AALBORG UNIVERSITY
DENMARK

Master's Thesis
Mathematical Engineering

Quantum Error-Correction for Distributed Quantum Computing

Decoding Performance of Entanglement-Assisted Stabilizer Codes with Noisy Ebits

Author:
Jakob Kaltoft Søndergaard

Supervisors:
Jan Østergaard
René Bødker Christensen

June 3, 2024

This project is written in LaTeX.

Mathematical Sciences - Aalborg University
Skjernvej 4A, 9220 Aalborg Ø
http://math.aau.dk

AALBORG UNIVERSITY
STUDENT REPORT

Title:
Quantum Error-Correction for Distributed Quantum Computing

Theme:
Decoding Performance of Entanglement-Assisted Stabilizer Codes with Noisy Ebits

Project Period:
September 2023 - June 2024

Author:
Jakob Kaltoft Søndergaard

Supervisors:
Jan Østergaard
René Bødker Christensen

Copies: 1

Numbered Pages: 134

Date of Completion:
June 3, 2024

Abstract:

The potential of quantum computers to disrupt many industries has captured the interest from research communities and Tech Giants. The main obstacle for quantum computers in reaching their potential is the high error rate in such computers. This thesis discusses how such errors may be corrected with quantum error-correcting codes. Particularly, general theory of quantum error-correction and the most popular type of quantum codes, stabilizer, is presented. It is thereafter generalised to entanglement-assisted (EA) codes, which are suitable codes for distributed quantum computing as they require that the sender and receiver share entanglement before the communication as in many quantum communication protocols. The general assumption of EA codes is that the receivers part of the shared entanglement is noiseless, which in practice is hard to enforce. This thesis therefore analyses the effect of removing this assumption.

An approach to analyse the effect of errors from the depolarising channel is derived for the $[[6, 1, 3; 1]]$ EA code. Based on this, it is analysed which error the code can correct, whereafter the performance of the code is measured with respect to how likely correctable errors are to occur. In the analysis, it is found that the code can correct some errors even when the assumption is removed, however, the performance of the code reduces significantly when the receiver's part of the shared entanglement is corrupted by noise. Similar results are shown for the $[[5, 1, 3; 2]]$ EA code, where the $[[5, 1, 3; 2]]$ EA code generally has better performance than the $[[6, 1, 3; 1]]$ EA code. It is furthermore seen that the decoding performance for the $[[5, 1, 3; 2]]$ EA code decreases when the number of errors on the receiver's part of the shared entanglement increases.

Mathematical Sciences - Aalborg University
Skjernvej 4A, 9220 Aalborg Ø
http://math.aau.dk

AALBORG UNIVERSITY
STUDENT REPORT

Titel:
Quantum Error-Correction for Distributed Quantum Computing

Emne:
Decoding Performance of Entanglement-Assisted Stabilizer Codes with Noisy Ebits

Projektperiode:
September 2023 - June 2024

Yo:
Jakob Kaltoft Søndergaard

Vejledere:
Jan Østergaard
René Bødker Christensen

Antal Kopier: 1

Nummeret Sideantal: 134

Dato for Aflevering:
June 3, 2024

Abstract:

Kvantecomputere har potentialet til at revolutionere adskillige industrier, hvilket har øget interessen for dem i forskningsmiljøer og højteknologiske virksomheder. Den største forhindring for kvantecomputere i at opnå deres potentiale er, at de er meget støjfyldte. Dette speciale undersøger hvordan det er muligt at rette fejl der opstår i kvanteberegninger ved hjælp af kvante-kodningsteori. En general kvante-kodningsteori vil blive præsenteret sammen med teorien for den mest populære type af kvantekoder, stabiliseringskoder. Dette er derefter generaliseret til sammenfiltningsstøttede koder som er en oplagt kandidat for kvantekoder for distribuerede kvanteberegninger da de bygger på at afsenderen og modtageren for sådan en kvanteberegning deler et sammenfiltret system som også er antagelsen i mange kvantekommunikationsprotokoller. Den generelle antangelse for sammenfiltningsstøttede koder er at modtagerens del af den delte sammenfiltning er fejlfri, hvilket er svært at opnå i praksis. I dette speciale er effekten af at fjerne denne antagelse derfor analyseret.
Først bliver der udviklet en måde at analysere hvordan fejl fra depolariseringskanalen påvirker den sammenfiltningsstøttede $[[6,1,3;1]$ kode. Baseret på denne metode analyseres det hvilke fejl koden kan rette, og kodens ydeevne bliver målt på hvor sandsynlige de fejl som koden kan rette er. I analysen ses det at koden stadig kan rette nogle fejl selvom antagelsen bliver fjernet. Kodens ydeevne falder dog drastisk når modtagerense del af den delte sammenfiltning ikke er fejlfri. Samme resultater vises for den sammenfiltningsstøttede $[[5,1,3;2]]$ kode, hvor det ses at $[[5,1,3;2]]$ koden generelt har bedre ydeevne end $[[6,1,3;1]]$ koden. Til sidst ses det også at $[[5,1,3;2]]$ kodens ydeevne falder når mængden af fejl på modtagerens del af den delte sammenfiltning stiger.

Preface

This master's thesis is written by Jakob Kaltoft Søndergaard from September 2023 to June 2024 as the culmination of the Master of Science in Mathematical Engineering at Aalborg University.

I would like to thank Jan Østergaard and René Bødker Christensen for inspiration to finding a research area, guidance throughout the project period, and correction of numerous errors (typos and ambiguous formulations, not phase flips).

The reader of this thesis is not assumed to have any prior knowledge on quantum computing. However, thorough knowledge about linear algebra, probability theory, information theory, and coding theory is assumed. Moreover, knowledge about the fundamental concepts of quantum mechanics is beneficial, although important terminology is explained such that the thesis is somewhat self-contained.

Citations are made with the AMS alpha method, which alphanumerically describe sources based on the last name(s) of the author(s) and year of publication. As traditionally done, citations are placed straight after presenting the idea, quote, or similar, used from the reference. However, as much of the theory presented throughout the thesis is inspired by the same source(s), a list of references used to comprehend the theory for a given chapter/section is often stated at the start of the given chapter/section (or the end of the preceding section).

Figures presented in this thesis is made with Ti*k*Z v.3.1.9a.

Aalborg Universitet, June 3, 2024.

Nomenclature and Notation

This thesis contains a mixture of several scientific fields, namely quantum mechanics, computer science, and mathematics. These fields often have different notational conventions, hence this section introduces the convention used throughout this thesis, which typically is the one found in quantum mechanics as it is the foundation of the thesis. The convention in quantum mechanics is to use the Dirac notation, also known as bra-ket notation, which is briefly summarised below.

A vector in some vector space is denoted $|\psi\rangle$, known as a ket, describes the state of a quantum system. Its dual is called a bra and is denoted $\langle\psi|$. Combining the two yields a bra-ket, $\langle\psi|\varphi\rangle$, which corresponds to the inner product between $|\psi\rangle$ and $|\varphi\rangle$. Similarly, the outer product of the two kets is $|\psi\rangle\langle\varphi|$.

Notation

Algebra	
$\mathbb{R}$	The set of real numbers
$\mathbb{C}$	The set of complex numbers
$\mathbb{F}_2$	The Galois field of order 2
F^n	The n-dimensional vector space over F, for $F \in \{\mathbb{R}, \mathbb{C}, \mathbb{F}_2\}$
$\mathcal{G}_1$	The Pauli group
$\mathcal{G}_n$	The n-fold Pauli group
$\mathcal{S}, \mathcal{B}$	Subgroup of a Pauli group
$\langle g_1, \dots, g_m\rangle$	Generators of a group
$\delta_{ij}, \delta(i,j)$	Kronecker Delta with variables i, j
Linear Algebra	
$\mathscr{H}$	Hilbert space
A, B	Quantum systems
$\mathscr{H}_A, \mathscr{H}_B$	State spaces of systems A and B
AB	Composite system of subsystems A and B
$\|\psi\rangle, \|\varphi\rangle$	Quantum states
$\langle\psi\|, \langle\varphi\|$	Dual of quantum states
$A^\dagger$	Hermitian transpose of A
α^*	Complex conjugate of α
$\otimes$	Tensor product
X, Y, Z	Pauli matrices
$\mathcal{E}, \mathcal{F}$	Quantum channels
Set Theory	
$\emptyset$	The empty set
$A \setminus B$	Set difference of the sets A and B
$\{A_i\}_i$	Indexed family with index set i
$\{0,1\}^n$	Bit string of length n
$\Pr(\cdot)$	Probability measure

Contents

1 | Problem Analysis

1.1 Introduction

Throughout the last decades, quantum technologies have experienced a continual increase in interest from scientific communities as illustrated in Figure 1.1, Tech Giants such as Google, IBM, and AWS, and innovative start-ups, all financially supported by billion of dollar investments from private investors and governments [MZ$^+$23, pp. 8,9,12,15,25], [Den19]. Particularly quantum computing has been publicised due to its potential to start a new revolution in various industries, e.g., pharmaceutics, sustainable energy, and finance [MZ$^+$23, p. 31]. The revolution is lead by a paradigm shift in the computing executed by quantum computers, which is a result of their hardware being fundamentally different from that of classical computers known today. The potential of quantum computing is a consequence of this paradigm shift, which enables solving problems that currently cannot be solved with today's computers. The importance of gaining such computational power was already proclaimed in the Ancient Greece by Plato: "*If you cannot calculate, you cannot speculate on future pleasure and your life will not be that of a human, but that of an oyster or a jellyfish.*" [Ifr01, p. 100]. The limitations of mankind are, therefore, essentially a result of lack of computational power, which manifest the potential of quantum computers. To fully grasp their potential, it is helpful to know the history of computers.

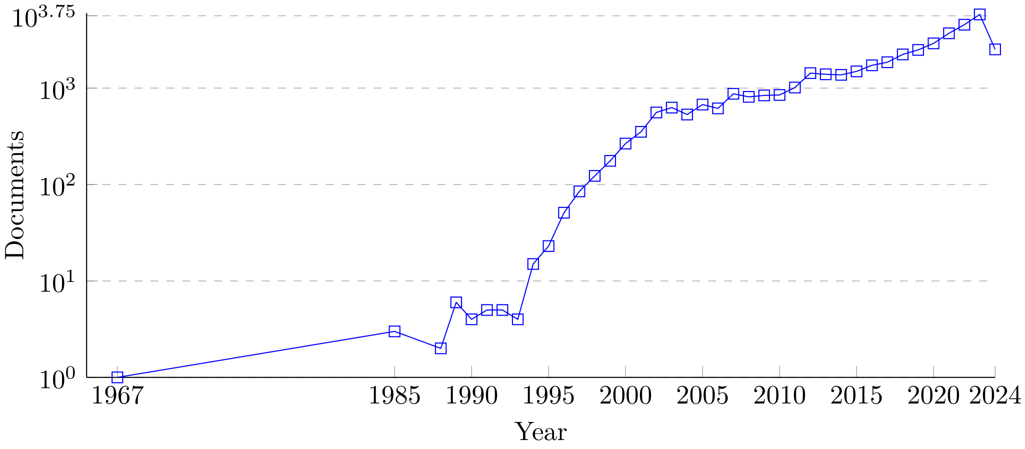


Figure 1.1: A semi-log plot of scientific documents such as papers, conference papers, and books published per year with phrases *quantum technology*, *quantum computing*, or *quantum computer* in the title, abstract, or keywords according to Scopus search the 2nd of June 2024 [Sco]. Years with no publications have no data points.

1.1.1 Computing Before Modern Computers

In ancient times, people used their fingers, and possibly also toes, to count. However, as society evolved, merchants needed to account for larger quantities, hence devices for doing so was invented. The abacus, commonly considered the earliest counting device, was invented around 2500 BCE. It was used to keep track of simple arithmetic by manually moving beads on rods [Ifr01, p. 11]. With time, civilisations progressed and so did the need for solving harder problems, particularly navigational and astronomical, hence devices capable of easing these problems were invented. A noteworthy example is the Antikythera mechanism from the 1st century BCE, which is renowned as the first analogue computer. It was used to predict the position of objects in the Solar System by turning gears, which was important for cultural reasons, but also the astronomical advancements made at the time [Ifr01, p. 155].

In the following centuries, the devices used for computations became more and more sophisticated, yet still built for specific problems, as the civilisations and sciences evolved. This culminated in the Scientific Revolution, where Blaise Pascal invented the Pascaline, the first commercialised mechanical machine capable of performing basic arithmetic. Despite its power to automate the tedious calculations performed by scientists, it never became popular [Ifr01, ch. 5.3]. Even if it had become widespread, it would be inadequate to solve many problems in science, astronomy, and engineering since such problems often rely on evaluating functions, e.g., polynomials, logarithms, and trigonometric functions. In order to use such functions, one had to use man-made look-up tables, which were not only tedious to create and inspect, but also prone to errors. These drawbacks combined with the mechanical progress during the industrial revolution led in the beginning of the 19th century Charles Babbage to imagine a mechanical machine capable of automatically creating large look-up tables of logarithmic functions based on finite differences of polynomials, hence it was named the Difference Engine. Babbage, however, kept imagining a more powerful machine, hence abandoned the Difference Engine before it was built [Ifr01, pp. 172-173] (a working example based on Babbage's design was constructed by the Science Museum in 1991 [Sci]). The new machine was called the Analytical Engine as it was capable of performing any conceivable calculation. This machine was separated into five units with their own purposes; a) an input/output unit, b) an unit moving the numbers into the correct position for calculations, c) a unit capable of storing intermediate results for speed-up or final results as output, d) a unit performing the calculations, and e) a printing unit. The units are analogous to the input/output unit, control unit, memory unit, and arithmetic logic unit that modern computers have, and is generally considered as the first programmable computer. Like the Difference Engine, the Analytical Engine was never built, however its concept have led Babbage to being considered the father of the computer [Ifr01, ch. 5.9].

Despite the concept of a programmable computer, the computers built in the following century were still designed for specific problems. A popular example is the Bombe, which was designed by Alan Turing, among others, at Bletchley Park during World War II to decipher secret messages by the Nazis that were encrypted by their Enigma machines. The Nazis believed that the Enigma was secure since its encryption was based on the settings of the machines, of which there were approximately 1.58×10^{20}. The Nazis furthermore changed the settings every day, thus, the British had to break the encryption in the matter of a day, which would be impossible by hand. However, the combination of cryptanalysis and the computational power of the Bombe enabled the British to break the encryption. It is believed by historians that the work done at Bletchley Park, which included decryption beyond the Bombe, shortened the war by up to two years [Ifr01, pp. 95-96, 218-220], [GL23].

1.1.2 The Modern Computer

Not only did Alan Turing help shortening World War II at Bletchley Park, Turing is also commonly considered the father of computer science after publishing the paper *On Computable Numbers, with an Application to the Entscheidungsproblem* in 1936 [Tur36]. In the paper, Turing conceptualised the mathematical foundation for computation by imaging a simple automatic machine that is now known as a Turing machine. At the risk of simplifying, a Turing machine consists of an infinite tape separated into squares each of which can contain a symbol from a finite set, a scanner of a single cell, and a set of elementary rules on how the machine may change the symbol of the cell that is currently scanned. These rules are limited to erasing the symbol, writing another valid symbol, moving the scanner once to the left or right, and halting. The rule applied at a particular time depends only on the symbol currently being scanned and the state of the machine. By performing these elementary operations sequentially based on the rules and state of the machine, it could automatically compute whatever the rules defined it to compute. By altering the rules and possible states of the machine, it computes something different, that is, something that can be written as an algorithm, which imply that there exists infinitely many of such Turing machines. The Pascaline, the Difference Engine, the Analytical Engine, and the Bombe are all examples of such Turing machines. Turing then continued to prove that there exists an universal Turing machine capable of computing what any Turing machine can compute, i.e., everything that is computable. Any computer capable of doing so is said to be Turing-complete, a concept which laid the theoretical foundation for computability and its limitations as well as leading a paradigm shift away from the problem-specific computers hitherto known and towards programmable computers [Ifr01, ch. 5.16].

In practice, no computer is Turing-complete as it requires infinite storage, however aside from that, all modern computers are. In fact, the Analytical Engine is generally believed to have been the first Turing-complete computer if ever built [Ifr01, pp. 220-222]. Hence, what Babbage tried to physically construct, Turing abstractly theorised, thus separating the physical construction of computers from its computability, which combined yield the computational power of a computer.

Physical Implementation

A decade after Turing's seminal paper, one of the first Turing-complete machines, ENIAC, was manufactured. However, it was large weighing around 30 tonnes, and slow to execute as data was fed to the machine by external switches and it was necessary to rewire it for re-programming. When running, it was nonetheless fast at the time, capable of performing up to $5,000$ operations each second [Ifr01, pp. 220-222]. Another important feature of ENIAC was that it was electronic, which quickly turned out to be the future of computers. This was further demonstrated in the 1950's when transistors, semiconducting components used for switching and amplification of electric currents, were implemented in computers. Compared to the vacuum tubes previously used, e.g., in ENIAC, they were smaller, faster to switch between on and off, and had a lower energy consumption, which enabled computers to decrease in size without reducing their computational power. This miniaturisation of computers was further accelerated with the inventions of MOSFETS, integrated circuits, and microchips [Ifr01, ch. 5.19]. Transistors has particularly been subject to this miniaturisation due to the computational power of a computer being positively correlated with its transistor count. This trend were observed by Gordon Moore in 1965, whereafter he projected that the amount of transistors on a microchip would double every two years or so [Moo65]. It has empirically turned out to be reasonably accurate, e.g., a few thousand transistors could be implemented on a single microchip in the early 1970's [Hit73], whereas today there can be over a hundred billion [App23]. Moore's projection is therefore known as Moore's law today. Loosely speaking, this implicates that the computational power of a computer such as ENIAC that filled an entire room is now significantly less powerful than a phone fitting in the pocket, which can perform more than 10^{13} operations per second [OW^{+}22]. Laptops and supercomputers are even more powerful, whereof Frontier is the leading system capable of performing over 10^{18} operations per second [Top23]. This development has naturally revolutionised the world drastically, and continues to do so, by enabling virtual banking, almost instantaneous world-wide communication, predictions of the future based on enormous amounts of data, large scientific simulations can be performed in the cloud, etc.

Continuation of the miniaturisation will continue this revolution, however Moore's law may soon be dead [L^{+}20]. Today, transistors belong to the 3nm process, which in size is sub-50nm and typically made of silicon [Kha16]. By shrinking transistors further, quantum tunnelling will become inevitable and introduce errors in computing due to size approaching the theoretical limit of silicon atoms. Therefore, new approaches is required in order to increase computation power in the future. Examples could be to use other materials than silicon or using a new geometric structure of the transistors, two approaches that are already large research fields [Kha16]. Another approach is to increase the computational power by designing and optimising algorithms used for computations.

Algorithmic Implementation

As previously discussed, computational power is not solely determined by hardware, but also, and generally more important, the algorithms used to solve problems [HLV10, p. 71]. Since modern computers are Turing-complete, they are capable of computing everything that is computable given infinite time, however, efficiency is a necessity in practice. Theoretically, efficiency is measured by the computational complexity of the algorithms used for solving the given problem, which loosely speaking is the limit of elementary operations performed when running the algorithm as the size/input of the problems tends to infinity, and is typically denoted $\mathcal{O}(\cdot)$. If the complexity of an algorithm is at least exponential, it is considered an intractable problem, meaning that it cannot be solved efficiently as the problem grow larger than very small, almost trivial, cases. A popular example of an intractable problem is the travelling salesman problem, in which the shortest path between n cities must be found. The naive approach is to simply calculate all paths and compare them, which has a factorial complexity, while the fastest known algorithm, the Held-Karp algorithm, has exponential complexity, more precisely $\mathcal{O}(n^2 2^n)$ [HK62, pp. 198-199]. Even in the fairly simple case with 100 cities, the Frontier supercomputer requires approximately 3×10^{132} years and 4×10^{8} years to solve the problem with the two methods, respectively. If, however, one could find an algorithm with polynomial complexity, say $\mathcal{O}(n^3)$, Frontier would be apply to solve the problem in approximately 10^{-12} seconds, which clearly illustrate the importance of having efficient algorithms.

The travelling salesman problem is only one of a whole class of problems for which there are no known efficient exact algorithm. One possible solution is to be content with either approximate or probabilistic algorithms. For many purposes, this may be sufficient, however, for some, it is crucial to known an exact solution.

Some of the problems with increasing the computational power of computers with respect to both hardware and algorithms in the future along with some potential solutions has been discussed. In fact, another potential solution exists, namely quantum computers.

