffman/Penrose.pdf.

[100] E. Knill and R. Laflamme, "A theory of quantum error-correcting codes," *Phys. Rev. A*, vol. 55, no. 2, pp. 900-911, 1997. doi: 10.1103/PhysRevA.55.900.

[101] C. Bennett, D. DiVincenzo, J. Smolin, and W. Wootters, "Mixed state entanglement and quantum error correction," *Phys. Rev. A*, vol. 54, no. 5, pp. 3824-3851, 1996. doi: 10.1103/PhysRevA.54.3824.

[102] P. W. Shor, "Scheme for reducing decoherence in quantum computer memory," *Phys. Rev. A*, vol. 52, no. 4, pp. R2493-R2496, 1995. doi: 10.1103/PhysRevA.52.R2493.

[103] D. Gottesman, "The Heisenberg representation of quantum computers," *Proc. Int. Conf. Group Theoretic Methods Phys.*, pp. 1–20, 1998. arXiv:quant-ph/9807006.

[104] C. E. Shannon, "A mathematical theory of communication," *Bell System Technical Journal*, vol. 27, pp. 379-423, 1948.

[105] J. Hou, P. H. Siegel, L. B. Milstein and H. D. Pfister, "Capacity-approaching bandwidth-efficient coded modulation schemes based on low-density parity-check codes," *IEEE Transactions on Information Theory*, vol. 49, no. 9, pp. 2141-2155, 2003, doi: 10.1109/TIT.2003.815777.

[106] A. Bennatan and D. Burshtein, "On the application of LDPC codes to arbitrary discrete-memoryless channels," *IEEE Transactions on Information Theory*, vol. 50, no. 3, pp. 417-438, 2004, doi: 10.1109/TIT.2004.824917.

[107] A. Bennatan and D. Burshtein, "Design and analysis of nonbinary LDPC codes for arbitrary discrete-memoryless channels," *IEEE Transactions on Information Theory*, vol. 52, no. 2, pp. 549-583, 2006, doi: 10.1109/TIT.2005.862080.

[108] E. Hof, I. Sason and S. Shamai, "Performance Bounds for Nonbinary Linear Block Codes Over Memoryless Symmetric Channels," *IEEE Transactions on Information Theory*, vol. 55, no. 3, pp. 977-996, 2009, doi: 10.1109/TIT.2008.2011443.

[109] R. W. Hamming, "Error detecting and error correcting Codes," *The Bell System technical journal*, vol. 29, no. 2, pp. 147-160, 1950. doi: 10.1002/j.1538-7305.1950.tb00463.x.

[110] R. Gallager, "Low-density parity-check codes," in *IRE Transactions on Information Theory*, vol. 8, no. 1, pp. 21-28, January 1962, doi: 10.1109/TIT.1962.1057683.

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

[112] D. J. C. MacKay, "Good error-correcting codes based on very sparse matrices," *IEEE Transactions on Information Theory*, vol. 45, no. 2, pp. 399-431, March 1999, doi: 10.1109/18.748992.

[113] D. J. C. MacKay and R. M. Neal, "Near Shannon limit performance of low density parity check codes," *Electronics Letters* , vol. 33, no. 6, pp. 457 - 458, 1997. online: https://www.semanticscholar.org/paper/Near-Shannon-Limit-Performance-of-Low-DensityParity-CodesDavid-C./27b1996dc57dd04f22a4b3aee4d0364855db4675.

[114] S. Y. Chung, G. D. Forney, T. J. Richardson and R. Urbanke, "On the design of low-density parity-check codes within 0.0045 dB of the Shannon limit," *IEEE Communications Letters*, vol. 5, no. 2, pp. 58-60, 2001. doi: 10.1109/4234.905935.

[115] L. Chen, J. Xu, I. Djurdjevic, and S. Lin, "Near-Shannon-limit quasi-cyclic low-density parity-check codes," *IEEE Transactions on Communications*, vol. 52, no. 7, pp. 1038 - 1042, 2004. doi: 10.1109/TCOMM.2004.831353.

[116] R. M. Tanner, "A recursive approach to low complexity codes," *IEEE Transactions on Information Theory*, vol. 27, no.5, pp. 533-547, 1981. doi: 10.1109/TIT.1981.1056404.