1.2 Quantum Computers

The field of quantum computing began in the beginning of the 1980's, when physicists Richard Feynman held a talk on the simulation of physics with computers that concluded with: "*[...] Nature isn't classical, dammit, and if you want to make a simulation of Nature, you'd better make it quantum mechanical, and by golly it's a wonderful problem, because it doesn't look so easy.*" [Fey82, p. 486]. Thus, Feynman realised that in order to simulate physics at a larger scale, computers required another structure than classically seen, thus laying the conceptual foundation for quantum computing. At the same time, Paul Benioff illustrated how to perform calculations with a quantum computer, and thereby founding theoretical quantum computing [Pre23, p. 4]. In 1985, David Deutsch enhanced the idea of quantum computers by formally describing a quantum equivalence to universal Turing machines. Furthermore, Deutsch proposed that such quantum computers is not only advantageous to classical computers with respect to simulating physics, but there also exists quantum algorithms for quantum computers with lower complexity than their classical counterparts [Deu85]. Although with little real-life application, an important example of such quantum algorithms is the Deutsch-Jozsa algorithm (see Chapter A for further details of the algorithm), which has an exponential speed-up compared to its classical counterpart [DJ92]. Two years later, in 1994 Peter Shor presented an algorithm for factoring large prime numbers, now known as Shor's algorithm [Sho94]. The complexity of Shor's algorithm is $\mathcal{O}((\log n)^2(\log\log n)(\log\log\log n))$, whereas the most efficient classical algorithm is the number field sieve having complexity $\mathcal{O}(\exp\{1.9(\log n)^{1/3}(\log\log n)^{2/3}\})$ [Sho99, p. 317]. For a normal classical computer, it would take approximately 2×10^{18} years to factor a 2048-bit number using the number field sieve. Such numbers are therefore used in the commonly used RSA encryption-scheme as they are essentially unbreakable. However, assuming Shor's algorithm could be run on that same computer, the factoring could be performed in less than 0.03 seconds (notice that this assumption is far from holding true today, however it illustrates the power of Shor's algorithm). With the potential to break RSA, the interest for quantum computing, specifically new quantum-safe encryption-schemes, exploded after Shor's algorithm were introduced as illustrated in Figure 1.1. The interest was not only theoretical, and before the turn of the century, the first physical implementation of a quantum computer was demonstrated [CGK98]. Since then, billion of dollars has been invested in quantum computing, which has supported exceptional advancements in the field. In order to comprehend the reasoning for such large investments, the potential of quantum computing is briefly discussed by examining potential applications.

1.2.1 Quantum Algorithms

As proclaimed by Feynman, quantum computers are suitable for simulating quantum mechanical systems, however, e.g., Deutsch and Shor later showed that there are quantum algorithms that are more efficient than their classical counterparts. Today, these superior quantum algorithms generally belong to either of three cases; a) solving the hidden subgroup problem, b) search algorithms, and c) quantum simulations [NC10, ch. 1.4.5].

As the name suggests, the aim of the hidden subgroup problem is to determine a subgroup with a given property by evaluating an oracle. Shor's algorithm and the Deutsch-Jozsa algorithm are both examples of algorithms capable of solving a special case of the hidden subgroup problem with an exponential speed-up compared to the classical counterparts [NC10, p. 38]. Although such algorithms are currently believed to have modest long-term impact [Pre23, p. 199], they illustrate the potential of quantum computers.

Quantum search algorithms are, as the name suggests, used for searching for a solution in some database. Such algorithms are more applicably than those solving the hidden subgroup problem, however, their speed-ups are less impressive. The most famous quantum search algorithm is Grover's algorithm, which has a quadratic speed-up compared to its classical counterpart [Gro96]. One example of its usage is in a quantum algorithm for the travelling salesman problem, which has complexity $\mathcal{O}(1.728^n)$ [ABI+19], meaning that a quantum computer with computational power equivalent to Frontier would be capable of solving the problem with 100 cities in about 6.5 days as opposed to the 4×10^8 years in the most efficient classical case. Again, such quantum computer is far from existing, however the comparison once again illustrates the power of quantum computing. Generally, quantum computers can analyse enormous of data efficiently, hence the potential for quantum machine learning is immense as well.

Perhaps most importantly, quantum computers can be used for simulating Nature, something which classical computers cannot do efficiently for most real-life applications, e.g., the many-body problem. More specifically, quantum computers can simulate quantum systems such as chemical reactions occurring in the development of new medicine, energy sources, and food. When such simulations can be performed at a large scale, solutions to some of the crises in the world may be found [NC10, p. 39].

Having broadly described some applications of quantum algorithms, it should be clear that quantum computers are not superior to classical computers in every scenario. As a result, they are not intended to be some kind of new and faster personal computer, but a completely new type of computer for research and industry that can be utilised in solving problems that cannot be solved efficiently as of today.

To understand how quantum algorithms can be more efficient than their classical counterparts, it is necessary to understand the physical implementation of quantum computers.

1.2.2 Physical Implementation

The hardware of quantum and classical computers are fundamentally different as the former exploit quantum mechanical phenomena (see Chapter 2 for explanation of the terminology), while the performance of the latter is reduced if such phenomena occur. Classical computers use bits as the basic unit of information, which during calculations are being manipulated by logic gates build of transistors. On the other hand, quantum computers use qubits as the basic unit of information, which during calculations are manipulated, e.g., by a laser [NC10, p. 84]. The potential of quantum computers arise from the superposition principle and entanglement of qubits, which enable a function to be evaluated at all inputs simultaneously, a feature known as quantum parallelism [NC10, ch. 1.4.2]. Unfortunately, measuring the outcome of such an calculation will only yield a single function evaluation due to the probabilistic behaviour of qubits. In other words, qubits contain hidden information, however this hurdle can be overcome by using quantum algorithms utilising interference, that is, by maximising the probability of measuring the desired outcome. The Deutsch-Jozsa algorithm described in Chapter A is a great example of such.

In essence, quantum algorithms achieve their superiority over their classical counterparts by utilising the quantum mechanical phenomena originating from qubits. Thus, it is worth discussing how qubits are physically implemented. Theoretically, every two-level quantum system, e.g., the spin of an electron or polarisation of a photon, can be used as a qubit, however, some are more well-suited for computation than others. Important factors include the capability to manipulate and measure the qubits, but also a long decoherence time (see Section 3.3.3) such that the time it can be used for computation is sufficiently long. This has implicated that different hardware designs for quantum computers has been proposed whereof the most popular is superconducting qubits [MZ$^+$23, p. 24]. At the risk of simplifying, superconducting qubits are formed by placing a Josephson junction in a circuit through which Cooper pairs can tunnel. The presence or absence of such Cooper pairs then determine the state of the qubit. In order for such to work, the temperature needs to be close to absolute zero, hence they are placed inside a dilution refrigerator with a temperature around 10 mK [KSB$^+$20, pp. 371-372].

As of writing, the largest quantum computers consist of around $1,100$ qubits [Gam23], [Ato23]. In principle, quantum computers of such size are capable of factoring a 512-bit number using Shor's algorithm [PZ03, p. 336]. However, the decoherence time for qubits is too small to perform such large computation and the error rate of manipulating the qubits is so faulty, around 10^{-3} [AI23, p. 676] as opposed to 10^{-18} for classical computers [Kas19, p. 4], implicating that such a large number cannot be factored with quantum computers in practice as of today. In fact, the largest number factored with Shor's algorithm is 21 achieved in 2012 [MLLL$^+$12], whereas the largest RSA-number factored with a classical computer is a 829-bit number [GZT$^+$20]. In this regard, current quantum computers are therefore still inferior to classical computers despite their potential. The low decoherence time and high error rate of current quantum computers heavily limits their potential, hence they are said to belong to the class of noisy intermediate-scale quantum (NISQ) computers, which are not expected to revolutionise the world on their own despite the potential of quantum computing [Pre23, p. 100]. Nonetheless, Google demonstrated the potential when they executed a quantum algorithm in 200 seconds that otherwise would take approximately $10,000$ years for a supercomputer to do. Unfortunately, the algorithm has no real-life application, yet they achieved the so-called quantum supremacy [AAB$^+$19]. In order to have real-life applications, that is achieve quantum advantage, it is necessary to substantially reduce the high error rate, however, it is expected to be impossible to achieve this solely by improving the hardware [Kas19, p. 4]. Another solution is to correct the errors occurring with error-correction, however, this requires larger-scale computers, i.e., more qubits. When taking error-correction into account, it is expected the factoring an 2048-bit number requires 8 hours with a quantum computer consisting of 20 million qubits [GE21], which seem absurd given the size of current computers. It is therefore believed that the quantum advantage is achieved by distributed quantum computing, that is computations are executed on several smaller quantum computers that form a quantum computing cluster [CAF+22, p. 4]. To realise such distributed quantum computing, one need to be able to communicate quantum information, i.e., the state of qubits, between the different computers in a quantum network.

1.3 Quantum Communication

The transmission of quantum information has generally been achieved by encoding a quantum state into photons, which in turn are physically transmitted through either optical fibres or terrestrial free-space. However, these approaches has been limited to about 100 kilometres as decoherence becomes inevitable for longer transmissions [RXY+17]. Substantial work in different approaches is therefore required in order to achieve a global quantum network in the future. A popular approach is quantum teleportation, which by exploiting entanglement enables one to not even need to physically transmit quantum information but only classical information. Shortly said, Alice can communicate quantum information of a qubit to Bob by physically transmitting two bits given that they each possess an ebit of an EPR pair. Despite its name, it should be mentioned that the physical transmission of bits prohibit communication that is faster than light, which is generally understood by teleportation in popular science [NC10, ch. 1.3.7]. It is furthermore important to remark that even though it is adequate for Alice to transmit bits, which does not need a quantum network, the requirement that she and Bob share an Einstein-Podolsky-Rosen (EPR) pair impose the need for a quantum network. However, this quantum network does not necessarily need to link Alice and Bob directly as entanglement swapping can be utilised with a quantum repeater [YBL+10, ch. 5.3.1]. For comparison, the longest demonstration of quantum teleportation is $1,400$ km with satellites [RXY+17].

Another important aspect of communication is security. As previously discussed, classical encryption-schemes such as RSA are only secure as long as quantum computers belong to the class of NISQ computers. However, by once again exploiting quantum mechanical phenomena, one can achieve quantum cryptographic schemes that are theoretically safe, even post NISQ computers. The most famous example is quantum key distribution that enables Alice and Bob to establish a secret key by performing both classical and quantum communication. This key cannot be obtained by an eavesdropper, Eve, without Alice and Bob noticing, which makes it secure. After the key generation, Alice and Bob therefore possess a secure communication channel [YBL+10, ch. 3.1].

Rather than considering Alice and Bob as people that communicate but as quantum computers in a cluster, the protocols described above applies naturally to distributed quantum computing, at least in theory. However, in order for quantum communication to be applicable in distributed quantum computing no matter the protocol used, one needs to ensure that the quantum information sent has not been destroyed by decoherence or other types of noise during transmission. Error-correction is therefore also an important aspect of quantum communication.

1.4 Problem Delineation

Quantum computing and communication are extensively researched in almost every scientific field. Researched topics include the physical implementation of quantum computers such as the ability of scaling the qubit count as well as achieving high decoherence times for qubits, development of quantum software, quantum algorithms, or quantum communication protocols, and discovering real-life applications of quantum computing in various fields. All of these are important in order to realise the potential of quantum computing, however, contributing to all of these areas in this thesis is inconceivable. As previously discussed, error-correction is required in order to achieve large-scale quantum computers or a quantum network, which in turn is required for quantum algorithms to be applicable to solve real-life problems. The focus in this thesis is therefore quantum error-correction. More specifically, the issues with low decoherence times and high error rates must be overcome in order to perform quantum computations, while decoherence and other types of noise occurring during quantum communication must be diminished in order to achieve a reliable quantum network. The physical types of noise occurring during quantum computations may differ from those in quantum communication, however, all of these are covered simultaneously by considering distributed quantum computing. This limitation leads to the following initial problem statement:

How can quantum error-correcting codes be constructed in order to diminish errors occurring in distributed quantum computing?

1.5 Quantum Error-Correction

The general idea of quantum error-correction is similar to the classical case; the sender introduces redundancy when encoding the message that must be transmitted over some noisy communication channel such that potential errors hopefully can be corrected by the receiver. However, there is a trade-off between the rate of information being sent and the probability of successfully decoding the message. In 1948, Claude Shannon proved that for any rate lower than the capacity of the channel, there exists a coding scheme with error rate tending to zero as the length of the code tends to infinity [Sha48]. A variety of coding schemes have been designed since then. Unfortunately, since qubits are a generalisation of bits, the classical coding theory cannot be applied directly to the quantum case. As described in Chapter 4, there are essentially three additional challenges; a) the decoder cannot measure the state without it collapsing, b) the no-cloning theorem prohibits copying quantum information, and c) a continuum of possible errors may corrupt the quantum state. Peter Shor, nonetheless, demonstrated that none of these challenges are fatal as he designed the first quantum error-correcting code in 1995 capable of correcting an arbitrary error, including decoherence, on a single qubit [Sho95]. With that, the foundation for quantum error-correction was laid, and a quest for quantum codes began. Shortly after, it was illustrated how dual-containing classical codes could be used to create quantum codes [CS96], [Ste96]. This work were then generalised to define the most popular class of codes, namely stabilizer codes [CRSS97], [Got96].

In general, quantum error-correcting codes encodes the quantum information, say of a single qubit, into an entangled state consisting of several qubits (see Chapter 4, Chapter 5, and Chapter 6 for further information). In other words, the state is encoded into a subspace of a state space. Designing quantum error-correcting codes as such has two clear limitations. Firstly, the computational power is significantly reduced by utilising some of the physical qubits to error-correction rather than pure computations. Secondly, it requires additional qubits and operations which themselves are prone to error, thus, to reduce errors, one essentially risks introducing more errors. Without further considerations, errors will eventually propagate from qubit to qubit during long computations making them unreliable [Sho96]. It is therefore necessary to perform operations without the risk of potential errors propagating, that is being fault-tolerant, which in turn require additional operations. Hence, fault-tolerance quantum (FTQ) computing is seemingly an impossible task. It has, however, been shown that by using quantum error-correction, one can in theory perform arbitrarily long quantum computations given that the error rate of each operations is below some threshold [Sho96], [KLZ98]. In fact, the length of the computation by including said error-correction is only polylogarithmically longer than the original non-FTQ computation [NC10, p. 481]. By achieving FTQ computers, the potential of quantum computing can be fulfilled at last. The threshold for achieving such depends on several factors, particularly the quantum error-correcting codes used. For example, stabilizer codes can be concatenated to achieve FTQ computing with a threshold in the order of 10^{-5} [AGP08], which is an order of two from current error rates. Another popular approach to achieve FTQ computing is surface codes [Kit03], [DKLP02], which due to their clever local topological structure have a threshold around 10^{-2} [BCG$^+$24]. They, however, have a poor coding rate, thus are not suitable for large-scaling. Quantum LDPC codes overcome this hurdle partially by achieving similar threshold as surface codes but with significantly better coding rate [BCG$^+$24]. Despite its absolute cruciality in fulfilling the potential of quantum computing, FTQ computing is not discussed further in this thesis. It is henceforth merely assumed that quantum error-correction can be performed fault-tolerantly, that is without itself introducing additional errors.

Instead, the focus is on analysing codes that may be suitable for distributed quantum computing. In principle, any quantum error-correcting code can be used to encode the information that must be transmitted between quantum computers for distributed computing, however the class of entanglement-assisted (EA) error-correcting codes developed in [BDH06] is a great candidate for two reasons. Firstly, it is a generalisation of stabilizer codes in which the sender, Alice, and the receiver, Bob, are assumed to share some amount of EPR pairs, an assumption similar to that of many quantum communication protocols, e.g., quantum teleportation. It has been shown that the sharing of such EPR pairs can enable better performance compared to simply using quantum teleportation with a stabilizer coding scheme [BDH06, p. 2]. Secondly, an EA code can be constructed from any classical linear code, which greatly simplifies the construction process compared to designing stabilizer codes from classical dual-containing codes. Since the development of the framework of EA codes, a plethora of such codes have been defined, see for example [FLZ$^+$22], [AAH$^+$22], and [LB12]. An issue with EA codes is, however, that Bobs ebits are assumed to be noiseless, an assumption that is hard to realise in practice due to decoherence or the errors occurring when generating the EPR pairs. The effect of slackening this assumption has been addressed in [LB12], [FLZ$^+$22], and [FXDC23], where different procedures for handling the noisy ebits are considered. Particularly, in [LB12], it is analysed whether performing entanglement distillation before using an EA code is better than simply using a larger EA code on the noisy ebits over a depolarisation channel. Inspired by this work, the aim of this thesis is to analyse how decoherence on Bob's ebits affect his decoding performance. More specifically, since maintaining noiseless ebits on Bob's half requires substantial work, there is a trade-off between him maintaining noiseless ebits and the performance of the EA code used for error-correction. The aim is then to examine how this performance degrades. For example, if Alice and Bob share five EPR pairs, can Bob then still decode some information if only one of his ebits is corrupted by noise? And does this amount decrease if another ebit is corrupted? To the best of the author's knowledge, there is no published research on this area. A scenario describing the problem further is given for illustrative purposes.

1.6 Illustrative Scenario of the Problem

Although the following scenario may be too simplistic to directly implement in practice, it illustrates the problem at hand.

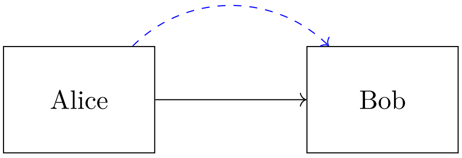


Figure 1.2: Block diagram of the quantum network for the scenario. The black arrow indicates a noisy quantum channel, while the blue indicates a channel capable of distributing ebits from an EPR pair.

Consider a quantum network consisting of two quantum computer called Alice and Bob that are connected by some noisy quantum channel as illustrated in Figure 1.2. Alice has some combinatorial optimisation problem to solve, say the travelling salesman problem, using quantum annealing [KN98]. Due to lack of computational power, Alice distributes some of the computing to Bob. Before doing so, Alice has through some other quantum algorithm obtained a region that is likely to contain the optimal solution. This information is encoded in an unknown quantum state. Naturally, Alice wants Bob to obtain this information such that Bob can take this into account in some manner, e.g., by using the state as initial state for quantum annealing or adding it into some penalty parameters. Since Alice only can communicate the state to Bob through the noisy quantum channel linking them, Alice has to encode the state with some quantum code. Assuming that Alice can generate EPR pairs as well as distributing half of the ebits to Bob, an EA code can be used. In practice, however, the EPR pairs cannot be maintained fully entangled, which is the underlying assumption in the EA coding framework. This can be overcome by using additional qubits to either encode the ebits with some additional quantum code or perform entanglement distillation. However, this would reduce the amount of qubits allocated for computations, hence the computational power. By slackening the assumption of the EA coding framework, Bob can allocate the additional qubits for computing the quantum annealing algorithm, however, at the cost of reducing the capability of correcting the noisy quantum state received by Alice. On the one hand, if the decoding is faulty, then Bob potentially uses a wrong optimal region, which generally would reduce the performance of the quantum annealing algorithm. On the other hand, the computational power gained by not allocating additional qubits for error-correction should increase the performance of the algorithm. This trade-off directly translates to keeping Bob's ebits noiseless by error-correction versus accuracy of the decoded quantum information received by Alice. Particularly, considering that Bob's ebits are prone to decoherence, to what extend would it be beneficial to use error-correction or entanglement distillation to overcome this hurdle considered that it should increase the decoding performance, however, at the cost of reducing the computational power of the system?

1.7 Problem Statement

Based on the preceding discussion, the problem statement for this thesis can be formulated:

How does noise on the receiver's ebits degrade the error-correcting capability of an entanglement-assisted quantum error-correcting code?

2 | Quantum Mechanics

In the beginning of the 20th century, classical mechanics was inadequate in describing phenomena at an atomic level, hence the field of quantum mechanics arose. Some of the properties occurring in quantum mechanics can be utilised in computing, which can revolutionise how computers work. A general, yet brief, mathematical introduction to quantum mechanics is therefore presented in this chapter. The focus is the properties that are of interest in quantum computing, however the use cases of the properties will be apparent in subsequent chapters. This chapter is based upon [NC10, ch. 2.2, 2.4], unless otherwise stated.

Quantum mechanics was evolved from observations that entities at the atomic scale, e.g., electrons and photons, behave both as waves and particles, a phenomenon known as the wave-particle duality. Classical waves are functions of time and space that over a given domain evolve continuously according to some wave equation, whereas a particle is localised in space, have quantised attributes, and moves according to classical mechanics. Hence, waves and particles are fundamentally different, so the wave-particle duality of quantum entities is quite paradoxical, yet verifiable by the famous double slit experiment [FLS13, ch. 1]. The field of quantum mechanics contains plenty of such counter-intuitive phenomena due to it describing physics at the atomic scale, which is unfamiliar to mankind in everyday-life. The paradoxical behaviour of quantum mechanics was even clear to the Nobel-winning physicist Richard Feynman, who said "*[...], I think I can safely say that nobody understands quantum mechanics.*" [Fey65, ch. 6], which should be understood as quantum mechanics behaves differently from what is experienced in everyday-life, hence it is hard to intuitively grasp, not that quantum mechanics cannot be used or even described mathematically. In fact, quantum mechanics can be described rigorously by basing it upon a few postulates that are presented in the following sections. The reader is encouraged to read the nomenclature if unfamiliar with the Dirac notation used in quantum mechanics to describe linear algebra. It should furthermore be noted that the term 'quantum' is occasionally omitted henceforth when it leads to no ambiguity.

2.1 State Spaces and States of Quantum Systems

The first postulate is about the states of entities which obey the quantum mechanics, more precisely the space in which quantum states occur.

> **Postulate 2.1: State Space**
> The state space of any isolated physical system is some complex Hilbert space, $\mathscr{H}$. The system is completely described by a unit vector in the state space, $|\psi\rangle$, known as its state vector. [NC10, p. 80]

Even though some of the theory presented in this chapter may apply to infinite-dimensional state spaces, all Hilbert spaces considered throughout this thesis are assumed to be finite-dimensional as these are used in quantum computing. It is furthermore generally omitted to state that they are complex.

The postulate deserves two remarks. Firstly, an isolated system is an idealised system that has no interaction with anything outside the system. Although isolated systems may be impossible to find in reality, only considering such systems simplifies the theory significantly. Secondly, the formulation of the postulate is intentionally vague and cannot be used to determine the state space, nor the state, of a particular system, only that it is some Hilbert space. A simple example is therefore in order.

> **Example 2.2: State Space for the Spin of an Electron**
> Consider the case of the spin of an electron, which only can take on two states, namely up, $|\uparrow\rangle$, and down, $|\downarrow\rangle$, as verified by the Stern-Gerlach experiment [NC10, ch. 1.5.1]. These two states, typically called the basis states, span the state space for the spin of an electron [NC10, p. 279], thus the associated state space is $\mathbb{C}^2$. Hence, every linear combination of the basis states is also a state of the system given that

it has unit norm. That is, an arbitrary state of the system, $|\psi\rangle$, can be written as

$$|\psi\rangle = \frac{\alpha\,|\uparrow\rangle + \beta\,|\downarrow\rangle}{\sqrt{|\alpha|^2 + |\beta|^2}}, \quad \alpha, \beta \in \mathbb{C}. \tag{2.1}$$

The scalars α, β are often referred to as amplitudes for the basis states $|\uparrow\rangle, |\downarrow\rangle$, respectively. Instead of explicitly including the normalisation factor for $|\psi\rangle$ in (2.1), the state is often written with the following equivalent normalisation constraint:

$$|\psi\rangle = \alpha\,|\uparrow\rangle + \beta\,|\downarrow\rangle\,, \quad |\alpha|^2 + |\beta|^2 = 1, \tag{2.2}$$

where it is implicit that $\alpha, \beta \in \mathbb{C}$. This is the convention used throughout this thesis.

Not only does Example 2.2 provide an example of the state space for a particular system, it also provides significant information about the states of the system. As explicitly written in (2.2), the spin of an electron is not only limited to be either up or down, but any 'unit' combination hereof, e.g., $|\leftarrow\rangle$, or $|\nearrow\rangle$. Said differently, the spin can be in both $|\uparrow\rangle$ and $|\downarrow\rangle$ simultaneously, a property known as superposition. This property is not only limited to the spin of an electron, but holds true for any quantum system due to state spaces being Hilbert spaces. Despite the mathematical simplicity of superposition, its implication is counter-intuitive as it does not occur in classical mechanics, e.g., a table cannot have several positions simultaneously. This is popularly illustrated by Schrödinger's cat [NC10, p. 387].

In reality, many systems of interest are more complex than that of a single particle, however, these can in general be constructed from several simple ones as described in the following postulate.

Postulate 2.3: State Space of Composite Systems
Let n systems have state spaces $\mathscr{H}_1, \ldots, \mathscr{H}_n$, respectively. The state space of the composition of these spaces, $\mathscr{H}$, is then the tensor product of the state spaces of the n subsystems. That is,

$$\mathscr{H} = \bigotimes_{i=1}^{n} \mathscr{H}_i.$$

[NC10, p. 94]

Elementary details regarding the tensor product and the closely related Kronecker product can be found in Section B.1. Particularly, it contains proofs of several properties of the Kronecker product that are used throughout this thesis.

By comparing the state space of composite quantum system to the classical case that instead use the Cartesian product to describe its composite state space and state, one would expect that the state of a quantum system is given by the tensor product of the individual states. By continuing on Example 2.2, it can be seen whether this expectation holds true.

Example 2.4: State Space for the Spin of Two Electrons
Consider now the case of the spin of two electrons. Assume that Alice and Bob each possess an electron with state spaces $\mathscr{H}_A$ and $\mathscr{H}_B$, respectively. They now want to know the spin of their electrons as a composite system, which cf. Postulate 2.3 is $\mathscr{H}_A \otimes \mathscr{H}_B$, henceforth denoted $\mathscr{H}_{AB}$. Since $\{|\uparrow\rangle, |\downarrow\rangle\}$ constitutes a basis for both $\mathscr{H}_A$ and $\mathscr{H}_B$, then $\{|\uparrow\rangle \otimes |\uparrow\rangle, |\uparrow\rangle \otimes |\downarrow\rangle, |\downarrow\rangle \otimes |\uparrow\rangle, |\downarrow\rangle \otimes |\downarrow\rangle\}$ is a basis for $\mathscr{H}_{AB}$ (see for example [DF04, p. 421]). This basis is henceforth denoted $\{|\uparrow\uparrow\rangle, |\downarrow\uparrow\rangle, |\downarrow\uparrow\rangle, |\downarrow\downarrow\rangle\}$, a convention commonly used throughout the thesis. This implies that an arbitrary state in the composite system has the form

$$|\psi\rangle_{AB} = \varepsilon_{\uparrow\uparrow}\,|\uparrow\uparrow\rangle + \varepsilon_{\uparrow\downarrow}\,|\uparrow\downarrow\rangle + \varepsilon_{\downarrow\uparrow}\,|\downarrow\uparrow\rangle + \varepsilon_{\downarrow\downarrow}\,|\downarrow\downarrow\rangle\,, \quad \sum_{i,j\in\{\uparrow,\downarrow\}} |\varepsilon_{ij}|^2 = 1. \tag{2.3}$$

To see whether it is reasonable to expect that the state of the composite system is the tensor product of states, the states of the subsystems must be defined. Thus, let the states of the spin of the electrons when seen as separate system be

$$|\psi\rangle_A = \alpha_\uparrow\,|\uparrow\rangle + \alpha_\downarrow\,|\downarrow\rangle\,, \quad |\psi\rangle_B = \beta_\uparrow\,|\uparrow\rangle + \beta_\downarrow\,|\downarrow\rangle\,, \quad |\alpha_\uparrow|^2 + |\alpha_\downarrow|^2 = |\beta_\uparrow|^2 + |\beta_\downarrow|^2 = 1.$$

If the state of the composite system is the tensor product of the two states, it must have the form

$$|\psi\rangle_A \otimes |\psi\rangle_B = \alpha_\uparrow\beta_\uparrow |\uparrow\uparrow\rangle + \alpha_\uparrow\beta_\downarrow |\uparrow\downarrow\rangle + \alpha_\downarrow\beta_\uparrow |\downarrow\uparrow\rangle + \alpha_\downarrow\beta_\downarrow |\downarrow\downarrow\rangle \,. \tag{2.4}$$

Notice that this state is immediately normalised due to the normalisation constraints of the subsystems.

Thus, to see whether the state of the composite system is the tensor product of states, the amplitudes of the state given in (2.4) must be compared to those of (2.4). To do so, there are two cases to consider. The composite system is either such that the two electrons do not interact by any means or such that they have some dependency.

In the first case, with no dependency, the intuitive method to determine the amplitudes for the composite state would be to multiply the corresponding amplitudes for the separate states, e.g., $\varepsilon_{\uparrow\uparrow} = \alpha_\uparrow\beta_\uparrow$. By doing so, one see that the state of the composite system in (2.4) is exactly the one obtained by taking the tensor product of the individual states as done in (2.4). Thus, when there is no dependency in the composite system, the expectation of the state being the tensor product holds true.

In the second case, with dependency, assume that the states are related in a way such that the spin of the electrons always is in the same direction; always up, always down, or any superposition thereof. Any state of this system thus has the form

$$|\varphi\rangle_{AB} = \zeta_\uparrow |\uparrow\uparrow\rangle + \zeta_\downarrow |\downarrow\downarrow\rangle \,, \quad |\zeta_\uparrow|^2 + |\zeta_\downarrow|^2 = 1.$$

To see whether this is a tensor product, the amplitudes of this state are related to those in (2.4), which is equivalent to solving the system

$$\alpha_\uparrow\beta_\uparrow = \zeta_\uparrow, \quad \alpha_\uparrow\beta_\downarrow = \alpha_\downarrow\beta_\uparrow = 0, \quad \alpha_\downarrow\beta_\downarrow = \zeta_\downarrow, \quad |\zeta_\uparrow|^2 + |\zeta_\downarrow|^2 = 1,$$

which only has a solution when $|\zeta_\uparrow| = 0$ and $|\zeta_\downarrow| = 1$ or vice versa. Hence, the tensor product of states does generally not provide the correct state of the composite system. Naturally, the state $|\varphi\rangle$ naturally still has the form of (2.4), namely with coefficients

$$\varepsilon_{\uparrow\uparrow} = \zeta_\uparrow, \quad \varepsilon_{\uparrow\downarrow} = \varepsilon_{\downarrow\uparrow} = 0, \quad \varepsilon_{\downarrow\downarrow} = \zeta_\downarrow.$$

In other words, $|\varphi\rangle_{AB}$ cannot be written as the tensor product of states of the subsystem, however, it can be written as a linear combination of tensor products of the basis states of the subsystems.

In Example 2.4, it was shown that any state of a composite system can be written as a linear combination of tensor products of basis states of the subsystems, however generally not as a tensor product of the states of the subsystems. In cases where the latter cannot be done, the state of the composite system is said to be entangled, an extremely important phenomenon in quantum mechanics. It therefore deserves a more rigorous definition.

Definition 2.5: Separable States and Entangled States
Let n quantum systems have state spaces $\mathscr{H}_1, \ldots, \mathscr{H}_n$ with fixed bases $\{|i\rangle_1\}_i, \ldots, \{|j\rangle_n\}_j$, respectively. Furthermore, let $\mathscr{H}$ denote the composite system of the n subsystems. An arbitrary state of the composite system has the form

$$|\psi\rangle = \sum_i \cdots \sum_j \varepsilon_{i\cdots j} |i\cdots j\rangle \,, \quad \sum_i \cdots \sum_j |\varepsilon_{i\cdots j}|^2 = 1.$$

If there exists scalars $\varepsilon_{i_1}, \ldots, \varepsilon_{j_n}$ such that $\varepsilon_{i\ldots j} = \varepsilon_{i_1} \cdots \varepsilon_{j_n}$, for all $i, \ldots, j$, then $|\psi\rangle$ is said to be an separable state, and the k-th subsystem is in state

$$|\varphi\rangle_k = \sum_i \varepsilon_{i_k} |i\rangle_k \,.$$

Otherwise, the state is said to be entangled. [NC10, pp. 95-96]

It should be noted that if one considers an entangled state, it is impossible to determine a state for each of the subsystems, thus, the entire composite system is needed to be considered when dealing with entanglement.

Relating the rigorous definition of entangled states to Example 2.4, it easily follows that the composite system with no dependency is separable as the amplitudes of the state in the composite system are products of amplitudes of the corresponding states in the subsystems, e.g., $\varepsilon_{\uparrow\uparrow} = \alpha_{\uparrow}\beta_{\uparrow}$. The composite system with dependency is nonetheless entangled in non-trivial cases, as not all the amplitudes in the composite system are a product of amplitudes in the subsystems. Thus, composite systems that has some dependency is said to be entangled. If the amplitudes of the entangled state in Example 2.4 are chosen appropriately, the state is actually one of the so-called EPR pairs, also known as Bell states, which typically are used to illustrate entanglement. These pairs are now defined in the framework of spin of electrons.

Definition 2.6: EPR Pairs
Consider a composite quantum system consisting of two two-level subsystems with basis states $|{\uparrow}\rangle$ and $|{\downarrow}\rangle$. The following four states of the composite system are called EPR pairs:

$$\begin{aligned} |\Phi^{+}\rangle &= \frac{1}{\sqrt{2}}(|{\uparrow\uparrow}\rangle + |{\downarrow\downarrow}\rangle), \quad &|\Phi^{-}\rangle &= \frac{1}{\sqrt{2}}(|{\uparrow\uparrow}\rangle - |{\downarrow\downarrow}\rangle), \\ |\Psi^{+}\rangle &= \frac{1}{\sqrt{2}}(|{\uparrow\downarrow}\rangle + |{\downarrow\uparrow}\rangle), \quad &|\Psi^{-}\rangle &= \frac{1}{\sqrt{2}}(|{\uparrow\downarrow}\rangle - |{\downarrow\uparrow}\rangle). \end{aligned}$$

[NC10, p. 25]

The EPR pairs are simply defined for now, however, their importance will become apparent as the theory of quantum mechanics is developed throughout this chapter.

The arena for quantum mechanics has now been introduced and it is time to examine how it behaves.

2.2 Evolution of Quantum Systems

The behaviour of a quantum systems is essentially described by a single equation, the Schrödinger equation.

Postulate 2.7: Time Evolution of States
The time evolution of a state, $|\psi\rangle \in \mathscr{H}$, of an isolated system satisfies the Schrödinger equation

$$\mathrm{i}\hbar \frac{\mathrm{d}}{\mathrm{d}t} |\psi(t)\rangle = \hat{H}(t) |\psi(t)\rangle ,$$

where i is the imaginary unit, $\hbar$ is the reduced Planck constant, and $\hat{H}(t) \in \mathrm{End}(\mathscr{H})$ is a Hermitian operator known as the Hamiltonian of the system. [NC10, p. 82]

Once again, the postulate is vaguely formulated as it does not provide information about the Hamiltonian or how to find it, only that it must be used to characterise the evolution of the system. In order to consider the evolution in a general sense, that is without referring to a specific system with a given Hamiltonian, one only needs to know that the Hamiltonian is some Hermitian operator, which is exactly the framework considered in this thesis. In that case, it is possible to determine a general solution to Schrödinger equation. In the simplest case where the Hamiltonian is time-independent, the state of the system at any time t is

$$|\psi(t)\rangle = \exp\left\{ \frac{-\mathrm{i}\hat{H}}{\hbar} t \right\} |\psi(0)\rangle , \tag{2.5}$$

where $|\psi(0)\rangle$ is the initial state of the system at time $t = 0$. Since the operator $\exp\left\{\frac{-\mathrm{i}\hat{H}}{\hbar}t\right\}$ determines the behaviour of the system, it would be desirable to determine how it affects the amplitudes of the state of the system, particularly whether it preserves inner products. In other words, is it unitary? To test this, it is helpful to realise that since $\hat{H}$ is Hermitian by Postulate 2.7, it is also normal. This has the implication that $\exp\{\alpha H\}\exp\{\beta H^\dagger\} = \exp\{\alpha H + \beta H^\dagger\}$ for $\alpha, \beta \in \mathbb{C}$ cf. Theorem B.6. By utilising this, it follows that

$$\exp\left\{\frac{-\mathrm{i}\hat{H}}{\hbar}t\right\}\left(\exp\left\{\frac{-\mathrm{i}\hat{H}}{\hbar}t\right\}\right)^\dagger = \exp\left\{\frac{-\mathrm{i}\hat{H}}{\hbar}t\right\}\exp\left\{\frac{\mathrm{i}\hat{H}^\dagger}{\hbar}t\right\} = \exp\left\{\frac{\mathrm{i}(\hat{H}^\dagger - \hat{H})}{\hbar}t\right\} = \exp\{0\} = I,$$

where the penultimate equality follows from $\hat{H}$ being Hermitian. Thus, the operator $\exp\left\{\frac{\mathrm{i}\hat{H}}{\hbar}t\right\}$ is unitary. In other words, the postulate of having an time-independent Hermitian Hamiltonian is equivalent to postulating that the evolution of a system is described by a unitary operator. Since this new formulation does not require a differential equation, it is in some sense simpler, hence it is the one used in this thesis. It therefore deserves its own rigorous description.

Postulate 2.8: Discrete Evolution of States
The state of an isolated system with state space $\mathscr{H}$ at time t, $|\psi(t)\rangle$, is described by a unitary operator, $U \in \mathrm{End}(\mathscr{H})$, that acts on the initial state of the system, $|\psi(0)\rangle$, i.e.,

$$|\psi(t)\rangle = U(t)\,|\psi(0)\rangle\,.$$

[NC10, p. 81]

Two remarks regarding the postulate is in order. Firstly, since the evolution of a system is unitary, it is reversible. Secondly, since this merely is a reformulation of Postulate 2.7, neither does this provide insight on how to determine the unitary operator defining the evolution of the system. However, this also implies that any unitary operator is suitable for some system, at least in theory. The framework for the evolution of systems used in this framework is therefore that it is given by some unitary operator, similarly to how the Hamiltonian simply was considered some Hermitian operator.

Some of the most important unitary operators in quantum mechanics are the Pauli matrices due to constituting a basis for operators in $\mathrm{End}(\mathbb{C}^2)$ cf. Theorem B.7. Since they will be used numerously throughout this thesis, they are now defined.

Definition 2.9: Pauli Matrices
The following three matrices in $\mathrm{End}(\mathbb{C}^2)$ are known as the Pauli matrices:

$$X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \quad Y = \begin{bmatrix} 0 & -\mathrm{i} \\ \mathrm{i} & 0 \end{bmatrix}, \quad Z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}.$$

Occasionally, the set of Pauli matrices also include $I \in \mathrm{End}(\mathbb{C}^2)$. [NC10, p. 65]

The Pauli matrices are examined a bit further in Section B.3. It is for example proven that they are unitary and thereby suitable to describe the evolution of some system. Such systems are now examined.

Example 2.10: Evolution by Pauli Matrices
Consider a simple classical two-level system, e.g., the side of a coin after a flip. The state of the system can be described by a bit.

Consider a simple classical two-level system whose state can be described by a bit and evolves accordingly to some Pauli matrix. Although this is not a quantum system, it can be considered as a simple one in which superposition is impossible. Let the two states be represented as the canonical basis vectors $e_1, e_2 \in \mathbb{C}^2$ denoted as $|0\rangle, |1\rangle$, respectively (see Section 3.1 for further discussion of the notation).

If the initial state of the system is $|0\rangle$, the system evolves accordingly to one of the following three equations:

$$X\,|0\rangle = |1\rangle\,, \quad Y\,|0\rangle = \mathrm{i}\,|1\rangle\,, \quad Z\,|0\rangle = |0\rangle\,.$$

Similarly, if the initial state is $|1\rangle$, it evolves as one of

$$X\,|1\rangle = |0\rangle\,, \quad Y\,|1\rangle = -\mathrm{i}\,|0\rangle\,, \quad Z\,|1\rangle = -\,|1\rangle\,.$$

> By comparing the two sets of equations, it can be seen that X flips the bit, Y flips the bit, multiplies with i, and flips the sign of one of the bits, while Z flips the sign of one of the bits. The operators are therefore also commonly referred to as the bit flip operator, the bit-phase flip operator, and the phase flip operator, respectively.

Now that three examples of evolution of quantum states has been considered, Postulate 2.8 deserves another remark. It only holds for isolated systems, and since no system can be perfectly isolated, except the Universe as a whole, the postulate can be seen as an ideal version of how systems actually evolve. In practice, any system is to some extent connected with the environment in which case the system evolves differently than expected, that is, it may not evolve according to a unitary operator, thus it may not be irreversible. The effect is that the superposition property of states may be lost to the environment such that the state of the system eventually is one of its basis states. Said differently, the theory of quantum mechanics presented hitherto may case to hold, and classical mechanics is sufficient to describe the system. This process is known as decoherence and is further discussed in Section 3.3. An almost paradoxical scenario in which a system cannot be isolated is when a measurement device is connected to a system in order to measure its state. Measuring a system is an absolute necessity in order to obtain information about the system, however, it seemingly cannot be done without potentially ruining quantum mechanical behaviour such as superposition and unitary evolution. This inconvenience is now examined.

2.3 Measurements of Quantum Systems

The outcome of performing a measurement of a system as well as the effect it has on the system is now postulated.