[117] N. Wiberg, H.-A. Loeliger, and R. Kotter, "Codes and iterative decoding on general graphs," *Proceedings of 1995 IEEE International Symposium on Information Theory*, pp. 468, 1995. doi: 10.1109/ISIT.1995.550455.

[118] I. Granada, "Rate compatible joint-source channel coding for point-to-point and multiple access channels," *PhD dissertation*, Tecnun - University of Navarra, San Sebastian, EUS, Spain, 2021. online: https://dadun.unav.edu/handle/10171/61091.

[119] C. Berrou, A. Glavieux and P. Thitimajshima, "Near Shannon limit error-correcting coding and decoding: Turbo-codes. 1," in *Proceedings of ICC '93 - IEEE International Conference on Communications*, Geneva, Switzerland, 1993, pp. 1064-1070 vol.2, doi: 10.1109/ICC.1993.397441.

[120] D. J. C. MacKay, "Good error-correcting codes based on very sparse matrices," *IEEE Transactions on Information Theory*, vol. 45, no. 2, pp. 399-431, 1999. doi: 10.1109/18.748992.

[121] D. J. Mackay, "Information Theory, Inference, and Learning Algorithms," *Cambridge University Press*, 2003. USA.

[122] D. Declercq and M. Fossorier, "Decoding Algorithms for Nonbinary LDPC Codes Over GF(q)," *IEEE Transactions on Communications*, vol. 55, no. 4, pp. 633 - 643, 2007. doi: 10.1109/TCOMM.2007.894088.

[123] V. S. Ganepola, R. A. Carrasco, I. J. Wassell, and S. Le Goff, "Performance study of non-binary LDPC Codes over GF(q)," *6th International*

Symposium on Communication Systems, Networks and Digital Signal Processing, pp. 585-589, 2008. doi: 10.1109/CSNDSP.2008.4610743.

[124] S. Lin and D. J. Costello, "Error Control Coding: Fundamentals and Applications," *Prentice-Hall*, 2011.

[125] E. R. Berlekamp, R. J. McEliece, and H. van Tilborg, "On the inherent intractability of certain coding problems," *IEEE Trans. Info. Theor.*, vol. 24, no. 3, pp. 384 - 386, 1978.

[126] T. M. Cover and J. A. Thomas, "Elements of Information Theory," John Wiley Sons, Hoboken, New Jersey, 2006.

[127] T. A. Brun, I. Devetak, and M. Hsieh, "Catalytic Quantum Error Correction," *IEEE Trans. Inf. Theory*, vol. 60, no. 6, pp. 3073-3089, 2014. doi: 10.1109/TIT.2014.2313559

[128] T. Brun, I. Devetak, and M. Hsieh, "Correcting Quantum Errors with Entanglement," *Science*, vol. 314, no. 5798, pp. 436-439, 2006. doi: 10.1126/science.1131563.

[129] A. R. Calderbank, E. M. Rains, P. M. 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. doi: 10.1109/18.681315.

[130] N. Rengaswamy, R. Calderbank, S. Kadhe and H. D. Pfister, "Logical Clifford Synthesis for Stabilizer Codes," *IEEE Transactions on Quantum Engineering*, vol. 1, pp. 1-17, 2020, 2020. doi: 10.1109/TQE.2020.3023419.

[131] D. Lidar, and T. Brun, "Quantum Error Correction," *Cambridge: Cambridge University Press*, 2013. doi: 10.1017/CBO9781139034807.

[132] C. Crépeau, D. Gottesman, and A. Smith, "Approximate Quantum Error-Correcting Codes and Secret Sharing Schemes," *Advances in Cryptology – EUROCRYPT 2005.* Springer, 2005. doi: 10.1007/11426639_17.

[133] G. Smith and J. A. Smolin, "Degenerate Quantum Codes for Pauli Channels," *Phys. Rev. Lett.*, vol. 98, no.3, pp. 030501, 2007. doi: 10.1103/PhysRevLett.98.030501.

[134] N. Raveendran and B. Vasić, "Trapping Sets of Quantum LDPC Codes," arXiv:2012.15297, 2020

[135] A. Rigby, J. C. Olivier, and P. Jarvis, "Modified belief propagation decoders for quantum low-density parity-check codes," *Physical Review A*, vol. 100, no. 1, pp. 012330, 2019. doi: 10.1103/PhysRevA.100.012330.