> **Postulate 2.11: Measurement of States**
> A measurement of a state in $\mathscr{H}$ is described by a collection of operators, $\{M_m\}_m \in \mathrm{End}(\mathscr{H})$, that satisfies the completeness equation
>
> $$\sum_m M_m^\dagger M_m = I.$$
>
> The operators are known as observables and the index m refers to possible measurement outcomes.
>
> If the state of the system before the measurement is $|\psi\rangle$, then the probability of measuring outcome m is
>
> $$\Pr(m) = \langle\psi|M_m^\dagger M_m|\psi\rangle\,, \tag{2.6}$$
>
> and the state after the measurement is
>
> $$|\psi'\rangle = \frac{M_m\,|\psi\rangle}{\sqrt{\langle\psi|M_m^\dagger M_m|\psi\rangle}}.$$
>
> [NC10, pp. 84-85]

Before discussing the postulate, it is firstly shown that $\Pr(m)$ in (2.6) actually is a probability measure. Since $\{m\}_m$ describes the possible outcomes of the measurement, showing that the probability of the sample space is one is equivalent to showing that $\sum_m \Pr(m) = 1$:

$$\sum_m \Pr(m) = \sum_m \langle\psi|M_m^\dagger M_m|\psi\rangle = \langle\psi|\left(\sum_m M_m^\dagger M_m\right)|\psi\rangle \overset{(a)}{=} \langle\psi|\psi\rangle = 1, \tag{2.7}$$

where (a) follows from the completeness equation.

To show that $\Pr(m) \in [0,1]$, it is firstly shown that $\Pr(m) \in \mathbb{R}$. It is trivial to see that $M_m^\dagger M_m$ is Hermitian, which implies that for any $|\psi\rangle$ in the state space, then

$$(\Pr(m))^* = \left(\langle\psi|M_m^\dagger M_m|\psi\rangle\right)^* = (\langle\psi|M_m^\dagger M_m|\psi\rangle) = \Pr(m),$$

where $\Pr(m)^*$ denotes the complex conjugate of $\Pr(m)$. This implies that $\Pr(m) \in \mathbb{R}$. The non-negativity of $\Pr(m)$ follows from

$$\Pr(m) = \langle\psi|M_m^\dagger M_m|\psi\rangle = \|M_m\,|\psi\rangle\|^2 \geq 0.$$

Thus, $M_m^\dagger M_m$ is in fact positive semi-definite. By combining this with (2.7), it follows that $\Pr(m) \in [0,1]$.

The last probability axiom, the probability of a finite union of disjoint events corresponds to a summation of probabilities, follows from the linearity of inner products and the fact that a Hermitian matrix can be decomposed into a sum of Hermitian matrices. That is, for $\{m\}_m$ describing disjoint events, then

$$\Pr\left(\bigcup_m m\right) = \sum_{m,n} \langle\psi|M_m^\dagger M_n|\psi\rangle \overset{(a)}{=} \sum_{m,n} \langle\psi|M_m^\dagger M_n|\psi\rangle\, \delta_{mn} = \sum_m \Pr(m),$$

where (a) follows from the events being disjoint when $m \neq n$. This concludes that $\Pr(m)$ is a probability measure.

At this point, some remarks about the observables are in order. The postulate does not provide any insight how to choose observables, but any set of operators satisfying the completeness equation can theoretically be chosen. No matter the properties of the observables, they are dependent on the system and how the measurement is performed, e.g., the possible outcomes of an the spin of an electron is up or down, hence there should be two observables, one for each direction. The observables are often chosen to be a set of orthogonal projections on the state space, in which case the measurement is said to be a projective measurement. To understand why, consider a quantum system A with state space $\mathscr{H}_A$ that is to be measured with observables $\{M_m\}_m$. Consider furthermore an additional system, B, chosen such that its state space $\mathscr{H}_B$ has dimension equal to the number of observables for system A. Denote an orthonormal basis of $\mathscr{H}_B$ as $\{|m\rangle\}_m$ such that there is a bijection between the basis states and the possible outcomes of measurement of A. Assume now that their composite system AB with state space $\mathscr{H}_{AB} = \mathscr{H}_A \otimes \mathscr{H}_B$ is isolated. Then for a fixed state $|\varphi\rangle \in \mathscr{H}_B$, define an operator $T \in \mathrm{End}(\mathscr{H}_{AB})$ such that

$$T(|\psi\rangle_A\,|\varphi\rangle_B) = \sum_m (M_m \otimes I_B)(|\psi\rangle_A \otimes |m\rangle_B), \quad |\psi\rangle \in \mathscr{H}_A.$$

It then follows that

$$\begin{aligned}(\langle\psi|_A\,\langle\varphi|_B)T^\dagger T(|\psi'\rangle_A\,|\varphi\rangle_B) &= \sum_{m,n} (\langle\psi|_A \otimes \langle m|_B)(M_m^\dagger \otimes I_B)(M_n \otimes I_B)(|\psi'\rangle_A \otimes |n\rangle_B) \\ &= \sum_{m,n} \langle\psi|M_m^\dagger M_n|\psi'\rangle\,\langle m|n\rangle \overset{(a)}{=} \sum_m \langle\psi|M_m^\dagger M_m|\psi'\rangle \overset{(b)}{=} \langle\psi|\psi'\rangle\,,\end{aligned}$$

where (a) follows from $\{|m\rangle\}_m$ being an orthonormal basis and (b) from the observables $\{M_m\}_m$ satisfying the completeness equation. This implies that T preserves inner products of states in the 1-dimensional subspace spanned by $|\psi\rangle_A\,|\varphi\rangle_B$. However, since AB is assumed to be an isolated system, T is not a valid operator to describe the evolution of the system cf. Postulate 2.8 as it is not unitary on $\mathscr{H}_{AB}$. Nonetheless, it can easily be extended to an unitary operator such that it describes some evolution of the system. To do so, let the subspace spanned by $|\psi\rangle_A\,|\varphi\rangle_B$ be denoted W. Then, let $W, W^\perp, \mathrm{Im}\{T\}, \mathrm{Im}\{T\}^\perp$ have orthonormal bases $\{|w_i\rangle\}_i, \{|w_j'\rangle\}_j, \{|v_i\rangle\}_i$, and $\{|v_j'\rangle\}_j$, respectively, where $W^\perp$ denotes the orthogonal complement to W in $\mathscr{H}_{AB}$ and $\mathrm{Im}\{T\}$ the image of T. One example of such an unitary extension of T to $\mathscr{H}_{AB}$ is

$$U = \sum_i |v_i\rangle\,\langle w_i| + \sum_j |v_j'\rangle\,\langle w_j'|\,. \tag{2.8}$$

The fact that it is an extension of T follows since for all $|w\rangle \in W$, it holds that

$$U\,|w\rangle = \left(\sum_i |v_i\rangle\,\langle w_i| + \sum_j |v_j'\rangle\,\langle w_j'|\right)|w\rangle = \sum_i |v_i\rangle\,\langle w_i|w\rangle = \sum_i T\,|w_i\rangle\,\langle w_i|w\rangle = T\,|w\rangle\,.$$

The unitarity of U also follows easily. Thus, U defined in (2.8) is a valid operator describing a possible evolution of AB. After having applied U to the composite system, another set of observables $\{P_m\}_m$

defined as $P_m = I_A \otimes |m\rangle\langle m|$ is used for measurement. It is trivial to see that they satisfy the completeness equation. It can furthermore be seen that

$$P_mP_n = (I_A \otimes |m\rangle\langle m|)(I_A \otimes |n\rangle\langle n|) = I_A \otimes (|m\rangle\langle m|n\rangle\langle n|) = \begin{cases} P_m, & \text{if } m = n, \\ 0, & \text{if } m \neq n. \end{cases}$$

The set $\{P_m\}_m$ therefore also constitute a set of orthogonal projections on $\mathscr{H}_{AB}$, hence they define a projective measurement on system B. Using $\{P_m\}_m$ as observables after applying U to the state $|\psi\rangle_A|\varphi\rangle_B$, the probability of measuring outcome m is given by

$$\begin{aligned}\Pr(m) &= (\langle\psi|_A\langle\varphi|_B)U^\dagger P_m^\dagger P_m U(|\psi\rangle_A|\varphi\rangle_B) \\ &= \sum_{i,j}(\langle\psi|_A\langle i|_B)(M_i^\dagger \otimes I_B)(I_A \otimes |m\rangle\langle m|)(M_j \otimes I_B)(|\psi\rangle_A|j\rangle_B) \\ &= \sum_{i,j}\langle\psi|M_i^\dagger M_j|\psi\rangle\langle i|m\rangle\langle m|j\rangle = \langle\psi|M_m^\dagger M_m|\psi\rangle,\end{aligned}$$

which is equivalent to that of Postulate 2.11. The state after the projective measurement is

$$\begin{aligned}\frac{P_mU(|\psi\rangle_A|\varphi\rangle_B)}{\sqrt{\Pr(m)}} &= \sum_n \frac{(I_A \otimes |m\rangle\langle m|)(M_n \otimes I_B)(|\psi\rangle_A|n\rangle_B)}{\langle\psi|M_m^\dagger M_m|\psi\rangle} = \sum_n \frac{(M_n|\psi\rangle) \otimes (|m\rangle\langle m|n\rangle)}{\langle\psi|M_m^\dagger M_m|\psi\rangle} \\ &= \frac{(M_m|\psi\rangle) \otimes |m\rangle}{\langle\psi|M_m^\dagger M_m|\psi\rangle}.\end{aligned}$$

If one then only considers the state in subsystem A, this yields $(M_m|\psi\rangle)/(\langle\psi|M_m^\dagger M_m|\psi\rangle)$, which also is equivalent to that of Postulate 2.11. Thus, performing a measurement of a system with observables $\{M_m\}$ is equivalent to performing a projective measurement combined with the ability to introduce an additional system and evolving the composite system unitarily. In other words, measurements described in Postulate 2.11 are equivalent to projective measurements in isolated systems. Although such projective measurements therefore does not represent all scenarios in practice, they are often used in the literature, hence also in this thesis.

Instead of a defining a set of observables $\{M_m\}_m$, a projective measurement only requires one Hermitian observable M. Being Hermitian, H has spectral decomposition $M = \sum_m mP_m$, where P_m is the projection onto the m-eigenspace of M, i.e., the eigenspace associated with eigenvalue m. In this case, $\{m\}_m$ still refer to the possible outcomes. Thus, for a projective measurement, it is sufficient to define orthogonal projectors $\{P_m\}_m$ satisfying $\sum_m P_m = I$. If $\{|m\rangle\}_m$ is an orthonormal basis of the state space, then using $\{P_m\}_m$ defined as $P_m = |m\rangle\langle m|$ is commonly referred to as measuring in the $\{|m\rangle\}_m$ basis. A simple example of a projective measurement now follows.

Example 2.12: Projective Measurements
Consider performing a projective measurement of the spin of an electron. In that case, the state space is $\mathbb{C}^2$ in which $\{|\uparrow\rangle, |\downarrow\rangle$ is an orthonormal basis. Thus, a projective measurement can be defined by

$$P_\uparrow = |\uparrow\rangle\langle\uparrow|, \quad P_\downarrow = |\downarrow\rangle\langle\downarrow|,$$

which corresponds to measuring in the $\{|\uparrow\rangle, |\downarrow\rangle\}$ basis.

Let the state before measurement be $|\psi\rangle = \alpha|\uparrow\rangle + \beta|\downarrow\rangle$. The probability of measuring $|\uparrow\rangle$ is then

$$\Pr(\uparrow) = \langle\psi|P_\uparrow|\psi\rangle = (\alpha^*\langle\uparrow| + \beta^*\langle\downarrow|)|\uparrow\rangle\langle\uparrow|(\alpha|\uparrow\rangle + \beta|\downarrow\rangle) = |\alpha|^2\langle\uparrow|\uparrow\rangle\langle\uparrow|\uparrow\rangle = |\alpha|^2,$$

and the state after measurement is

$$|\psi'\rangle = \frac{P_\uparrow|\psi\rangle}{\sqrt{\Pr(\uparrow)}} = \frac{|\uparrow\rangle\langle\uparrow|(\alpha|\uparrow\rangle + \beta|\downarrow\rangle)}{|\alpha|} = \frac{\alpha}{|\alpha|}|\uparrow\rangle.$$

It follows analogously that

$$\Pr(\downarrow) = |\beta|^2, \quad |\psi'\rangle = \frac{\beta}{|\beta|}|\downarrow\rangle.$$

Thus, the state after the measurement is the normalised projection onto the eigenspace of $P_\uparrow$ or $P_\downarrow$ depending on the outcome. This is also referred to as the collapse of the wave-function as it no longer exists in a superposition but a definite state.

From the example, it can once again be verified that $|\alpha|^2 + |\beta|^2 = 1$ is a necessary condition as they correspond to probabilities of measuring a system in a given state, a result commonly known as Born's rule. This gives another counter-intuitive phenomenon in quantum mechanics, namely that even when measuring a system, the state of it is only known probabilistically, something not seen in classical mechanics. In the special case where the state of the system before doing a measurement is in state $|\uparrow\rangle$ ($|\downarrow\rangle$), then $\alpha = 1$ ($\beta = 1$), hence one would with certainty know that state of the system. However, it is actually only possible to achieve such certainty of the state if the possible outcomes are orthonormal.

> **Theorem 2.13: Measurement of Non-orthonormal States**
> Two non-orthonormal states $|\psi_0\rangle, |\psi_1\rangle$ cannot be distinguished with certainty by any measurement.
> [NC10, p. 87]

Proof
The result is shown by contradiction, hence assume that a measurement of some system can distinguish two non-orthonormal states $|\psi_0\rangle, |\psi_1\rangle$ with certainty. That is, assume the states $|\psi_0\rangle, |\psi_1\rangle$ have observables M_0, M_1 that satisfy the completeness equation and

$$\langle\psi_0| M_0^\dagger M_0 |\psi_0\rangle = 1, \quad \langle\psi_1| M_1^\dagger M_1 |\psi_1\rangle = 1.$$

There are now two cases to consider. The state before measurement is either $|\psi_0\rangle$ or $|\psi_1\rangle$.

If the state before measurement is $|\psi_1\rangle$, then

$$1 = \sum_{m\in\{0,1\}} \Pr(m) = \sum_{m\in\{0,1\}} \langle\psi_1|M_m^\dagger M_m|\psi_1\rangle \implies \langle\psi_1|M_0^\dagger M_0|\psi_1\rangle = 0 \implies M_0 |\psi_1\rangle = 0, \tag{2.9}$$

where the first implication follows from the assumptions.

Before considering the case where the state before measurement is $|\psi_0\rangle$, it it noted that due to $|\psi_0\rangle, |\psi_1\rangle$ being non-orthonormal, it is possible to decompose $|\psi_0\rangle$ as

$$|\psi_0\rangle = \alpha |\psi_1\rangle + \beta |\varphi\rangle, \quad \text{s.t.} \quad |\alpha|^2 + |\beta|^2 = 1, |\beta| < 1,$$

for $|\varphi\rangle$ being orthonormal to $|\psi_1\rangle$. This implies that given the state before measurement is $|\psi_0\rangle$, then

$$\begin{aligned}\Pr(0) = \langle\psi_0|M_0^\dagger M_0|\psi_0\rangle &= (\alpha^* \langle\psi_1| + \beta^* \langle\varphi|)M_0^\dagger M_0(\alpha |\psi_1\rangle + \beta |\varphi\rangle) \overset{(a)}{=} |\beta|^2 \langle\varphi|M_0^\dagger M_0|\varphi\rangle \\ &\leq |\beta|^2 \sum_{m\in\{0,1\}} \langle\varphi|M_m^\dagger M_m|\varphi\rangle \overset{(b)}{=} |\beta|^2 \langle\varphi|\varphi\rangle = |\beta|^2 < 1,\end{aligned}$$

where (a) follows from (2.9) and (b) from the observables satisfying the completeness equation. This last equation is a contradiction to the assumptions. ■

Another important result about measurements is regarding rotations of scales. Assume some system has observable M. For $\theta \in \mathbb{R}$, it then follows that

$$\langle\psi| \mathrm{e}^{-\mathrm{i}\theta} M^\dagger M \mathrm{e}^{\mathrm{i}\theta} |\psi\rangle = \langle\psi| M^\dagger M |\psi\rangle.$$

Hence, the state $\mathrm{e}^{\mathrm{i}\theta} |\psi\rangle$ cannot be distinguished from $|\psi\rangle$ using measurements, thus the states are said to be equivalent up to some phase θ, commonly referred to as a global phase. This implies that the systems which are equivalent up to a global phase represent the same quantum system, i.e., that θ does not possess any physical meaning in the sense that it can be chosen arbitrarily for mathematical convenience. With this in mind, one could reformulate Postulate 2.1 to define states as rays in a projective Hilbert space and by taking care of the normalisation at some point such that the probabilities when performing a measurement sum to one. For this thesis, however, states are simply unit vectors that are equivalent up to a global phase.

The general concepts of quantum mechanics required for quantum computing have now been presented using states as unit vectors in some finite-dimensional Hilbert space. However, another formulation is often more mathematically convenient in quantum information theory, hence it is now presented.

2.4 Density Operator Formalism

Recall Example 2.4 in which Alice and Bob each possess an electron which together form a composite system of two two-level systems. In the case where the electrons are entangled, the state of the composite system has the form

$$|\varphi\rangle_{AB} = \zeta_\uparrow |\uparrow\uparrow\rangle + \zeta_\downarrow |\downarrow\downarrow\rangle), \quad |\zeta_\uparrow|^2 + |\zeta_\downarrow|^2 = 1. \tag{2.10}$$

Since the state is entangled, it is impossible for Alice or Bob to describe the state of their electron without also considering the other's electron, which is problematic. This is easier seen if there are, say, a million entangled electrons. Then Alice (Bob) would have to consider all of these in order to describe anything about their own electron. In order to provide insights to such subsystems, quantum states must be formulated differently than unit vectors in the state space, which leads to the following reformulation of Postulate 2.1.

Postulate 2.14: State Space
The state space of any isolated physical system is some complex Hilbert space, $\mathscr{H}$. The system is completely described by its density operator, $\rho \in \operatorname{End}(\mathscr{H})$, which is a positive semi-definite operator with trace one that acts on the state space. [NC10, p. 102]

A new formulation of quantum states is now possible, however it is yet to be shown how to determine density operators. Since such scenarios often occur in practice, density operators are often constructed from an ensemble of states, which is a set of ordered pairs $\{(p_i, |\psi_i\rangle)\}_i$ indicating that the state of the system is $|\psi_i\rangle$ with probability p_i. In that case, the density operator is given as

$$\rho = \sum_i p_i |\psi_i\rangle \langle\psi_i| . \tag{2.11}$$

Although the entangled state given in Example 2.4, written again in (2.3), is not written as an ensemble, it can easily be formulated as such. By Born's rule, the probability of measuring $|\uparrow\uparrow\rangle$ is $|\zeta_\uparrow|^2$, while measuring $|\downarrow\downarrow\rangle$ happens with probability $|\zeta_\downarrow|^2$. Hence, $|\varphi\rangle_{AB}$ can similarly be described by the ensemble $\{(|\zeta_\uparrow|^2, |\uparrow\uparrow\rangle), (|\zeta_\downarrow|^2, |\downarrow\downarrow\rangle)\}$, hence its density operator description is

$$\rho = |\zeta_\uparrow|^2 |\uparrow\uparrow\rangle \langle\uparrow\uparrow| + |\zeta_\downarrow|^2 |\downarrow\downarrow\rangle \langle\downarrow\downarrow| .$$

A few remarks regarding (2.11) are in order. Firstly, if the ensemble defining ρ only consists of one term $(1, |\psi\rangle)$, i.e., the state of the system is $|\psi\rangle$ with certainty, then the state is said to be pure. Oppositely, if the ensembles contains more than one pair, then the state is said to be mixed. This distinction between states is, however, not generally used henceforth unless it is essential. Secondly, it should be verified that this definition of a density operator actually is positive semi-definite and has trace one as well as any operator with these properties can be written in this form.

Theorem 2.15: Properties of Density Operators
A density operator $\rho \in \operatorname{End}(\mathscr{H})$ constructed by the ensemble $\{(p_i, |\psi_i\rangle)\}_i$ is positive semi-definite and has trace one if and only if it has the form

$$\rho = \sum_i p_i |\psi_i\rangle \langle\psi_i| .$$

[NC10, p. 101]

Proof
Suppose $\rho = \sum_i p_i |\psi_i\rangle \langle\psi_i|$ constructed by the ensemble $\{(p_i, |\psi_i\rangle)\}_i$. Let $|\varphi\rangle \in \mathscr{H}$ be given. Then

$$\langle\varphi|\rho|\varphi\rangle = \langle\varphi| \left(\sum_i p_i |\psi_i\rangle \langle\psi_i| \right) |\varphi\rangle = \sum_i p_i \langle\varphi|\psi_i\rangle \langle\psi_i|\varphi\rangle = \sum_i p_i |\langle\varphi|\psi_i\rangle|^2 \geq 0,$$

hence ρ is positive semi-definite. The trace of ρ is given as

$$\operatorname{Tr}(\rho) = \operatorname{Tr}\left(\sum_i p_i |\psi_i\rangle \langle\psi_i| \right) \overset{(a)}{=} \sum_i p_i \operatorname{Tr}(|\psi_i\rangle \langle\psi_i|) \overset{(b)}{=} \sum_i p_i \operatorname{Tr}(\langle\psi_i|\psi_i\rangle) = \sum_i p_i = 1,$$

where (a) follows from the linearity of the trace, (b) from the trace being cyclic, and the last equality from p_i being a probability distribution. Thus ρ also has trace one.

Conversely, suppose that ρ is positive semi-definite and has trace one. The semi-definiteness implies that ρ has spectral decomposition

$$\rho = \sum_i \lambda_i \left|i\right\rangle \left\langle i\right| ,$$

where $\lambda_i \geq 0$ for all i and the eigenvectors $\{\left|i\right\rangle\}_i$ constitute an orthonormal basis for $\mathscr{H}$. Since ρ is diagonal in the $\{\left|i\right\rangle\}_i$ basis with $\rho_{ii} = \lambda_i$, the assumption of trace one yields that $\sum_i \lambda_i = 1$. This implies that λ_i gives a probability distribution such that the system can be described by the ensemble $\{(\lambda_i, \left|i\right\rangle)\}_i$, which yield the density operator ρ by change of variables. ■

At this point, it have been shown how to construct a density operator given an ensemble. However, it would also be desirable to go in the other direction, i.e., to determine which ensemble that has generated the given density operator. An example of this follows.

Example 2.16: Determining Ensemble Generating Density Operator
Consider the density operator

$$\rho = \frac{1}{2}I, \quad \rho \in \mathrm{End}(\mathbb{C}^2).$$

The aim is now to determine the ensemble that has generated this density operator. As in Example 2.10, denote the canonical basis vectors $e_1, e_2 \in \mathbb{C}^2$ as $\left|0\right\rangle, \left|1\right\rangle$, respectively. It then trivially follows that

$$\rho = \frac{1}{2}\left(\left|0\right\rangle \left\langle 0\right| + \left|1\right\rangle \left\langle 1\right|\right),$$

which corresponds to ρ being generated by the ensemble $\left\{\left(\frac{1}{2}, \left|0\right\rangle\right), \left(\frac{1}{2}, \left|1\right\rangle\right)\right\}$, i.e., that the state is either $\left|0\right\rangle$ or $\left|1\right\rangle$ with equal probability.

However, by writing ρ differently, another generating ensemble is easily seen, e.g.,

$$\begin{aligned}\rho = \frac{1}{2}I = \frac{1}{4}\left(\begin{bmatrix}1 & 1\\ 1 & 1\end{bmatrix} + \begin{bmatrix}1 & -1\\ -1 & 1\end{bmatrix}\right) &= \frac{1}{4}\left((\left|0\right\rangle + \left|1\right\rangle)(\left\langle 0\right| + \left\langle 1\right|) + (\left|0\right\rangle - \left|1\right\rangle)(\left\langle 0\right| - \left\langle 1\right|)\right)\\ &= \frac{1}{2}\left(\frac{1}{2}(\left|0\right\rangle + \left|1\right\rangle)(\left\langle 0\right| + \left\langle 1\right|) + \frac{1}{2}(\left|0\right\rangle - \left|1\right\rangle)(\left\langle 0\right| - \left\langle 1\right|)\right),\end{aligned}$$

corresponding to ρ being generated by the ensemble $\left\{\left(\frac{1}{2}, \frac{1}{\sqrt{2}}(\left|0\right\rangle + \left|1\right\rangle)\right), \left(\frac{1}{2}, \frac{1}{\sqrt{2}}(\left|0\right\rangle - \left|1\right\rangle)\right)\right\}$. Thus, two different ensembles that generate ρ have been found. In both cases is the states in the ensembles equally likely, hence a state on the form $I/2 \in \mathrm{End}(\mathbb{C}^2)$ is referred to as the maximally mixed state.

As shown in Example 2.16, when given a density operator, it is generally not possible to determine a unique ensemble that generates it. This is more formally described in the following theorem.

Theorem 2.17: Unitary Freedom of Ensembles for Density Operator
Let two ensembles $\{(p_i, \left|\psi_i\right\rangle)\}_i, \{(q_j, \left|\varphi_j\right\rangle)\}_j$ be defined on the same system. If the cardinality of the ensembles are different, the smallest can simply be zero-padded. Then the two set of states $\{\left|\psi_i'\right\rangle\}_i = \{\sqrt{p_i}\left|\psi_i\right\rangle\}_i, \{\left|\varphi_j'\right\rangle\}_j = \{\sqrt{q_j}\left|\varphi_j\right\rangle\}_j$ generate the same density operator, i.e., $\rho = \sum_i \left|\psi_i'\right\rangle \left\langle\psi_i'\right| = \sum_j \left|\varphi_j'\right\rangle \left\langle\varphi_j'\right|$, if and only if there exists a unitary operator $U \in \mathrm{End}(\mathbb{C}^j)$ such that $\left|\psi_i'\right\rangle = \sum_j u_{ij}\left|\varphi_j'\right\rangle$ for all i, where u_{ij} is the (i,j)-th entry of U. [NC10, pp. 103-104]

Proof
Assume that $\{\left|\psi_i'\right\rangle\}_i$ and $\{\left|\varphi_j'\right\rangle\}_j$ generate the same density operator ρ. Since ρ is a density operator, it has spectral decomposition $\rho = \sum_n \lambda_n \left|n\right\rangle \left\langle n\right|$ for $\lambda_n \geq 0$ for all n and $\{\left|n\right\rangle\}_n$ constituting an orthonormal

basis cf. Theorem 2.15. For all n such that $\lambda_n \neq 0$, let $|n'\rangle = \sqrt{\lambda_n}\,|n\rangle$ such that $\rho = \sum_n |n'\rangle\langle n'|$. Notice that if $\lambda_n = 0$ for some n, then $\{|n'\rangle\}_n$ is not a basis (notice that n is abusively used as the index set of both $\{|n\rangle\}_n$ and $\{|n'\rangle\}_n$ although thee cardinalities may differ, however, the usage should be clear from context). Particularly, the cardinality of $\{|n'\rangle\}_n$ equals the rank of ρ.

Since both $\{|n'\rangle\}_n$ and $\{|\psi_i'\rangle\}_i$ generate ρ, these sets can be related. To do so, let $|\phi\rangle$ be any state orthonormal to the span of $\{|n'\rangle\}_n$. In other words, $|\phi\rangle$ is in the span of $\{|n\rangle\}_n$ for all n satisfying $\lambda_n = 0$. It then follows that

$$0 = \sum_n \langle\phi|n'\rangle\langle n'|\phi\rangle = \langle\phi|\rho|\phi\rangle = \sum_i \langle\phi|\psi_i'\rangle\langle\psi_i'|\phi\rangle = \sum_i |\langle\phi|\psi_i'\rangle|^2,$$

which implies that $\langle\phi|\psi_i'\rangle = 0$ for all i. Therefore, $|\phi\rangle$ is orthogonal to all states in $\{|\psi_i'\rangle\}_i$. By construction, it is furthermore orthogonal to all states in $\{|n'\}\rangle_n$ as it spans the subspace orthogonal to $|\psi\rangle$. This implies that all states in $\{|\psi_i'\rangle\}_i$ must be in the span of $\{|n\rangle\}_n$, i.e., they all have the form

$$|\psi_i'\rangle = \sum_n v_{in}\,|n'\rangle \tag{2.12}$$

for some complex matrix $V \in \mathrm{Hom}(\mathbb{C}^i, \mathbb{C}^n)$ with elements v_{in}. Notice that since $\rho = \sum_i |\psi_i'\rangle\langle\psi_i'|$, it is a sum of i rank-1 matrices, hence $\mathrm{rank}(\rho) \leq i$. This implies that $n \leq i$, meaning that V has at least as many rows as it has columns.

Since both $\{|n'\rangle\}_n$ and $\{|\psi_i'\rangle\}_i$ generate ρ, it now follows from (2.12) that

$$\sum_n |n'\rangle\langle n'| = \rho = \sum_i |\psi_i'\rangle\langle\psi_i'| = \sum_{i,n,m} v_{in}v_{im}^*\,|n'\rangle\langle m'|\,.$$

Since $\{|n'\rangle\}_n$ is a set of linearly independent states, $\{|n'\rangle\langle m'|\}_{n,m}$ is a set of linearly independent operators. This implies that $\sum_i v_{in}v_{im}^* = \delta_{nm}$. In other words, $VV^\dagger = I$. Now, if $V \in \mathrm{End}(\mathbb{C}^i)$, then V is unitary. If it has more rows than columns, then $i - n$ additional columns can be appended to V to make it square and thereby unitary. The set $\{|n'\rangle\}_n$ is zero-padded accordingly.

It can analogously be shown that for some unitary $W \in \mathrm{End}(\mathbb{C}^j)$ that for all j, then

$$|\varphi_j'\rangle = \sum_n w_{jn}\,|n'\rangle\,.$$

Now that both $\{|\psi_i'\rangle\}_i$ and $\{|\varphi_j'\rangle\}_j$ are expressed in terms of $\{|n'\rangle\}_n$, these two sets can be related. This is done by noticing that for any j, then

$$\begin{aligned}
|\varphi_j'\rangle = \sum_n w_{jn}\,|n'\rangle \implies \sum_m w_{jm}^*\,|\varphi_j'\rangle &= \sum_{m,n} w_{jm}^* w_{jn}\,|n'\rangle \\
\implies \sum_{j,m} w_{jm}^*\,|\varphi_j'\rangle &= \sum_{j,m,n} n w_{jm}^* w_{jn}\,|n'\rangle = \sum_{m,n} \delta_{mn}\,|n'\rangle \\
\implies \sum_{j,m} v_{im}w_{jm}^*\,|\varphi_j'\rangle &= \sum_{m,n} v_{im}\delta_{mn}\,|n'\rangle = \sum_n v_{in}\,|n'\rangle = |\psi_i'\rangle\,.
\end{aligned}$$

Defining the unitary operator $U = VW^\dagger$, gives the desired result, namely

$$|\psi_i'\rangle = \sum_j u_{ij}\,|\varphi_j'\rangle\,.$$

Conversely, assume that there exists a unitary $U \in \mathrm{End}(\mathbb{C}^j)$ relating the two set of states as described. Then

$$\rho = \sum_i |\psi_i'\rangle\langle\psi_i'| = \sum_{i,j,k} u_{ij}u_{ik}^*\,|\varphi_j'\rangle\langle\varphi_k'| = \sum_{j,k} \delta_{jk}\,|\varphi_j'\rangle\langle\varphi_k'| = \sum_j |\varphi_j'\rangle\langle\varphi_j'|\,,$$

which imply that both sets of states generate ρ, hence concluding the proof. ■

Thus, Theorem 2.17 states that density operators are defined up to some unitary operator analogue to state vectors being defined up to some global phase. To complete Example 2.16 by relating it to Theorem 2.17, the unitary operator that relates the two ensembles is $U = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$, a matrix known as the Hadamard operator.

At this point, important characteristics of the density operator has been considered. However, only the formulation of quantum states in the general picture of quantum mechanics has been changed, hence evolution and measurements of systems are still unitary operators but slightly reformulated in the density operator formalism. The evolution of states given in Postulate 2.8 is

$$|\psi(t)\rangle = U(t)\,|\psi(0)\rangle\,.$$

If the initial state of the system is described by the ensemble $\{(p_i, |\psi_i(0)\rangle)\}_i$ generating the density operator $\rho(0)$, the evolution of the system instead becomes

$$\rho(t) = \sum_i p_i U\,|\psi_i(0)\rangle\,\langle\psi_i(0)|\,U^\dagger = U\rho(0)U^\dagger. \tag{2.13}$$

If a measurement process is defined by observables $\{M_m\}_m$ and the state prior to measurement is $|\psi\rangle$, then Postulate 2.11 gives that the probability of measuring outcome m is

$$\Pr(m) = \langle\psi|M_m^\dagger M_m|\psi\rangle\,,$$

while the state after measurement is

$$|\psi'\rangle = \frac{M_m\,|\psi\rangle}{\sqrt{\Pr(m)}}. \tag{2.14}$$

The observables are still required in the density operator formalism, however, the state before measurement is now given by the ensemble $\{(p_i, |\psi_i\rangle)\}_i$. The probability of measuring outcome m is in that case given as

$$\begin{aligned} \Pr(m) &= \sum_i \Pr(m|i)p_i = \sum_i p_i\,\langle\psi_i|M_m^\dagger M_m|\psi_i\rangle = \sum_i p_i \operatorname{Tr}\left(\langle\psi_i|M_m^\dagger M_m|\psi_i\rangle\right) \\ &= \sum_i p_i \operatorname{Tr}\left(M_m^\dagger M_m\,|\psi_i\rangle\,\langle\psi_i|\right) = \operatorname{Tr}\bigl(M_m^\dagger M_m \rho\bigr), \end{aligned} \tag{2.15}$$

where the penultimate equality follows from the trace being cyclic and the last from its linearity.

The state of the system after the measurement can be found by inserting the formulas derived in (2.13) and (2.15) into (2.14), which yields

$$\rho' = \frac{M_m \rho M_m^\dagger}{\operatorname{Tr}\Bigl(M_m^\dagger M_m \rho\Bigr)}. \tag{2.16}$$

By combining Postulate 2.14, (2.13), (2.15) and (2.16), one obtain the postulates of quantum mechanics in the density operator formalism. The last postulate, which actually motivated the density operator formalism, it that the state of a composite system can be written as a tensor product of density operators. That is, the state of a composite of n subsystems with states $\rho_1, \ldots, \rho_n$ is given as $\otimes_{i=1}^n \rho_i$. This is true even in the presence of entanglement, which indicates that it is possible to determine the states of the subsystems. This is done by using the reduced density operator.

Definition 2.18: Reduced Density Operator
Let the composite system, AB, of the two system A, B with state spaces $\mathscr{H}_A, \mathscr{H}_B$ respectively, be described by the state ρ_{AB}. The reduced density operator for system A is defined as

$$\rho_A = \operatorname{Tr}_B(\rho_{AB}),$$

where Tr_B is the partial trace over system B. The reduced density operator for B is defined analogously.

[NC10, p. 105]

In order to determine the reduced density operator, the partial trace is needed. For the composite system AB with subsystems A, B having state spaces $\mathscr{H}_A, \mathscr{H}_B$, respectively, the partial trace over B, $\mathrm{Tr}_B \in \mathrm{Hom}(\mathscr{H}_{AB}, \mathscr{H}_A)$, is defined as

$$\mathrm{Tr}_B((|\psi_1\rangle_A \langle\psi_2|_A) \otimes (|\varphi_1\rangle_B \langle\varphi_2|_B)) = |\psi_1\rangle_A \langle\psi_2|_A \mathrm{Tr}(|\varphi_1\rangle_B \langle\varphi_2|_B) = \langle\varphi_2|\varphi_1\rangle_B |\psi_1\rangle_A \langle\psi_2|_A ,$$

where the states in the systems are chosen arbitrarily. Taking the partial trace over subsystem B is also called 'tracing out B', which essentially corresponds to forgetting about subsystem B. The linearity of the partial trace then directly imply that the reduced density operator of a composite system actually yields the density operator on the corresponding subsystem. Explicitly, let ρ_A, ρ_B be generated by the ensembles $\{(p_i, |\psi_i\rangle)\}_i, \{(q_j, |\varphi_j\rangle)\}_j$, respectively. Then

$$\begin{aligned}\mathrm{Tr}_B(\rho_{AB}) &= \mathrm{Tr}_B\left[\left(\sum_i p_i |\psi_i\rangle\langle\psi_i|\right) \otimes \left(\sum_j q_j |\varphi_j\rangle\langle\varphi_j|\right)\right] = \sum_{i,j} p_i q_j \,\mathrm{Tr}_B\left[(|\psi_i\rangle\langle\psi_i|) \otimes (|\varphi_j\rangle\langle\varphi_j|)\right] \\ &= \sum_{i,j} p_i q_j \langle\varphi_j|\varphi_j\rangle |\psi_i\rangle\langle\psi_i| = \sum_i p_i |\psi_i\rangle\langle\psi_i| = \rho_A.\end{aligned}$$

Hence, given a density operator of a composite system, it is actually possible to determine a (reduced) density operator for the subsystems. In other words, one can describe the state of subsystems without considering the entire composite system, which was the issue with state vectors in the case of entanglement. An example of this follows.

Example 2.19: Reduced Density Operator for Spin of Two Electrons
Consider once again the case where Alice and Bob each possess an electron. Let their composite system be entangled such that it has the form of the EPR pair $|\Phi^+\rangle$ defined in Definition 2.6. That is,

$$|\varphi\rangle_{AB} = \frac{1}{\sqrt{2}}(|\uparrow\uparrow\rangle + |\downarrow\downarrow\rangle),$$

where $\{|\uparrow\rangle, |\downarrow\rangle\}$ is an orthonormal basis of $\mathbb{C}^2$. The density operator on the composite system is then given as

$$\rho_{AB} = |\varphi\rangle_{AB}\langle\varphi|_{AB} = \frac{1}{2}(|\uparrow\uparrow\rangle\langle\uparrow\uparrow| + |\uparrow\uparrow\rangle\langle\downarrow\downarrow| + |\downarrow\downarrow\rangle\langle\uparrow\uparrow| + |\downarrow\downarrow\rangle\langle\downarrow\downarrow|).$$

Now, Alice wants to know something about her system without taking Bob's electron into account. She therefore traces system B out, yielding

$$\begin{aligned}\rho_A = \mathrm{Tr}_B(\rho_{AB}) &= \frac{1}{2}(\langle\uparrow|\uparrow\rangle |\uparrow\rangle\langle\uparrow| + \langle\downarrow|\uparrow\rangle |\uparrow\rangle\langle\downarrow| + \langle\uparrow|\downarrow\rangle |\downarrow\rangle\langle\uparrow| + \langle\downarrow|\downarrow\rangle |\downarrow\rangle\langle\downarrow|) \\ &= \frac{1}{2}(|\uparrow\rangle\langle\uparrow| + |\downarrow\rangle\langle\downarrow|) = \frac{1}{2}I,\end{aligned}$$

where the orthonormality of the states has been used repeatedly. Thus her electron is in the maximally mixed state. This implies that, if Alice were to perform a local projective measurement, i.e., only on her electron, defined by the observables $P_\uparrow = |\uparrow\rangle\langle\uparrow|, P_\downarrow = |\downarrow\rangle\langle\downarrow|$, (2.15) gives that the probability of her measuring the spin in the up state is

$$\Pr(\uparrow) = \mathrm{Tr}(P_\uparrow \rho_A) = \frac{1}{2}\mathrm{Tr}(|\uparrow\rangle\langle\uparrow|) = \frac{1}{2}.$$

Thus, Alice's measurement is completely random. Say that she measures the electron in the up state. After the measurement, which corresponds to $(P_\uparrow \otimes I)$, the state of the composite system changes according to Equation (2.13), which yields

$$\begin{aligned}\rho'_{AB} = \frac{(P_\uparrow \otimes I)\rho_{AB}(P_\uparrow \otimes I)}{\Pr(\uparrow)} &= (|\uparrow\rangle\langle\uparrow| \otimes I)(|\uparrow\uparrow\rangle\langle\uparrow\uparrow| + |\uparrow\uparrow\rangle\langle\downarrow\downarrow| + |\downarrow\downarrow\rangle\langle\uparrow\uparrow| + |\downarrow\downarrow\rangle\langle\uparrow\uparrow|)(|\uparrow\rangle\langle\uparrow| \otimes I) \\ &= \sum_{i,j\in\{\uparrow,\downarrow\}} (|\uparrow\rangle\langle\uparrow|i\rangle\langle j|\uparrow\rangle\langle\uparrow|) \otimes (|i\rangle\langle j|) \overset{(a)}{=} (|\uparrow\rangle\langle\uparrow|) \otimes (|\uparrow\rangle\langle\uparrow|) = |\uparrow\uparrow\rangle\langle\uparrow\uparrow| ,\end{aligned}$$

> where (a) follows from $\{|{\uparrow}\rangle, |{\downarrow}\rangle\}$ constituting an orthonormal basis for $\mathbb{C}^2$. This implies that the state of the composite system is $|{\uparrow\uparrow}\rangle$. Hence, Bob's electron is completely determined by Alice performing a measurement of her electron. The above arguments also holds if Bob's performs the measurement rather than Alice, or if the state of the composite system was any of the other EPR pairs. Since a local measurement of any EPR pair will determine the state of the other electron with certainty, EPR pairs are said to be a maximally entangled state of a composite system consisting of two two-level subsystems.

The general framework for quantum mechanics has now been presented with two different, but equivalent, formalism. Both of these will be used throughout the thesis depending on which is more convenient.