[136] P. Hu and H. Zhao, "Improved method for detecting the short cycles of LDPC codes," *2010 IEEE International Conference on Information Theory and Information Security*, pp. 841-844, 2010. doi: 10.1109/ICITIS.2010.5689706.

[137] M. Karimi and A. H. Banihashemi, "Counting Short Cycles of Quasi Cyclic Protograph LDPC Codes," *IEEE Communications Letters*, vol. 16, no.3, pp. 400-403, 2012. doi: 10.1109/LCOMM.2012.020212.112311.

[138] M. Karimi and A. H. Banihashemi, "On the Tanner Graph Cycle Distribution of Random LDPC, Random Protograph-Based LDPC, and Random Quasi-Cyclic LDPC Code Ensembles," *IEEE Trans. on Information Theory*, vol. 64, no.6, pp. 4438-4451, 2018. doi: 10.1109/TIT.2018.2805906.

[139] T. Tian, C. R. Jones, J. D. Villasenor, and R. D. Wesel, "Selective avoidance of cycles in irregular LDPC code construction," *IEEE Trans. on Communications*, vol. 52, no.8, pp. 1242-1247, 2004. doi: 10.1109/TCOMM.2004.833048.

[140] J. L. Kim, U.N. Peled, I. Perepelitsa, V. Pless, and S. Friedland, "Explicit construction of families of LDPC codes with no 4-cycles," *IEEE Trans. Inform. Theory*, vol. 50, no.10, pp. 2378-2388, 2004. doi: 10.1109/TIT.2004.834760.

[141] D. V. Nguyen, S. K. Chilappagari, M. W. Marcellin, and B. Vasic, "On the Construction of Structured LDPC Codes Free of Small Trapping Sets," *IEEE Trans. Inform. Theory*, vol. 58, no.4, pp. 2280 - 2302, 2012. doi: 10.1109/TIT.2011.2173733.

[142] G. Lechner, "The effect of cycles on binary message-passing decoding of LDPC codes," *2010 Australian Communications Theory Workshop (AusCTW)*, pp. 43-47, 2010. doi: 10.1109/AUSCTW.2010.5426759.

[143] J. A. McGowan and R. C. Williamson, "Loop removal from LDPC codes," *Proceedings 2003 IEEE Information Theory Workshop*, pp. 230-233, 2003. doi: 10.1109/ITW.2003.1216737.

[144] M. Hagiwara and H. Imai, "Quantum quasi-cyclic LDPC codes," *Proc. IEEE Int. Symp. Inf. Theory*, pp. 806–810, 2007. doi: 10.1109/ISIT.2007.4557323.

[145] M. Hagiwara, K. Kasai, H. Imai, and K. Sakaniwa, "Spatially coupled quasi-cyclic quantum LDPC codes," *Proc. IEEE Int. Symp. Inf. Theory*, pp. pp. 638–642, 2011. doi: 10.1109/ISIT.2011.6034208.

[146] K. Kasai, M. Hagiwara, H. Imai, and K. Sakaniwa, "Non-binary quasi-cyclic quantum LDPC codes," *IEEE Int. Symp. Inf. Theory Proceedings*, 637-657, 2011. doi: 10.1109/ISIT.2011.6034212.

[147] K. Kasai, M. Hagiwara, H. Imai and K. Sakaniwa, "Quantum Error Correction Beyond the Bounded Distance Decoding Limit," *IEEE Transactions on Information Theory*, vol. 58, no. 2, pp. 1223-1230, 2012. doi: 10.1109/TIT.2011.2167593.

[148] K. Chung and J. Heo, "Improved Belief Propagation (BP) Decoding for LDPC Codes with a large number of short cycles," *2006 IEEE 63rd Vehicular Technology Conference*, pp. 1464-1466, 2006. doi: 10.1109/VETECS.2006.1683078.

[149] Y. Wang, B. C. Sanders, B. Bai and X. Wang, "Enhanced Feedback Iterative Decoding of Sparse Quantum Codes," *IEEE Trans. Inform. Theory*, vol. 58, no. 2, pp. 1231-1241, 2012. doi: 10.1109/TIT.2011.2169534.

[150] N. Raveendran, "Trapping Sets of Iterative Decoders for Quantum and Classical Low-Density Parity-Check Codes," *PhD Dissertation*, The University of Arizona, 2021. online: https://www.proquest.com/docview/2598659782?pq-origsite=gscholar&fromopenview=true.