3 | Quantum Information Theory

Since noise is one of the greatest impediments for quantum computers to achieve their potential, it is necessary to correct such noise. However, in order to do so, it is necessary to precisely define what is meant by noise in this scenario, which in turn requires description of the information that is contained in quantum systems. A gentle introduction to these concepts is presented in this chapter, which is based on several chapters from [NC10].

Quantum information theory can to a large extend be considered as a generalisation of classic information theory. Much of the terminology is therefore presented subsequent to a brief review of the classical analogous. However, when dealing with quantum mechanical systems, new results also occur.

The basic unit of classical information is a bit, which is generalised to a quantum bit, commonly known as a qubit. Although briefly introduced in Section 1.2.2, qubits are properly described in the following section, which is based on [NC10, ch. 1.2].

3.1 Qubits

A bit is a general unit to describe the state of any two-level classical, e.g., the coin showing heads/tails after a coin flip or the presence/absence of electrons in a circuit, and is the foundation for modern classical computing. For quantum systems and quantum computing, bits are generalised to qubits that describe any two-level quantum system, e.g., the spin of an electron as exemplified in Example 2.2. Thus, rather than having to think about the actual physical system, say of an electron, one can simply consider the more abstract qubit, which then should have the same information as the system. Analogue to the classical case, the two levels of a two-level quantum system are referred to as the ground state and the excited state. However, due to the superposition property, quantum states can be in a superposition of the ground and excited states. From a purely mathematical viewpoint, a qubit is cf. Postulate 2.1 a unit vector in $\mathbb{C}^2$, hence a linear combination of any basis of $\mathbb{C}^2$. Although any basis can be used to describe a qubit, the canonical basis is often chosen for simplicity. That is, the ground state and the excited state each correspond to one of the canonical basis vectors. There are, however, cases where another basis is more convenient, but this will be explicitly emphasised. Now, to make the analogue between a bit and a qubit as clear as possible, some notation is introduced. Let the ground state and excited state be given as

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

respectively. With this notation, the set $\{|0\rangle, |1\rangle\}$ is an orthonormal basis for $\mathbb{C}^2$. Thus, for $\alpha, \beta \in \mathbb{C}$, a qubit $|\psi\rangle$ has the form

$$|\psi\rangle = \alpha |0\rangle + \beta |1\rangle, \quad |\alpha|^2 + |\beta|^2 = 1. \tag{3.1}$$

Notice that if either α or β is zero, there is no superposition, and the qubit corresponds to a classical bit, i.e., it is either 0 or 1, depending on which amplitude that is zero.

In order to gain a deeper understanding of a qubit, another representation than (3.1) is convenient, although it should be noted that (3.1) is the form typically used henceforth. Generally, a vector in $\mathbb{C}^2$ is described by four real variables, however this can be reduced since a qubit is a unit vector. To see this, it is convenient to express α, β in polar form, yielding

$$|\psi\rangle = r_0 \mathrm{e}^{\mathrm{i}\theta_0} |0\rangle + r_1 \mathrm{e}^{\mathrm{i}\theta_1} |1\rangle, \quad r_0^2 + r_1^2 = 1,$$

where $r_0, r_1 \geq 0$ and $\theta_0, \theta_1 \in [0, 2\pi)$. As described in Section 2.3, a qubit is equivalent up to a global phase, thus it can be rotated such that one of the coefficients α, β is real, e.g.,

$$|\psi\rangle \equiv \ \mathrm{e}^{-\mathrm{i}\theta_0} |\psi\rangle = r_0 |0\rangle + r_1 \mathrm{e}^{\mathrm{i}(\theta_1 - \theta_0)} |1\rangle, \quad r_0^2 + r_1^2 = 1.$$

The qubit has now three real variables, namely the magnitudes r_0, r_1 and the relative phase between the coefficients, $\theta_1 - \theta_0$. By using Cartesian coordinates again, the qubit can be written as

$$|\psi\rangle = r_0 |0\rangle + (x_1 + \mathrm{i}y_1) |1\rangle , \quad r_0^2 + x_1^2 + y_1^2 = 1.$$

The constraint is equivalent to saying that a qubit is a point (x_1, y_1, r_0) on the unit sphere in $\mathbb{R}^3$, hence it can conveniently be written in spherical coordinates as

$$|\psi\rangle = \cos(\theta) |0\rangle + (\sin(\theta)\cos(\phi) + \mathrm{i}\sin(\theta)\sin(\phi)) |1\rangle = \cos(\theta) |0\rangle + \sin(\theta)\mathrm{e}^{\mathrm{i}\phi} |1\rangle , \quad \theta \in [0, \pi], \phi \in [0, 2\pi).$$

The issue with this parametrisation is that $|\psi\rangle = |0\rangle$ when $\theta = 0$ or $\theta = \pi$ due to only being defined up to a global phase. Hence, the state is not uniquely described. This can nonetheless be fixed by using half-angles such that

$$|\psi\rangle = \cos\left(\frac{\theta}{2}\right) |0\rangle + \sin\left(\frac{\theta}{2}\right) \mathrm{e}^{\mathrm{i}\phi} |1\rangle , \quad \theta \in [0, \pi], \phi \in [0, 2\pi).$$

It should be noted that the above form only uniquely defines a state given (θ, ϕ), not the other way around. As an example, the two orthogonal basis states $|0\rangle$ and $|1\rangle$ are found when $(\theta = 0, \phi)$ and $(\theta = \pi, \phi)$, respectively, where $\phi \in [0, 2\pi)$ is arbitrary. Thus a qubit can be visualised on a sphere that is often known as a Bloch sphere in which the classical bit corresponds to the north and south poles as shown in Figure 3.1

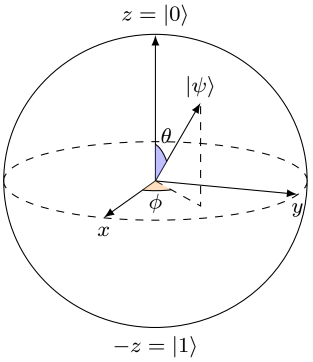


Figure 3.1: Illustration of the qubit $|\psi\rangle$ using a Bloch sphere.

The Bloch sphere illustrates how a qubit in the two extremes is reduced to a classical bit, but also can be in a superposition of it. Since there are an infinite amount of different such superpositions, a qubit seemingly contain an infinite amount of information. However, as described in Section 2.3, a qubit can only be measured in two states and therefore collapses into a classical bit in the measurement process. A qubit can therefore be considered as a generalisation of a bit that contains an infinite amount of hidden information, which is exactly the property that can be utilised in quantum computing. This becomes more apparent when considering a setup with multiple qubits.

3.1.1 Multiple Qubits

When considering a composite system of n two-level systems, the state space is the n-fold tensor product of $\mathbb{C}^2$ with itself, henceforth denoted as $(\mathbb{C}^2)^{\otimes n}$, cf. Postulate 2.3. If the system is separable, the state of the composite system is merely the tensor product of the states of the subsystems, however, this does not hold in the case of entanglement. In the general case, any state of a composite system of two two-level systems has the form

$$|\psi\rangle = \alpha_{00} |00\rangle + \alpha_{01} |01\rangle + \alpha_{10} |10\rangle + \alpha_{11} |11\rangle , \quad |\alpha_{00}|^2 + |\alpha_{01}|^2 + |\alpha_{10}|^2 + |\alpha_{11}|^2 = 1.$$

More generally, for a composite system of n two-level systems, any state can be written as

$$|\psi\rangle = \sum_{i \in \{0,1\}^n} \alpha_i |i\rangle , \quad \sum_{i \in \{0,1\}^n} |\alpha_i|^2 = 1,$$

where $\{0,1\}^n$ indicate bit strings of length n. Thus, such a state requires 2^n complex coefficients, whereas a classical system of n two-level components only requires real n bits. Quantum states therefore have an exponential growth in coefficients, whereas it for classical states only is linear. It is in fact this increase in coefficients combined with the capability of quantum parallelism that in many cases make quantum computing superior to classical computing.

In the case of an entangled composite system, it is often more convenient to use the density operator formalism to represent qubits.

3.1.2 Qubits in the Density Operator Formalism

If the state of a two-level quantum system is a pure qubit, then the general form for a qubit given in (3.1), $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$, can be used to determine the corresponding representation in the density operator formalism. Explicitly,

$$\begin{aligned}\rho = |\psi\rangle\langle\psi| = (\alpha|0\rangle + \beta|1\rangle)(\alpha^*\langle 0| + \beta^*\langle 1|) &= |\alpha|^2|0\rangle\langle 0| + \alpha\beta^*|0\rangle\langle 1| + \alpha^*\beta|1\rangle\langle 0| + |\beta|^2|1\rangle\langle 1| \\ &= \begin{bmatrix} |\alpha|^2 & \alpha\beta^* \\ \alpha^*\beta & |\beta|^2. \end{bmatrix}\end{aligned}$$

Due to the normalisation constraint $|\alpha|^2 + |\beta|^2 = 1$ and the non-diagonal terms being each others' complex conjugate, it is also often written on the form

$$\rho = \begin{bmatrix} \rho_{00} & \rho_{01} \\ \rho_{01}^* & 1-\rho_{00} \end{bmatrix}, \tag{3.2}$$

where the subscript refers to which coefficient it represents not its entry, e.g., ρ_{00} corresponds to the coefficient in front of the $|0\rangle\langle 0|$. From the normalisation constraint, it follows that $\rho_{00} \in [0,1]$.

The Bloch sphere also provides an illustrative interpretation of qubits in the density operator formalism. Since density operators are Hermitian, it follows from Theorem B.7 that any density can be written in the form

$$\rho = aI + bX + cY + dZ = \begin{bmatrix} a+d & b-\mathrm{i}c \\ b+\mathrm{i}c & a-d \end{bmatrix}, \quad a,b,c,d \in \mathbb{R}.$$

However, ρ must furthermore have trace one and be positive semi-definite in order to be a density operator cf. Theorem 2.15. Considering the former requirement, it trivially follows that $a = 1/2$. Thus, ρ can also be expressed on the form

$$\rho = \frac{1}{2}(I + bX + cY + dZ), \quad b,c,d \in \mathbb{R}, \tag{3.3}$$

where the coefficients abusively have been represented by the same letters. The coefficients are often represented in the so-called Bloch vector $r \in \mathbb{R}^3$, which have entries b, c, d.

The latter requirement implies that ρ has non-negative eigenvalues, which is equivalent to solving $\det(\rho - \lambda I) \geq 0$. By using the representation of ρ given in (3.3), the determinant yields

$$\begin{aligned}\det(\rho - \lambda I) = \det\left(\frac{1}{2}\begin{bmatrix} 1+d-2\lambda & b-\mathrm{i}c \\ b+\mathrm{i}c & 1-d-2\lambda \end{bmatrix}\right) &= \frac{1}{4}\left(4(\lambda^2 - \lambda) + 1 - b^2 - c^2 - d^2\right) \\ &= \lambda^2 - \lambda + \frac{1}{4}(1 - b^2 - c^2 - d^2).\end{aligned}$$

Solving for λ with the quadratic formula yields

$$\lambda = \frac{1}{2}\left(1 \pm \sqrt{1 - (1 - b^2 - c^2 - d^2)}\right) = \frac{1}{2}\left(1 \pm \sqrt{b^2 + c^2 + d^2}\right) = \frac{1}{2}(1 \pm \|r\|).$$

The non-negativity of the eigenvalues then imply that $\|r\| \leq 1$ in order for ρ to be a valid density operator. If $\|r\| = 1$, the eigenvalues of ρ are 0 and 1, which imply that ρ has rank one, which in turn imply that it has the form $\rho = |\psi\rangle\langle\psi|$ for some state $|\psi\rangle \in \mathbb{C}^2$. In other words, ρ is pure. Oppositely, if $\|r\| \leq 1$, then ρ

must be a mixed state. Thus, pure states lie on the Bloch sphere, while mixed states are in the interior of the Bloch ball.

As the standard form for qubits have been presented, the question of how much information they contain has been left partially unanswered, as it only has been argued that they contain an infinite amount of hidden information. In order to answer it more precisely, information must be defined properly, which is done in the following section based on [NC10, ch. 11.1, 11.3]

3.2 Entropy

Before discussing information contained in a quantum system, the information in a classical system is briefly summarised. This also enables a comparison of some of their characteristics.

3.2.1 Shannon Entropy

Defining information in general is somewhat philosophical, however, Claude Shannon quantified information in the seminal paper *A Mathematical Theory of Communication* ([Sha48]). Loosely speaking, the amount of information gained by receiving a message depends on how surprising that message it. For example, a lottery ticket buyer does not gain much information by being told that his ticket did not win the jackpot as it would be very unlikely to win. However, he would be very surprised to learn that his ticket in fact did win. This idea of information is quantified by the Shannon entropy. More precisely, if X is a discrete random variable with probability distribution $\{p_x\}_x$, the Shannon entropy of X is given by

$$\mathrm{H}(X) = -\sum_x p_x \log(p_x),$$

where the convention that $0\log(0) = 0$ is used (as justified by the limit of $p\log(p) = 0$ for p tending to 0^+). The binary logarithm is typically used, in which case the Shannon entropy is measured in bits. In that case, another useful interpretation of the Shannon entropy is that is measures the expected number of binary questions needed to determine the outcome of X (or equivalently how many bits one in average need to describe the outcome). This interpretation is illustrated in the following example.

Example 3.1: Entropy of Coin Flips
Consider a game where Alice flip a coin, whereafter Bob must determine its outcome with as few binary questions as possible. In this game, Bob knows the distribution of the coin being flipped.

In the first game, Alice flips a fair coin. Considering that both outcomes are equally likely, Bob can do no better than asking “is it tails?”. Since there are only two possible outcomes, Bob knows the outcome with certainty no matter what Alice responds. Thus, a single binary question is needed in this case.

In the second game, Alice flips a very thick coin that lands on its edge half of the times while heads or tails is a forth of the time each. Considering that there are three possible outcomes, Bob seemingly needs two binary questions to know the outcome with certainty. However, he can do better in average due to knowing the distribution of the biased coin. If he first asks “is it edge?”, one question is sufficient half of the time. If it is not edge, then he knows that it is heads or tails with equal probability, hence one additional question is enough cf. the arguments given in game one. Thus, half og the time is one question sufficient, while two is needed in the other half. In other words, Bob needs 1.5 binary questions on average in the second game.

These ideas are now compared to the entropy of the random variables for the fair coin, X_f, and the thick coin, X_t:

$$\begin{aligned}
\mathrm{H}(X_f) &= -\left(\frac{1}{2}\log_2\left(\frac{1}{2}\right) + \frac{1}{2}\log_2\left(\frac{1}{2}\right)\right) = -\log_2\left(\frac{1}{2}\right) = 1, \\
\mathrm{H}(X_t) &= -\left(\frac{1}{2}\log_2\left(\frac{1}{2}\right) + \frac{1}{4}\log_2\left(\frac{1}{4}\right) + \frac{1}{4}\log_2\left(\frac{1}{4}\right)\right) = -\frac{1}{2}\left(\log_2\left(\frac{1}{2}\right) + \log_2\left(\frac{1}{4}\right)\right) = 1.5.
\end{aligned}$$

Thus, the Shannon entropies are consistent with the number of binary questions needed in the two games.

The interpretation of the Shannon entropy is quite intuitive, something which also generalises to many of its uses. One example of such intuition is that one cannot decrease the entropy of a system by introducing another random variable into it. More formally, the joint entropy two random variables X and Y with joint probability distribution $\{p_{xy}\}_{x,y}$ is generalised naturally from the Shannon entropy as

$$\mathrm{H}(X,Y) = -\sum_{x,y} p_{xy}\log(p_{xy}).$$

Then, intuition indicates that $\mathrm{H}(X) \leq \mathrm{H}(X,Y)$, which in fact is easily shown. From this, it is also naturally to expect that one cannot become more uncertain about a random variable given knowledge about another. In other words, two random variables cannot contain negative information about each other, something which is quantified by the mutual information

$$\mathrm{I}(X,Y) = \mathrm{H}(X) + \mathrm{H}(Y) - \mathrm{H}(X,Y).$$

Thus, the mutual information should be non-negative by intuition, which again is easily shown to be true.

The above formulae are only two of many examples of the usage of the Shannon entropy. In fact, is has become the fundamental entity in classical information theory, thus arising essentially everywhere in one form or another. Particularly, it arises naturally in Shannon's noiseless coding theorem, while also appearing in Shannon's noisy-channel theorem through the channel capacity [Sha48]. This topic therefore deserves a more complete discussion, however, it is not the focus in this thesis. The aim is instead to find a similar measure for quantum information.

3.2.2 Von Neumann Entropy

When considering quantum systems, there are generally two types of uncertainty; a) the quantum uncertainty due to not knowing the amplitudes in the superposition of states, and b) the classical uncertainty due to typically not knowing the exact state vector, only an ensemble of states. Since the amplitudes are unattainable, they contain hidden information, hence cannot be quantified. They aim is therefore to quantify the classical uncertainty, which typically is done with the von Neumann entropy.

Definition 3.2: Von Neumann Entropy
Let a quantum system be described by a density operator ρ. The von Neumann of the system is then defined as

$$S(\rho) = -\operatorname{Tr}(\rho\log_2(\rho)).$$

[NC10, p. 510]

Considering the definition of the von Neumann entropy, it is not clear that it describes uncertainty about the state of the system. Furthermore, the computation of it is quite cumbersome as it requires finding the matrix logarithm of the density operator (its existence follows from ρ being Hermitian). It is therefore convenient to find another representation for the von Neumann entropy. The first hurdle is to find the matrix logarithm of ρ, that is to determine a matrix A such that $2^A = \rho$. The general approach to do so is to diagonalise ρ since the matrix exponential in that case simply is to take the exponential element-wise. Thus, consider the spectral decomposition of ρ, i.e., $\rho = UDU^\dagger$ where U is unitary and D is diagonal containing the non-negative eigenvalues $\{\lambda_i\}_i$ due to the ρ being positive semi-definite. It then follows that

$$2^{UDU^\dagger} = \mathrm{e}^{\ln(2)UDU^\dagger} = \sum_{n=0}^{\infty}\frac{(\ln(2))^n}{n!}(UDU^\dagger)^n = U\left(\sum_{n=0}^{\infty}\frac{(\ln(2))^n}{n!}D^n\right)U^\dagger = U\mathrm{e}^{\ln(2)D}U^\dagger = U2^DU^\dagger.$$

It therefore in turn follows that

$$2^{U\log_2(D)U^\dagger} = U2^{\log_2(D)}U^\dagger = UDU^\dagger \implies \log_2(UDU^\dagger) = U\log_2(D)U^\dagger.$$

Thus, inserting the spectral decomposition of ρ into the definition of the von Neumann entropy yields

$$\begin{aligned} S(\rho) &= -\operatorname{Tr}\left(UDU^\dagger\log_2\left(UDU^\dagger\right)\right) = -\operatorname{Tr}\left(UDU^\dagger U\log_2(D)U^\dagger\right) \\ &\overset{(a)}{=} -\operatorname{Tr}(D\log_2(D)) \overset{(b)}{=} -\sum_i \lambda_i\log_2(\lambda_i), \end{aligned} \tag{3.4}$$

where (a) follows from U being unitary and the trace being cyclic and (b) from the D being diagonal such that the matrix logarithm is taking element-wise. Again, the convention that $0 \log_2(0) = 0$ is used. In this form, the von Neumann entropy is similar to the Shannon entropy where the eigenvalues of ρ take the place of the probability distribution of X. In fact, since ρ is positive semi-definite and has trace one cf. Theorem 2.15, all eigenvalues are non-negative and sum to one, which makes the resemblance of the two entropies even more coherent. It should explicitly be noted that the form of the von Neumann entropy given in (3.4) is given in the eigenbasis of ρ. This implies that if ρ is a pure state such that its eigenvalues are 0 and 1, then the entropy is zero. Thus, pure states in the quantum case corresponds to deterministic variables in the classical case. This is intuitively true, since one can theoretically always choose a suitable basis to measure the state such that the outcome is deterministic. Explicitly, if the state of a system is $|\psi\rangle = \alpha |0\rangle + \beta |1\rangle$ such that $\rho = |\psi\rangle \langle\psi|$, then one could simply measure in the canonical basis and obtain $|0\rangle$ (or $|1\rangle$) with probability $|\alpha|^2$ (or $|\beta|^2$). However, if one measure in the eigenbasis, which in this case is $\{|\psi_0\rangle, |\psi_1\rangle\}$ corresponding to eigenvalues 0 and 1, respectively. They are easily seen to be

$$|\psi_0\rangle = \beta^* |0\rangle - \alpha^* |1\rangle, \quad |\psi_1\rangle = |\psi\rangle .$$

Hence, one could theoretically measure in the $\{|\psi_0\rangle, |\psi_1\rangle\}$ basis and then measure the state as $|\psi\rangle$ with certainty. In practice, it is however impossible to determine this suitable basis due to the amplitudes being unknown. In this sense, the von Neumann entropy measures the uncertainty of the state in the best basis. A simple, yet very insightful, example of calculating the von Neumann entropy for a system follows.

> **Example 3.3: Von Neumann Entropy of an EPR Pair**
> Consider the case where Alice and Bob share the EPR pair $|\Phi^+\rangle$ such that the state of their composite system is $|\psi\rangle_{AB} = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$. Let ρ_{AB} denote the corresponding density operator. Since the state of the composite system is a pure state, its von Neumann entropy is zero, i.e., $S(\rho_{AB}) = 0$. Now, consider the case where Alice wants to determine the uncertainty of her subsystem alone, which is done by calculating the von Neumann entropy of the reduced density operator ρ_A. By Example 2.19, the state of her subsystem is maximally mixed implying that its uncertainty is maximal. Explicitly, $\rho_A = I/2$, which implies that $S(\rho_A) = 1$. Since the composite system is maximally entangled, the qubit possessed by Alice is known as an entangled bit (ebit), which has maximal uncertainty. At first this may seem as a minor result, however it demonstrate the power of entanglement. The von Neumann entropies illustrate that although one may be certain about the state of a system, this certainty does not necessary apply to the subsystems. In other words, $S(\rho_A) \not\leq S(\rho_{AB})$ in general, which is in contrast to the classical case, where $\mathrm{H}(X) \leq \mathrm{H}(X, Y)$ always holds.

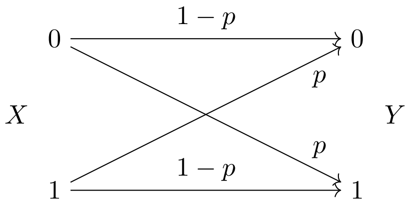
The difference between the two entropy measures illustrated in Example 3.3 demonstrates the power of entanglement as a resource as it can be used to reduce the uncertainty of a system. Another scenario where the power of entanglement become apparent is when considering how much classical information one can encode into a quantum system. This is quantified by Holevo's bound, which at the risk of simplifying, states that it is impossible to obtain more than n bits of classical information by n qubits alone [NC10, ch. 12.1]. However, it has been shown in the superdense coding communication protocol that it in fact is possible to obtain $2n$ bits of classical information by n qubits given that the sender and receiver share an EPR pair [NC10, ch. 2.3].

The above results of quantum information theory indicate that although it to a large extent is similar to the classical counterpart, the presence of entanglement lead to some remarkable results. Examples of similarities is Schumacher's noiseless coding theorem and the Holevo-Schumacher-Westmoreland noisy-channel theorem for classical information as generalisations of Shannons theorems in the classical case [NC10, ch. 12.2-12.3]. These results deserve a discussion in their own, however the information-theoretic approach to communication is not considered further in this thesis. Instead, a purely coding-theoretic approach is taken. Before doing so, it is necessary to discuss the necessity of quantum coding theory by considering how quantum information may be corrupted by noise. This is done in the following section based on [NC10, ch. 8]

3.3 Quantum Noise

As discussed in the introduction, one of the main obstacles for quantum computing is that quantum computers are noisy due to not being perfectly isolated. Although the physical attributes of the noise may depends on the particular quantum computer architecture in question, the effect of noise can be described in a general manner. This will in fact also be able to describe noisy occurring in quantum communication that may be used in a network of quantum computers. Before describing the general concepts and providing some examples, noise in classical computing/communication is briefly considered.

In the classical case is information encoded in some manner into a signal, which is sent through some physical medium, in which errors may occur. Physically, the medium may be a wire from gate to gate or from memory to the CPU, while it for communication may be free space. Rather than considering the physical attributes of the medium, which may vary significantly, it is convenient to simply consider a theoretical model for a medium known as a channel as depicted in Figure 1.2. In that way, one can consider errors occurring in computing and communication simultaneously, although a given channel may be more suitable in one scenario than the other. Assuming that the information is binary, it may be corrupted by a bit flip or a bit may even be lost. A channel is then often described by a stochastic matrix that acts upon a stochastic vector describing the information to be transmitted. The outcome is then another stochastic vector that is received by the receiver. A simple example is the binary symmetric channel, which takes a bit as input (X), and flips it with probability p to yield the output (Y). This is both visually and mathematically illustrated in Figure 3.2.



(a) Graphical representation.

$$\begin{bmatrix} \Pr(Y=0) \\ \Pr(Y=1) \end{bmatrix} = \begin{bmatrix} 1-p & p \\ p & 1-p \end{bmatrix} \begin{bmatrix} \Pr(X=0) \\ \Pr(X=1) \end{bmatrix}.$$

(b) Stochastic process representation.

Figure 3.2: Two different representations of the binary symmetric channel.

In order to communicate a bit string of length n over a binary symmetric channel, it is simply used n times independently. In other words, errors are assumed to be independent and independently distributed. The success of the n bits being noiseless is therefore $(1-p)^n$.

The idea that classical channels, and thereby noise, can be described by some stochastic matrix acting upon the stochastic input/state vector can be generalised to describe quantum channels. The assumption that bit errors are independent and identically distributed is also used in the quantum case for simplicity in this thesis. Thus, it is sufficient to consider noise acting on a single qubit. The general theory of quantum channels will, however, be presented for an arbitrary state space.

3.3.1 Quantum Channels

In a closed quantum system with state space $\mathscr{H}$, states are by Postulate 2.14 a density operator, $\rho \in \mathrm{End}(\mathscr{H})$, that evolves according to some unitary transformation, $U \in \mathrm{End}(\mathscr{H})$, as described in Section 2.2. That is, the evolution of states has the form $\mathcal{E}(\rho) = U\rho U^\dagger$ as stated in (2.13). The map $\mathcal{E} : \mathrm{End}(\mathscr{H}) \to \mathrm{End}(\mathscr{H})$ is called a quantum operation or a quantum channel, but henceforth often referred to as a channel. In practice, quantum systems are nonetheless never closed as they always interact with the environment, often in a noisy manner, e.g., due to heat transfer. It is therefore necessary to define channels over a composite system consisting of the environment and the system of interest, i.e., the system of which ρ is a state, which henceforth is referred to as the principal system. Using (2.13) once again, a channel over the composite system has the form

$$\mathcal{E}(\rho \otimes \rho_e) = U(\rho \otimes \rho_e)U^\dagger,$$

where ρ, ρ_e are the states of the principal system and environment, respectively, before the channel acts upon the composite system. However in the end, the evolution of neither the composite system nor the environment is of interest, but only that of the principal system. This is illustrated in Figure 3.3, where $\mathcal{E}(\rho)$ denotes the mapping of the channel on the principal system.

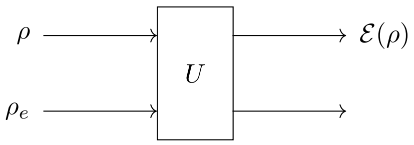


Figure 3.3: Channel over a composite system of a principal system and the environment, where only the effect of the channel on the principal system is of interest.

It should be noted that even though U is a unitary operator on the composite system, it is not necessarily unitary on the principal system (see Section B.1.1 for further details). Since only $\mathcal{E}(\rho)$ is of interest, this is found by tracing out the environment, i.e.,

$$\mathcal{E}(\rho) = \mathrm{Tr}_e(\mathcal{E}(\rho \otimes \rho_e) = \mathrm{Tr}_e\left(U(\rho \otimes \rho_e)U^\dagger\right). \tag{3.5}$$

Thus, a quantum channel on a principal system is given as a reduced density operator of a unitary transformation of the composite system consisting of the principle system and the environment. The problem with (3.5) is that the environment is much more complex than the principal system in the sense that the state space of the environment is enormous compared to the principal system. The dimensions of the unitary operator U must therefore also be large although the principal system may be very simple, making computations infeasible. In order to overcome this, let $\{|e_n\rangle\}_n$ be an orthonormal basis of the state space of the environment, $\mathscr{H}_E$. The state of the principal system after applying the channel can then by Theorem B.8 be written as

$$\mathcal{E}(\rho) = \mathrm{Tr}_e\left(U(\rho \otimes \rho_e)U^\dagger\right) = \sum_n (I \otimes \langle e_n|)U(\rho \otimes \rho_e)U^\dagger(I \otimes |e_n\rangle). \tag{3.6}$$

Now, without loss of generality, let $\rho_e = |e_0\rangle\langle e_0|$ (if ρ_e cannot be written as a pure state for the environment, one can always consider a larger environment in which it is a pure state, a process known as purification. For further details, see [NC10, ch. 2.5]). It then follows from Corollary B.4 that the initial state of the composite system is

$$\rho \otimes \rho_e = \rho \otimes (|e_0\rangle\langle e_0|) \overset{(a)}{=} (I\rho I) \otimes (|e_0\rangle\, 1 \,\langle e_0|) = (I \otimes |e_0\rangle)(\rho \otimes 1)(I \otimes \langle e_0|) = (I \otimes |e_0\rangle)\rho(I \otimes \langle e_0|),$$

where the identity matrices and scalar one are inserted in (a) for clarification. Inserting this expression into (3.6) yields

$$\mathcal{E}(\rho) = \sum_n (I \otimes \langle e_n|)U(I \otimes |e_0\rangle)\rho(I \otimes \langle e_0|)U^\dagger(I \otimes |e_n\rangle).$$

By defining the so-called Kraus operators

$$E_n = (I \otimes \langle e_n|)U(I \otimes |e_0\rangle),$$

the state $\mathcal{E}(\rho)$ can be simplified as

$$\mathcal{E}(\rho) = \sum_n E_n \rho E_n^\dagger,$$

which is known as the operator-sum representation of open quantum systems. Now, for $\mathcal{E}(\rho)$ to be a state, it must have trace one cf. Theorem 2.15. That is,

$$1 = \mathrm{Tr}(\mathcal{E}(\rho)) = \mathrm{Tr}\left(\sum_n E_n \rho E_n^\dagger\right) = \mathrm{Tr}\left(\sum_n E_n^\dagger E_n \rho\right).$$

Since $\mathrm{Tr}(\rho) = 1$, the above expression implies the completeness equation

$$\sum_n E_n^\dagger E_n = I.$$

The state $\mathcal{E}(\rho)$ must furthermore be positive semi-definite, however, this follows directly from its form and ρ being positive semi-definite.

With the operator-sum representation of the system, channels can be represented by a collection of Kraus operators, $\{E_n\}_n$, that only act on the principle system with the constraint that they satisfy the completeness equation. It should be noted that the Kraus operators are not unique as described in the following theorem.

> **Theorem 3.4: Unitary Freedom of Kraus Operators**
> Let the sets $\{E_1, \dots, E_n\}, \{F_1, \dots, F_m\}$ describe the channels $\mathcal{E}$ and $\mathcal{F}$, respectively. If $n \neq m$, the shortest collection of Kraus operators can be zero-padded such that $n = m$. Then $\mathcal{E} = \mathcal{F}$ if and only if there exists a unitary transformation $U \in \operatorname{End}(\mathbb{C}^n)$ such that $E_i = \sum_j u_{ij} F_j$ for all i. [NC10, p. 372]

Proof
Assume that $\mathcal{E} = \mathcal{F}$, i.e., the sets of Kraus operators $\{E_l\}_l, \{F_j\}_j$ satisfy $\sum_l E_l \rho E_l^\dagger = \sum_j F_j \rho F_j^\dagger$ for any state ρ. Let P denote the principal system with state space $\mathscr{H}_P$ that has orthonormal basis $\{|i_P\rangle\}_i$. Now, introduce some ancillary system A with state space $\mathscr{H}_A$ of the same size as $\mathscr{H}_P$ and orthonormal basis $\{|i_A\rangle\}_i$. Define the entangled state $|\varphi\rangle \in \mathscr{H}_A \otimes \mathscr{H}_P$ as

$$|\varphi\rangle = \sum_i |i_A\rangle \otimes |i_P\rangle\,.$$

Define furthermore the density operator $\sigma \in \operatorname{End}(\mathscr{H}_A \otimes \mathscr{H}_P)$ that leaves $\mathscr{H}_A$ unchanged and applies $\mathcal{E}$ to $\mathscr{H}_P$, i.e.,

$$\sigma(|\vartheta\rangle) = (I \otimes \mathcal{E})(|\vartheta\rangle\langle\vartheta|), \quad |\vartheta\rangle \in \mathscr{H}_A \otimes \mathscr{H}_P.$$

This is a valid density operator since $\{I \otimes \mathcal{E}\}$ defines a channel on $\mathscr{H}_A \otimes \mathscr{H}_P$ due to I and $\mathcal{E}$ being channels on $\mathscr{H}_A$ and $\mathscr{H}_P$, respectively. Using $|\varphi\rangle$ as the input to this density operator yields

$$\begin{aligned}\sigma(|\varphi\rangle) = (I \otimes \mathcal{E})(|\varphi\rangle\langle\varphi|) &= \sum_{i,j} (I \otimes \mathcal{E})(|i_A\rangle \otimes |i_P\rangle)(\langle j_A| \otimes \langle j_P|) = \sum_{i,j} (|i_A\rangle\langle j_A|) \otimes (\mathcal{E}(|i_P\rangle\langle j_P|) \\ &= \sum_{l,i,j} (|i_A\rangle\langle j_A|) \otimes (E_l |i_P\rangle\langle j_P| E_l^\dagger) = \sum_{l,i,j} (|i_A\rangle \otimes (E_l |i_P\rangle))(\langle j_A| \otimes (\langle j_P| E_l^\dagger).\end{aligned}$$

By defining $|e_l\rangle = \sum_i |i_A\rangle \otimes (E_l |i_P\rangle)$, this can be simplified to

$$\sigma(|\varphi\rangle) = \sum_l |e_l\rangle\langle e_l|\,.$$

Analogously, since $\mathcal{E} = \mathcal{F}$, it follows for $\sigma(|\vartheta\rangle) = (I \otimes \mathcal{F})(|\vartheta\rangle\langle\vartheta|)$ and $|f_j\rangle = \sum_i |i_A\rangle \otimes (F_j |i_P\rangle)$ that

$$\sigma(|\varphi\rangle) = \sum_j |f_j\rangle\langle f_j|\,.$$

Hence, σ is generated by both $\{|e_l\rangle\}_l$ and $\{|f_j\rangle\}_j$, which by Theorem 2.17 implies that there exists a unitary transformation $U \in \operatorname{End}(\mathbb{C}^j)$ such that

$$|e_l\rangle = \sum_j u_{lj} |f_j\rangle\,. \tag{3.7}$$

To show how the sets of Kraus operators are related, consider how E_l acts upon a general state $|\psi\rangle \in \mathscr{H}_P$. Any general state has the form $|\psi\rangle = \sum_i \psi_i |i_P\rangle$ for $\psi_i \in \mathbb{C}$. Now, define a corresponding state in $\mathscr{H}_A$ as

$|\psi'\rangle = \sum_i \psi_i^* |i_A\rangle$. Then,

$$\begin{aligned}
E_l |\psi\rangle &= \sum_i \psi_i E_l |i_P\rangle = \sum_i \langle\psi'|i_A\rangle E_l |i_P\rangle \overset{(a)}{=} \sum_i (\langle\psi'|i_A\rangle) \otimes (E_l |i_P\rangle) \\
&= \sum_i (\langle\psi'| \otimes I)(|i_A\rangle \otimes (E_l |i_P\rangle)) = (\langle\psi'| \otimes I) |e_l\rangle \overset{(b)}{=} \sum_j u_{lj} (\langle\psi'| \otimes I) |f_j\rangle \\
&= \sum_j u_{lj} \left[\left(\left(\sum_i \psi_i \langle i_A| \right) \otimes I \right) \left(\sum_k |k_A\rangle \otimes (F_j |k_P\rangle) \right) \right] = \sum_{i,j,k} u_{lj} (\psi_i \langle i_A|k_A\rangle) \otimes (F_j |k_P\rangle) \\
&\overset{(c)}{=} \sum_j u_{lj} \left(\sum_i \psi_i F_j |i_P\rangle \right) = \sum_j u_{lj} F_j |\psi\rangle \,,
\end{aligned}$$

where the Kronecker product in (a) is inserted for clarity, (b) follows by inserting (3.7), and (c) from $\{|i_A\rangle\}_i$ being an orthonormal basis for $\mathscr{H}_A$. This implies that $E_l = \sum_j u_{lj} F_j$ as desired.

Conversely, assume that there exists a unitary $U \in \mathrm{End}(\mathbb{C}^n)$ such that $E_i = \sum_j u_{ij} F_j$. Then for any $\rho \in \mathrm{End}(\mathscr{H}_P)$, it follows that

$$\mathcal{E}(\rho) = \sum_i E_i \rho E_i^\dagger = \sum_{i,j,l} u_{ij} u_{il}^* F_j \rho F_l^\dagger = \sum_{j,l} \left(\sum_i u_{ij} u_{il}^* \right) F_j \rho F_l^\dagger \overset{(a)}{=} \sum_{j,l} \delta_{jl} F_j \rho F_l^\dagger = \sum_j F_j \rho F_j^\dagger = \mathcal{F}(\rho),$$

where (a) follows from U being unitary. This shows that the channels are identical, which concludes the proof. ■

Since there are many different sets of Kraus operators that can be used to describe a channel by Theorem 3.4, it would be beneficial to determine a bound for how few of such Kraus operators one need for a given channel. An upper bound on this is now proven.

Theorem 3.5: Number of Kraus Operators Needed
Let a channel $\mathcal{E}$ defined on a state space $\mathscr{H} = \mathbb{C}^d$. The channel can then be described by at most d^2 Kraus operators $\{E_i\}_i$, i.e.,

$$\mathcal{E}(\rho) = \sum_{i=1}^{M} E_i \rho E_i^\dagger$$

for $1 \leq M \leq d^2$. [NC10, p. 373]

Proof
Let $\mathcal{E}$ be defined by the M Kraus operators $\{E_1, \ldots, E_M\}$ where $E_i \in \mathrm{End}(\mathbb{C}^d)$ for $i = 1, \ldots, M$. At most d^2 of the M Kraus operators can be linearly independent. Assume that the Kraus operators are arranged such that $\{E_1, \ldots, E_{d^2}\}$ are linearly independent. Then for $k = d^2 + 1, \ldots, M$, it follows that E_k is a linear combination of the first d^2 Kraus operators, i.e.,

$$E_k = \sum_{j=1}^{d^2} c_{kj} E_j, \quad c_{kj} \in \mathbb{C}. \tag{3.8}$$

Now, define the operator $W \in \mathrm{End}(\mathbb{C}^M)$ such that $w_{ij} = \mathrm{Tr}\left(E_i^\dagger E_j\right)$. It then follows that

$$w_{ij} = \mathrm{Tr}\left(E_i^\dagger E_j\right) = \mathrm{Tr}\left((E_j^\dagger E_i)^\dagger\right) = \left(E_j^\dagger E_i\right)^\dagger = \left(\mathrm{Tr}\left(E_j^\dagger E_i\right)\right)^* = w_{ji}^*,$$

which implies that W is Hermitian. Let the k-th columns of W be denoted $|w_k\rangle$, which by using the decimal representation of binary numbers can be written on the form

$$|w_k\rangle = \sum_{i=1}^{M} w_{ik} |i-1\rangle = \sum_{i=1}^{M} \mathrm{Tr}\left(E_i^\dagger E_j\right) |i-1\rangle \,.$$

The k-th column for $k = d^2 + 1, \dots, M$ therefore satisfies

$$|w_k\rangle = \sum_{i=1}^{M} \mathrm{Tr}\Big(E_i^\dagger E_k\Big) |i-1\rangle \overset{(a)}{=} \sum_{i=1}^{M} \mathrm{Tr}\left(E_i^\dagger \left(\sum_{j=1}^{d^2} c_{kj} E_j\right)\right) |i-1\rangle \overset{(b)}{=} \sum_{i=1}^{M}\sum_{j=1}^{d^2} c_{kj}\,\mathrm{Tr}\Big(E_i^\dagger E_j\Big) |i-1\rangle$$
$$= \sum_{j=1}^{d^2} c_{kj} \left(\sum_{i=1}^{M} \mathrm{Tr}\Big(E_i^\dagger E_j\Big) |i-1\rangle\right) = \sum_{j=1}^{d^2} c_{kj} |w_j\rangle ,$$

where (a) follows from inserting (3.8), and (b) from the linearity of the trace. This implies that the k-th column of W for $k = d^2 + 1, \dots, M$ is a linear combination of the first d^2 columns of W, which in turn imply that $\mathrm{rank}(W) \leq d^2$. Since W is Hermitian with rank at most d^2, it follows from the spectral theorem that there exists a unitary operator $U \in \mathrm{End}(\mathbb{C}^M)$ such that $UWU^\dagger$ is diagonal with at most d^2 non-zero entries. This U can then be by Theorem 3.4 be used to define another set of Kraus operators $\{F_i\}_i$ that describes the channel $\mathcal{E}$. Explicitly,

$$F_i = \sum_{j=1}^{M} u_{ij} E_j,$$

wherefrom it follows that at most d^2 operators in $\{F_i\}_i$ are non-zero due to U being diagonal with at most d^2 non-zero entries, which completes the proof. ■

The operator-sum representation of channels is convenient since it enables one to describe the channel on a state space $\mathscr{H}^d$ with at most d^2 Kraus operators in $\mathrm{End}(\mathbb{C}^d)$ even though the environment may that affects the system may be significantly larger.