[151] P. Panteleev and G. Kalachev, "Degenerate quantum LDPC codes with good finite length performance," arXiv:1904.02703, 2019.

[152] J. Roffe, D. R. White, S. Burton, and E. T. Campbell, "Decoding Across the Quantum LDPC Code Landscape," *Phys. Rev. Research*, vol. 2, no. 4, pp. 043423, 2020. doi: 10.1103/PhysRevResearch.2.043423.

[153] Kao-Yueh Kuo and Ching-Yi Lai, "Refined Belief Propagation Decoding of Sparse-Graph Quantum Codes," *IEEE Journal on Selected Areas in Information Theory*, vol. 1, no. 2, pp. 487-498, 2020. doi: 10.1109/JSAIT.2020.3011758.

[154] K.-Y. Kuo and C.-Y. Lai, "Exploiting degeneracy in belief propagation decoding of quantum codes," 2021. arXiv:2104.13659.

[155] J. Garcia-Frias and K. Liu, "Design of near-optimum quantum error-correcting codes based on generator and parity-check matrices of LDGM codes," *42nd. Annual Conference on Information Sciences and Systems*, pp. 562-567, 2008. doi: 10.1109/CISS.2008.4558588.

[156] T. Camara, H. Ollivier and J. Tillich, "A class of quantum LDPC codes: construction and performances under iterative decoding," *2007 IEEE International Symposium on Information Theory*, 2007, pp. 811-815, doi: 10.1109/ISIT.2007.4557324.

[157] Z. Babar, P. Botsinis, D. Alanis, S. Xin Ng and L. Hanzo, "Construction of Quantum LDPC Codes From Classical Row-Circulant QC-LDPCs," *IEEE Communications Letters*, vol. 20, no. 1, pp. 9-12, 2016, doi: 10.1109/LCOMM.2015.2494020.

[158] K. Kuo and C. Lu, "A further study on the encoding complexity of quantum stabilizer codes," *2010 International Symposium On Information Theory & Its Applications*, 2010, pp. 1041-1044, doi: 10.1109/ISITA.2010.5649496.

[159] M. Wilde, "Logical operators of quantum codes," *Phys. Rev. A*, vol. 79, pp. 062322, 2009. doi: 10.1103/PhysRevA.79.062322.

[160] P. Fuentes, J. Etxezarreta Martinez, P. M. Crespo, and J. Garcia-Frias, "Design of LDGM-based quantum codes for asymmetric quantum channels," *Phys. Rev. A*, vol. 103, pp. 022617, 2021. doi: 10.1103/PhysRevA.103.022617.

[161] P. Fuentes, J. Etxezarreta Martinez, P. M. Crespo, and J. Garcia-Frías, "Performance of non-CSS LDGM-based quantum codes over the Misidentified Depolarizing Channel," *IEEE International Conference on Quantum Computing and Engineering (QCE20)*, 2020. doi:10.1109/QCE49297.2020.00022.

[162] M. Jeruchim, "Techniques for Estimating the Bit Error Rate in the Simulation of Digital Communication Systems," *IEEE J. Selected Areas Commun,* **1984**, *2*, 153–170. doi: 10.1109/JSAC.1984.1146031.

[163] K. R. Colladay and E. J. Mueller, "Rewiring stabilizer codes," *New J. Phys.*, vol. 20, pp. 083030, 2018. doi: 10.1088/1367-2630/aad8dd.

[164] 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. 29, no. 1, pp. 970-1010, 2019. doi: 10.1109/COMST.2018.2861361.

[165] 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. doi: 10.1109/ACCESS.2015.2405533.

[166] C. Berrou, R. Pyndiah, P. Adde, C. Douillard and R. Le Bidan, "An overview of turbo codes and their applications," *The European Conference on Wireless Technology*, pp. 1-9, 2005. doi: 10.1109/ECWT.2005.1617639.

[167] D. J. C. MacKay and R. M. Neal, "Near Shannon limit performance of low density parity check codes," *Electronics Letters* , vol. 33, no. 6, pp. 457 - 458, 1997. online: https://www.semanticscholar.org/paper/Near-Shannon-Limit-Performance-of-Low-DensityParity-CodesDavid-C./27b1996dc57dd04f22a4b3aee4d0364855db4675.

[168] S. Y. Chung, G. D. Forney, T. J. Richardson and R. Urbanke, "On the design of low-density parity-check codes within 0.0045 dB of the Shannon limit," *IEEE Communications Letters*, vol. 5, no. 2, pp. 58-60, 2001. doi: 10.1109/4234.905935.

[169] L. Chen, J. Xu, I. Djurdjevic, and S. Lin, "Near-Shannon-limit quasi-cyclic low-density parity-check codes," *IEEE Transactions on Communications*, vol. 52, no. 7, pp. 1038 - 1042, 2004. doi: 10.1109/TCOMM.2004.831353.

[170] Z. Babar, S. X. Ng, and L. Hanzo, "EXIT-Chart-Aided Near-Capacity Quantum Turbo Code Design," *IEEE Transactions on Vehicular Technology*, vol. 64, pp. 866-875, 2015. doi: 10.1109/TVT.2014.2328638.

[171] 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. 10194-10209, 2016. doi: 10.1109/ACCESS.2016.2591910.

[172] 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. 52712-52730, 2019. doi: 10.1109/ACCESS.2019.2911515.

[173] J. Preskill, "Fault-tolerant quantum computation," 1997. arXiv:quant-ph/9712048

[174] R. Chao abd B.W. Reichardt, "Fault-tolerant quantum computation with few qubits," *npj Quantum Information*, vol. 4, no. 42, 2018. doi: 10.1038/s41534-018-0085-z.

[175] T. R. Oenning and J. Moon, "A low-density generator matrix interpretation of parallel concatenated single bit parity codes," *IEEE Transactions on Magnetics*, vol. 37, pp. 737-741, 2001. doi: 10.1109/20.917609.

[176] K. Liu and J. Garcia-Frias, "Optimization of LDGM-based quantum codes using Density Evolution," *48th Annual Allerton Conference on Communication, Control, and Computing*, 2010. doi: 10.1109/ALLERTON.2010.5707001.

[177] I. Granada, P. M. Crespo, and J. Garcia-Frías, "Asymptotic BER EXIT chart analysis for high rate codes based on the parallel concatenation of analog RCM and digital LDGM codes," EURASIP J Wireless Com Network, vol. 11, 2019. doi: 10.1186/s13638-018-1330-z.

[178] I. Granada, P. M. Crespo, and J. Garcia-Frías, "Combining the Burrows-Wheeler Transform and RCM-LDGM Codes for the Transmission of Sources with Memory at High Spectral Efficiencies," Entropy, vol. 21, pp. 378, April 2019. doi: 10.3390/e21040378.

[179] M. M. Wilde, "Quantum Information Theory," *Cambridge Univ. Press*, Cambridge, 2013. doi: https://doi.org/10.1017/CBO9781139525343.

[180] J. Etxezarreta Martinez, P. Fuentes, P. M. Crespo, and J. Garcia-Frias, "Quantum outage probability for time-varying quantum channels," 2021. arXiv:2108.13701.

[181] W. Zhong, H. Chai and J. Garcia-Frias, "Approaching the shannon limit through parallel concatenation of regular LDGM codes," in *Proceedings. International Symposium on Information Theory*, 2005. ISIT 2005., Adelaide, SA, Australia, 2005, pp. 1753-1757, doi: 10.1109/ISIT.2005.1523646.

[182] S. ten Brink, "Code doping for triggering iterative decoding convergence," *Proceedings. 2001 IEEE International Symposium on Information Theory,* Washington, DC, USA, 2001. doi: 10.1109/ISIT.2001.936098.

[183] Kejing Liu and J. Garcia-Frias, "Asymptotic analysis of LDGM-based quantum codes," *43rd Annual Conference on Information Sciences and Systems*, pp. 87-92, 2009. doi: 10.1109/CISS.2009.5054696.

[184] W. Zhong and J. Garcia-Frias, "LDGM Codes for Channel Coding and Joint Source-Channel Coding of Correlated Sources," *EURASIP Journal on Applied Signal Processing*, vol. 2005, no. 6, pp. 942-953, May 2005. doi: 10.1109/ICIP.2003.1247031.

[185] Y. Xie, J. Li, R. Malaney, and J. Yuan, "Channel identification and its impact on quantum LDPC code performance," *Australian Communications Theory Workshop (AusCTW)*, pp. 140-144, 2012. doi: 10.1109/AusCTW.2012.6164921.

[186] Y. Xie, J. Li, R. Malaney, and J. Yuan, "Improved quantum LDPC decoding strategies for the misidentified quantum depolarization channel," *24th European Signal Processing Conference (EUSIPCO)*, pp. 493-497, 2016. doi: 10.1109/EUSIPCO.2016.7760297.

[187] J. Li, H. Liu, Z. Wang, and Xuexi Yi, "Cramér-Rao bound and quantum parameter estimation with non-Hermitian systems," 2021. arXiv:2103.07099.

[188] S. L. Braunstein and C. M. Caves, "Statistical distance and the geometry of quantum states," *Phys. Rev. Lett.*, vol. 72, no. 22, pp. 3439-3443, 1994. doi: 10.1103/PhysRevLett.72.3439.

[189] D. Collins and J. Stephens, "Depolarizing-channel parameter estimation using noisy initial states," *Physical Review A*, vol. 92, pp. 032324, 2015. doi: 10.1103/PhysRevA.92.032324.

[190] A. Fujiwara, "A Quantum channel identification problem," Physical Review A, vol. 63, pp. 042304, 2001. doi: 10.1103/PhysRevA.63.042304.

[191] J. Etxezarreta Martinez, P. Fuentes, P. M. Crespo, and J. Garcia-Frias, "Time-varying quantum channel models for superconducting qubits," *npj Quantum Inf*, vol. 7, no. 115, 2021. doi: 10.1038/s41534-021-00448-5.

[192] J. Etxezarreta Martinez, P. Fuentes, P. M. Crespo, and J. Garcia-Frías, "Pauli channel online estimation protocol for quantum turbo codes," *IEEE International Conference on Quantum Computing and Engineering (QCE20)*, pp. 102-108, 2020. doi: 10.1109/QCE49297.2020.00023.

[193] L. Ioffe and M. Mézard, "Asymmetric quantum error-correcting codes," *Physical Review A*, vol. 75, no. 3, pp. 032345, 2007. doi: 10.1103/PhysRevA.75.032345.

[194] F Schmidt-Kaler, S Gulde, M Riebe1, 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 Opt. Phys.*, vol. 36, no. 623, 2013. doi: 10.1088/0953-4075/36/3/319.

[195] L. M. K. 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, pp. 883–887, 2001. doi: 10.1038/414883a.

[196] A. M. Tyryshkin, J. J. L. Morton, S. C. Benjamin2, A. Ardavan, G. A. D. Briggs, J. W. Ager and S. A. Lyon, "Coherence of spin qubits in silicon," *Journal of Physics: Condensed Matter*, vol. 18, no. 21, pp. 783-794, 2006. doi: 10.1088/0953-8984/18/21/s06.

[197] J. R. Petta, A. C. Johnson, J. M. Taylor, E. A. Laird, A. Yacobym, D. Lukinc, M. Marcus, P. Hanson and A. C. Gossard, "Coherent Manipulation of Coupled Electron Spins in Semiconductor Quantum Dots," *Science*, vol. 309, no. 5744, pp. 2180-2184, 2005. doi: 10.1126/science.1116955.

[198] P. Bertet, I. Chiorescu, G. Burkard, K. Semba, C. Harmans, D. P. DiVincenzo, and J. Mooij, "Dephasing of a superconducting qubit in-

duced by photon noise," *Physical Review Letter*, vol. 95, no. 25, pp. 257002, 2005. doi: 10.1103/PhysRevLett.95.257002.

[199] 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. 10194-10209, 2016. doi: 10.1109/ACCESS.2016.2591910.

[200] M. Chiani and L. Valentini, "Short Codes for Quantum Channels With One Prevalent Pauli Error Type," *IEEE Journal on Selected Areas in Information Theory*, vol. 1, no. 2, pp. 480-486, 2020. doi: 10.1109/JSAIT.2020.3012827.

[201] P. K. Sarvepalli, A. Klappenecker, and M. Rötteler, "Asymmetric quantum codes: constructions, bounds and performance," *Proceedings of the Royal Society London A*, vol. 465, no. 2105, pp. 1645-1672, 2009. doi: 10.1098/rspa.2008.0439.

[202] Z. W. E. Evans, A. M. Stephens, J. H. Cole, and L. C. L. Hollenberg, "Error correction optimisation in the presence of X/Z asymmetry," 2007. arXiv:0709.3875.

[203] A. M. Stephens, Z. W. E. Evans, S. J. Devitt, and L. C. L. Hollenberg, "Asymmetric quantum error correction via code conversion," *Physical Review A*, vol. 77, no.6, pp. 062335, 2008. doi: 10.1103/PhysRevA.77.062335.

[204] C. P. Williams and S. H. Clearwater, "Explorations in Quantum Computing," *Springer*, London, 2011. online: 10.1007/978-1-84628-887-6.

[205] F. J. Vazquez-Araujo, M. Gonzalez-Lopez, L. Castedo, and J. Garcia-Frias, "Serially-Concatenated LDGM Codes for MIMO Channels," *IEEE Transactions on Wireless Communications*, vol. 6, no. 8, pp. 2860-2871, 2007. doi: 10.1109/TWC.2007.05651.

[206] P. K. Sarvepalli, A. Klappenecker, and M. Rotteler, "Asymmetric quantum LDPC codes," *IEEE International Symposium on Information Theory*, pp. 305-309, 2008. doi: 10.1109/ISIT.2008.4594997.

[207] J. Preskill, "Quantum Computing in the NISQ era and beyond," *Quantum*, vol. 2, pp. 1–79, 2018. doi: 10.22331/q-2018-08-06-79.

[208] J. Ghosh, A. G. Fowler, and M. R. Geller, "Surface code with decoherence: An analysis of three superconducting architectures," *Phys. Rev. A*, vol. 86, no. 6, pp. 062318, 2012. doi: 10.1103/PhysRevA.86.062318.

[209] M. Silva, E. Magesan, D. W. Kribs, and J. Emerson, "Experimentally scalable protocol for identification of correctable codes," *Phys. Rev. A*, vol. 78, no. 1, pp. 012347, 2008. doi: 10.1103/PhysRevA.78.012347.

[210] J. Emerson, M. Silva, O. Moussa, C. Ryan, M. Laforest, J. Baugh, D. G. Cory, and R. Laflamme, "Symmetrized characterization of noisy quantum processes," *Science*, vol. 317, no. 5846, pp. 1893–1896, 2007. doi: 10.1126/science.1145699.

[211] D. Kretschmann and R. F. Werner, "Quantum channels with memory," *Phys. Rev. A*, vol. 72, no. 6, pp. 062323, 2005. doi: 10.1103/PhysRevA.72.062323.

[212] F. Caruso, V. Giovannetti, C. Lupo, and S. Mancini, "Quantum channels and memory effects," *Rev. Mod. Phys.*, vol. 86, no. 4, pp. 1203–1259, 2014. doi: 10.1103/RevModPhys.86.1203.

[213] J. J. Burnett, A. Bengtsson, M. Scigliuzzo, D. Niepce, M. Kudra, P. Delsing, and J. Bylander, "Decoherence benchmarking of superconducting qubits," *npj Quantum Inf.*, vol. 5, no. 54, pp. 1-8, 2019. doi: 10.1038/s41534-019-0168-5.

[214] S. Schlör, J. Lisenfeld, C. Müller, A. Bilmes, A. Schneider, D. Pappas, A. V. Ustinov, and M. Weides, "Correlating Decoherence in Transmon Qubits: Low Frequency Noise by Single Fluctuators," *Phys. Rev. Lett.*, vol. 123, no. 19, pp. 190502, 2019. doi: 10.1103/PhysRevLett.123.190502.

[215] A. Stehli, J. D. Brehm, T. Wolz, P. Baity, S. Danilin, V. Seferai, H. Rotzinger, A. V. Ustinov, and M. Weides, "Coherent superconducting qubits from a subtractive junction fabrication process," *Applied Physics Letters*, vol. 117, no. 12, pp. 124005, 2020. doi: 10.1063/5.0023533.

[216] Z. Wang, S. Shankar, Z. K. Minev, P. Campagne-Ibarcq, A. Narla, and M.H. Devoret, "Cavity Attenuators for Superconducting Qubits," *Phys. Rev. Applied*, vol. 11, no. 1, pp. 014031, 2019. doi: 10.1103/PhysRevApplied.11.014031.

[217] S. Pirandola, R. Laurenza, C. Lupo et al., "Fundamental limits to quantum channel discrimination," *npj Quantum Inf.*, vol. 5, pp. 50, 2019. doi: 10.1038/s41534-019-0162-y.

[218] G. Benenti and G. Strini, "Computing the distance between quantum channels: Usefulness of the Fano representation," *J. Phys. B: At. Mol. Opt. Phys.*, vol. 43, pp. 215508, 2010. doi: 10.1088/0953-4075/43/21/215508.

[219] H. Wymeersch, F. Penna, and V. Savic, "Uniformly reweighted belief propagation: A factor graph approach," *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, pp. 2000–2004, 2012. doi: 10.1109/ISIT.2011.6033905.

[220] H. Yuan and P. Kam, "The LLR Metric for q-ary LDPC Codes with MPSK Modulation over Rayleigh Channels with Imperfect CSI," *IEEE Transactions on Communications*, vol. 60, no. 7, pp. 1793-1799, 2012. doi: 10.1109/TCOMM.2012.050812.110129.

[221] I. Granada, P. M. Crespo, M. E. Burich and J. Garcia-Frías, "Rate Compatible Modulation for Correlated Information Sources," in IEEE Access, vol. 9, pp. 65449-65465, 2021, doi: 10.1109/ACCESS.2021.3073972.

[222] N. Delfosse and J.-P. Tillich, "A decoding algorithm for CSS codes using the X/Z correlations," *IEEE International Symposium on Information Theory*, pp. 1071–1075, 2014. doi: 10.1109/ISIT.2014.6874997.

[223] N. Merhav, G. Kaplan, A. Lapidoth, and S. Shamai, "On information rates for mismatched decoders," *IEEE Trans. Inform. Theory*, vol. 40, no. 6, pp. 1953–1967, 1994. doi: 10.1109/18.340469.

[224] A. Lapidoth and P. Narayan, "Reliable communication under channel uncertainty," *IEEE Trans. Inform. Theory*, vol. 44, no. 6, pp. 2148-2177, 1998. doi: 10.1109/18.720535.

[225] A. Ganti, A. Lapidoth, and Ë™I. Emre Telatar, "Mismatched Decoding Revisited: General Alphabets, Channels with Memory, and the Wide-Band Limit," *IEEE Trans. Inform. Theory*, vol. 46, no. 7, pp 2315 - 2328, 2000. doi: 10.1109/18.887846.

[226] J. Hagenauer, E. Offer and L. Papke, "Iterative decoding of binary block and convolutional codes," *IEEE Trans. Inform. Theory*, vol. 42, no.2, pp. 429-445, 1996. doi: 10.1109/18.485714.

[227] T. Richardson and R. Urbanke, "Modern coding theory," *Cambridge University Press*, 2008. doi: 10.1017/CBO9780511791338.

[228] A. Rigby, J. Olivier, H. C. Myburgh, C. Xiao, and B. P. Salmon, "Augmented decoders for LDPC codes," *EURASIP Journal on Wireless Communications and Networking 2018*, no. 189, 2018. doi: 10.1186/s13638-018-1203-5.

[229] M. P. C. Fossorier and S. Lin, "Soft-Decision Decoding of Linear Block Codes Based on Ordered Statistics," *IEEE Transactions on Information Theory*, vol. 41, no. 5, pp. 1379-1396, 1995. doi: 10.1109/18.412683.

[230] J. Zhang and M. P. C. Fossorier, "Shuffled iterative decoding," *IEEE Transactions on Communications*, vol. 53, pp. 209–213, 2005. doi: 10.1109/TCOMM.2004.841982.

[231] E. Sharon, S. Litsyn, and J. Goldberger, "Efficient serial message-passing schedules for LDPC decoding," *IEEE Trans. Inf. Theory*, vol. 53, pp. 4076–4091, 2007. doi: 10.1109/TIT.2007.907507.

[232] J. Chen and M. P. C. Fossorier, "Near optimum universal belief propagation based decoding of low-density parity check codes," *IEEE Transactions on Communications*, vol. 50, pp. 406–414, 2002. doi: 10.1109/26.990903. doi: 10.1109/26.990903.

[233] J. Chen, A. Dholakia, E. Eleftheriou, M. P. C. Fossorier, and X.-Y. Hu, "Reduced-complexity decoding of LDPC codes," *IEEE Transactions on Communications*, vol. 53, pp. 1288–1299, 2005. doi: 10.1109/TCOMM.2005.852852.

[234] M. R. Yazdani, S. Hemati, and A. H. Banihashemi, "Improving belief propagation on graphs with cycles," *IEEE Communications Letters*, vol. 8, pp. 57–59, 2004. doi: 10.1109/LCOMM.2003.822499.