Now that the theory of quantum channels have been presented, some important examples of such channels are presented in the following sections.

3.3.2 Pauli and Depolarising Channels

Some of the simplest, yet most important, channels in quantum computing acting on a single qubit are defined by the Pauli matrices introduced in Definition 2.9. In Example 2.10, X were seen to act as a bit flip operator, Z as a phase flip operator, and Y as a bit-phase flip operator. Using this intuition, the channels can be defined.

Similarly to how there exists bit flip channels for classical bits, e.g., the binary symmetric channel, a bit flip channel also exists for qubits. Since X is a bit flip operator for a qubit, the bit flip channel with crossover probability p can therefore be described by the Kraus operators

$$E_0 = \sqrt{1-p}I, \quad E_1 = \sqrt{p}X = \sqrt{p}\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}. \tag{3.9}$$

In other words, by using the general form for a density operator given in (3.2), the state of a qubit after being exposed to the bit flip, $\mathcal{E}_X$ channel is

$$\mathcal{E}_X(\rho) = (\sqrt{1-p}I)\rho(\sqrt{1-p}I)^\dagger + (\sqrt{p}X)\rho(\sqrt{p}X)^\dagger = (1-p)\rho + pX\rho X$$
$$= 1-p\begin{bmatrix} \rho_{00} & \rho_{01} \\ \rho_{01}^* & 1-\rho_{00} \end{bmatrix} + p\begin{bmatrix} 1-\rho_{00} & \rho_{01}^* \\ \rho_{01} & \rho_{00} \end{bmatrix},$$

which more clearly illustrate the crossover probability being p. This generally implies that even if ρ was a pure state, then $\mathcal{E}_X(\rho)$ is mixed, which hold for any channel.

Since qubits are characterised by the relative phase between the basis states, this phase can also be corrupted by noise. Particularly, the phase can be flipped. Since Z described such a phase flip, the phase flip channel with crossover probability p can be described by the Kraus operators.

$$E_0 = \sqrt{1-p}I, \quad E_1 = \sqrt{p}Z = \sqrt{p}\begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}.$$

Lastly, a qubit can also be corrupted by both a bit flip and a phase flip, i.e., a bit-phase flip. Using Y as the bit-phase flip operator, the bit-phase flip channel with crossover probability p can be described by the Kraus operators

$$E_0 = \sqrt{1-p}I, \quad E_1 = \sqrt{p}Y = \sqrt{p}\begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}.$$

In practice, the Pauli channels introduced above are often to simplistic to describe channels that occur in quantum computing. They can, however, be 'combined' to yield the so-called depolarising channel, $\mathcal{E}_D$, which with crossover probability p can be described by the Kraus operators

$$E_0 = \sqrt{1-p}I, \quad E_1 = \sqrt{p/3}X, \quad E_2 = \sqrt{p/3}Y, \quad E_3 = \sqrt{p/3}Z.$$

Thus, the depolarising channel leaves ρ as it is with probability $1-p$, and applies either of the Pauli matrices with probability $p/3$. The impact of the channel is more clear by using another representation than the the Kraus operators above. By writing out the state after the channel has been applied to the state similarly to (3.9), the state after being transmitted through the depolarising channel is

$$\begin{aligned}
\mathcal{E}_D(\rho) &= (1-p)\rho + \frac{p}{3}(X\rho X + Y\rho Y + Z\rho Z) \\
&= (1-p)\begin{bmatrix} \rho_{00} & \rho_{01} \\ \rho_{01}^* & 1-\rho_{00} \end{bmatrix} + \frac{p}{3}\left(\begin{bmatrix} 1-\rho_{00} & \rho_{01}^* \\ \rho_{01} & \rho_{00} \end{bmatrix} + \begin{bmatrix} 1-\rho_{00} & -\rho_{01}^* \\ -\rho_{01} & \rho_{00} \end{bmatrix} + \begin{bmatrix} \rho_{00} & -\rho_{01} \\ -\rho_{01}^* & 1-\rho_{00} \end{bmatrix}\right) \\
&= \begin{bmatrix} \frac{2p}{3} + \left(1-\frac{4p}{3}\right)\rho_{00} & \left(1-\frac{4p}{3}\right)\rho_{01} \\ \left(1-\frac{4p}{3}\right)\rho_{01}^* & 1-\frac{2p}{3}-\left(1-\frac{4p}{3}\right)\rho_{00} \end{bmatrix} = \frac{2p}{3}I + \left(1-\frac{4p}{3}\right)\rho \overset{(a)}{=} p'\frac{I}{2} + (1-p')\rho,
\end{aligned}$$

where (a) follows by letting $p' = (4p)/3$. With this parametrisation, the depolarising channel is seen to take the input state to the maximally mixed state with probability p', and leave it be with probability $1-p'$. In other words, the information of the state is lost with probability p'. Hence, the importance of the depolarising channel is that it in this sense is the worst possible noise, however it is not particularly realistic in practice. It therefore often serves as the model for the worst case scenario error.

In order to see the effect that channels has on a qubit, an example follows.

Example 3.6: Bit Flip Channel

Assume that two initial states of a two-level system are given as

$$|\psi_1\rangle = \frac{1}{2}(|0\rangle + \sqrt{3}\,|1\rangle), \quad |\psi_2\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle).$$

Assume now that they are sent through a bit flip channel with crossover probability p. In order to follow the operator-sum representation of channels, the intial states are described by density operators, i.e.,

$$\rho_1 = |\psi_1\rangle\langle\psi_1| = \frac{1}{4}\begin{bmatrix} 1 & \sqrt{3} \\ \sqrt{3} & 3 \end{bmatrix}, \quad \rho_2 = |\psi_2\rangle\langle\psi_2| = \frac{1}{2}\begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}.$$

After ρ_1 is sent through the channel, the state is

$$\mathcal{E}_X(\rho_1) = (1-p)\rho_1 + pX\rho_1 X = \frac{1-p}{4}\begin{bmatrix} 1 & \sqrt{3} \\ \sqrt{3} & 3 \end{bmatrix} + \frac{p}{4}\begin{bmatrix} 3 & \sqrt{3} \\ \sqrt{3} & 1 \end{bmatrix}.$$

This corresponds to the density operator being defined by the ensemble $\{(1-p, |\psi_1\rangle), (p, X\,|\psi_1\rangle)\}$ as expected. If the initial state instead is ρ_2, the outcome of the channel is

$$\mathcal{E}_X(\rho_2) = (1-p)\rho_2 + pX\rho_2 X = (1-p)\rho_2 + p\rho_2 = \rho_2.$$

Hence, when ρ_2 is sent though a bit flip channel, the state does not change.

Some fairly simple channels with scaled Pauli matrices as Kraus operators have now been defined. However, physical phenomena such as decoherence can also be described as channels as decoherence is the result of the principal system interacting with the environment.

3.3.3 Decoherence

As described in Section 2.2, decoherence is the process of a quantum system losing its quantum information through interaction with the environment. Although it is a complex phenomenon, decoherence can be modelled by two processes; loss of information due to loss of energy, i.e., energy dissipation, and loss of quantum information without loss of energy. These processes are described by the channels known as amplitude damping and phase damping, respectively. These channels are now considered separately based on [NC10, ch. 8.3.5-8.3.6].

Amplitude Damping

Every excited two-level quantum system may dissipate energy such that it decays into its ground state analogously to how an excited atom may spontaneously emits a photon in order to return to its ground state. Although the features of the dissipation is different for each system, they can modelled in the same manner. Without loss of generality, assume that the environment is in its ground state $|0\rangle_e$. For simplicity, assume now that the principal system is in either its ground state, $|0\rangle$, or its excited state, $|1\rangle$. If it is in its ground state, no energy dissipation can happen, hence the state of the composite system satisfies the mapping

$$|0\rangle|0\rangle_e \mapsto |0\rangle|0\rangle_e .$$

On the other hand, if the principal system is in its excited state, then it dissipates energy with some probability, say γ, such that the composite system satisfies

$$|1\rangle|0\rangle_e \mapsto \sqrt{1-\gamma}\,|1\rangle|0\rangle_e + \sqrt{\gamma}\,|0\rangle|1\rangle_e .$$

For a general qubit $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$, using the above mappings yields

$$|\psi\rangle|0\rangle_e \mapsto \Big(\alpha|0\rangle + \beta\sqrt{1-\gamma}\,|1\rangle\Big)|0\rangle_e + \beta\sqrt{\gamma}\,|0\rangle|1\rangle_e .$$

This means that the interaction with the environment transforms the state of the principal system to decohere into either state

$$|\psi_0\rangle = \frac{\alpha|0\rangle + \beta\sqrt{1-\gamma}\,|1\rangle}{\sqrt{|\alpha|^2 + |\beta|^2(1-\gamma)}} = \frac{\alpha|0\rangle + \beta\sqrt{1-\gamma}\,|1\rangle}{\sqrt{1-|\beta|^2\gamma}}$$

with probability $1-|\beta|^2\gamma$, or state

$$|\psi_1\rangle = \frac{\beta\sqrt{\gamma}\,|0\rangle}{\sqrt{|\beta|^2\gamma}} = |0\rangle$$

with probability $|\beta|^2\gamma$. Notice that $|\psi_0\rangle$ corresponds to no energy dissipation, which up to a normalisation leaves the amplitude of the ground state unchanged but reduces the amplitude of the excited state due to not having dissipated energy, which essentially corresponds to the principal system being in its excited state is less likely. The $|\psi_1\rangle$ state corresponds to the principal state having dissipated energy to the environment such that its excited state is changed to the ground state. Combining these two possible outcomes, the state after being prone to this channel is more likely to be in the ground state than it was prior to the channel. More rigorously, if the input to the channel is $\rho = |\psi\rangle\langle\psi|$, then the output of the channel is described by the ensemble $\{(1-|\beta|^2\gamma, |\psi_0\rangle), (|\beta|^2\gamma, |\psi_1\rangle)\}$. Explicitly, the amplitude damping channel, $\mathcal{E}_{AD}$, must satisfy the mapping

$$\rho = |\psi\rangle\langle\psi| \mapsto \mathcal{E}_{AD}(\rho) = (1-|\beta|^2\gamma)\,|\psi_0\rangle\langle\psi_0| + |\beta|\gamma\,|\psi_1\rangle\langle\psi_1| .$$

Writing it in matrix form yields

$$\begin{bmatrix} \rho_{00} & \rho_{01} \\ \rho_{01}^* & 1-\rho_{00} \end{bmatrix} \mapsto \begin{bmatrix} \rho_{00} & \rho_{01}\sqrt{1-\gamma} \\ \rho_{01}^*\sqrt{1-\gamma} & (1-\rho_{00})(1-\gamma) \end{bmatrix} + \begin{bmatrix} (1-\rho_{00})\gamma & 0 \\ 0 & 0 \end{bmatrix} .$$

Since the entry of the amplitude corresponding to the state being in its ground state is ρ_{00}, as prior to the channel, in addition to something non-negative, it is clear that the the channel increases the probability of the system being in its ground state at the cost of its excited state. One collection of Kraus operators satisfying the above mapping is

$$E_0 = \begin{bmatrix} 1 & 0 \\ 0 & \sqrt{1-\gamma} \end{bmatrix}, \quad E_1 = \begin{bmatrix} 0 & \sqrt{\gamma} \\ 0 & 0 \end{bmatrix}.$$

The dissipation of energy, γ, has so far been considered as a constant. In practice, the probability of energy dissipation increases exponentially with time, hence it is modelled as $\gamma(t) = 1 - \exp\{-t/T_1\}$, where T_1 is a scalar parameter for the specific system in question known as the relaxation time with respect to dissipation. Since the dissipation is exponentially distribution with parameter $1/T_1$, the expected time before dissipation is T_1. Thus, the relaxation parameter characterises the life-time of a qubit. Some examples of T_1 in some existing quantum computers based on different technologies are listed in Table 3.1.

Quantum Computer	T_1 (μs)
IBM Eagle r3 [Pla]	265.82
IonQ Aria [Sta24]	10^7
Rigetti Ankaa-2 [Rig]	12.9
Google Sycamore [AI21]	21

Table 3.1: Relaxations times, T_1, for different quantum computers.

It should be explicitly noted that since energy dissipation destroys the superposition of the system and thereby the power of quantum computing, long relaxation times are essential for quantum computing. However, long relaxation times often come with the trade-off of more complex technology, e.g., larger hardware, slower operation times, or lower qubit count. For example, the IBM Eagle has 128 qubits whereas the IonQ Aria only has 21.

Phase Damping

A system can also lose information without dissipation of energy, e.g., when a photon travelling through a waveguide scatters randomly. Since it travels through a waveguide, energy from the system is not lost, however information about the evolution of the system is lost and thereby also information regarding the relative phase of the state. The phenomenon is therefore known as dephasing or phase damping. As with energy dissipation, the physical features of dephasing are unique for each system, however they are all modelled by phase kicks. A phase kick is the random change in the relative phase of a qubit, ω, which can be modelled by the rotation operator

$$R(\omega) = \begin{bmatrix} 1 & 0 \\ 0 & \mathrm{e}^{\mathrm{i}\omega} \end{bmatrix}.$$

Assume that ω is Gaussian distributed with zero mean and variance 2τ, i.e., $\omega \sim \mathcal{N}(0, 2\tau)$. The effect of many such phase kicks gives the phase damping channel, $\mathcal{E}_{PD}$, which is found by averaging over the phase kicks ω. For the initial state $\rho = |\psi\rangle \langle\psi|$, that is

$$\mathcal{E}_{PD}(\rho) = \mathbb{E}[R(\omega) |\psi\rangle \langle\psi| R^\dagger(\omega)] = \frac{1}{\sqrt{4\pi\tau}} \int_{-\infty}^{\infty} \exp\left\{-\frac{\omega^2}{4\tau}\right\} R(\omega) |\psi\rangle \langle\psi| R^\dagger(\omega) \,\mathrm{d}\omega.$$

Letting $|\psi\rangle = \alpha |0\rangle + \beta |1\rangle$ and inserting the expression for $R(\omega)$ yields

$$\mathcal{E}_{PD}(\rho) = \frac{1}{\sqrt{4\pi\tau}} \int_{-\infty}^{\infty} \exp\left\{-\frac{\omega^2}{4\tau}\right\} \begin{bmatrix} \rho_{00} & \rho_{01}\mathrm{e}^{-\mathrm{i}\omega} \\ \rho_{01}^*\mathrm{e}^{\mathrm{i}\omega} & 1-\rho_{00} \end{bmatrix} \mathrm{d}\omega.$$

The integrals in the matrix are computed separately. The diagonals are simple as they are merely constants with respect to ω. For the $(1,1)$-th entry, the integral yields

$$\mathcal{E}_{PD}(\rho_{11}) = \frac{1}{\sqrt{4\pi\tau}} \int_{-\infty}^{\infty} \rho_{00} \exp\left\{-\frac{\omega^2}{4\tau}\right\} \mathrm{d}\omega = \rho_{00} \lim_{x\to\infty} F_\omega(x) = \rho_{00},$$

where $F_\omega(x)$ is the cumulative distribution function of ω evaluated at x. Analogously, $\mathcal{E}_{PD}(\rho_{22}) = 1 - \rho_{00}$.

The off-diagonal terms are on the other hand not constant, hence requires a bit more effort. For the $(1, 2)$-th entry, the integral is

$$\begin{aligned}\mathcal{E}_{PD}(\rho_{12}) &= \frac{1}{\sqrt{4\pi\tau}}\int_{-\infty}^{\infty}\rho_{01}\exp\{-\mathrm{i}\omega\}\exp\left\{-\frac{\omega^2}{4\tau}\right\}\mathrm{d}\omega = \frac{\rho_{01}}{\sqrt{4\pi\tau}}\int_{-\infty}^{\infty}\exp\left\{\left(\frac{\mathrm{i}\omega}{2\sqrt{\tau}}-\sqrt{\tau}\right)^2-\tau\right\}\mathrm{d}\omega \\ &= \frac{\rho_{01}\mathrm{e}^{-\tau}}{\sqrt{4\pi\tau}}\int_{-\infty}^{\infty}\exp\left\{-\frac{(\omega+\mathrm{i}2\tau)^2}{4\tau}\right\}\mathrm{d}\omega = \rho_{01}\mathrm{e}^{-\tau}\lim_{y\to\infty}F_Y(y) = \rho_{01}\mathrm{e}^{-\tau},\end{aligned}$$

where $F_Y(y)$ is the cumulative distribution function of $Y \sim \mathcal{N}(-\mathrm{i}2\tau, 2\tau)$. Similarly, $\mathcal{E}_{PD}(\rho_{21}) = \rho_{01}^*\mathrm{e}^{-\tau}$, where the only change in the calculation is a sign flip in the mean of Y. Combining these results yields

$$\mathcal{E}_{PD}(\rho) = \begin{bmatrix}\rho_{00} & \rho_{01}\mathrm{e}^{-\tau} \\ \rho_{01}^*\mathrm{e}^{-\tau} & 1-\rho_{00}\end{bmatrix}.$$

In reality, τ is a real-valued function of time such that the off-diagonal terms of $\mathcal{E}(\rho)$ decays exponentially, i.e., $\tau = t/T$, where T is the decoherence time with respect to dephasing for the system in question. Notice that dephasing also happens in the amplitude damping channel, hence τ depends on both T_1 and T_2, where T_2 is the decoherence time with respect to pure dephasing. Since the off-diagonal entries of the density operator vanishes, the superposition of the state is destroyed and the system merely becomes an ensemble such that it is in its ground state with probability ρ_{00} and its excited state with probability $1 - \rho_{00}$. The channel is typically defined by the Kraus operators

$$E_0 = \begin{bmatrix}1 & 0 \\ 0 & \sqrt{1-\lambda}\end{bmatrix}, \quad E_1 = \begin{bmatrix}0 & 0 \\ 0 & \sqrt{\lambda}\end{bmatrix},$$

where $\lambda(t) = 1 - \exp\{t/T_1 - 2t/T_2\}$ is the probability of dephasing [NH$^+$19, p. 976]. It should be explicitly noted that T_2 also characterises the life-span of a qubit for similar reasons as T_1, where the expression for λ implies that $T_2 \leq 2T_1$ due to λ being a probability that must be non-negative. Analogue to Table 3.1, different T2 values are shown in Table 3.2.

Quantum Computer	T_2 (μs)
IBM Eagle r3 [Pla]	86.42
IonQ Aria [Sta24]	10^6
Rigetti Ankaa-2 [Rig]	12.3
Atom Computing Pheonix [McD22]	4×10^7

Table 3.2: Relaxation times, T_2, for different quantum computers.

In practice does is a qubit prone to both time of decoherence at the same time, hence they are now combined into a new channel that describes decoherence more realistically.

Amplitude-Phase Damping

Combining the amplitude damping channel and the phase damping channel creates a new channel that described decoherence as a whole fairly accurately. This channel is therefore simply referred to as a decoherence channel, but is notated $\mathcal{E}_{APD}$ for amplitude-phase damping. Now, there are several ways to model such a combined channel. One possible parametrisation is by concatenating the channel as illustrated in Figure 3.4.

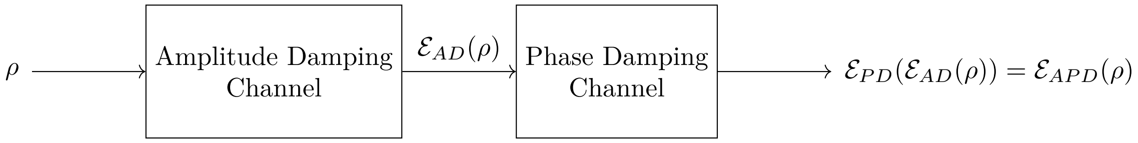

ρ
Amplitude Damping Channel
$\mathcal{E}_{AD}(\rho)$
Phase Damping Channel
$\mathcal{E}_{PD}(\mathcal{E}_{AD}(\rho)) = \mathcal{E}_{APD}(\rho)$

Figure 3.4: Amplitude-Phase damping channel as the serial concatenation of the amplitude damping channel and the phase damping channel.

As the figure illustrates, the decoherence channel is defined as $\mathcal{E}_{APD}(\rho) = (\mathcal{E}_{PD} \circ \mathcal{E}_{AD})(\rho)$. Let the Kraus operators related to $\mathcal{E}_{AD}$ be denoted A_0, A_1, and those related to $\mathcal{E}_{PD}$ be P_0, P_1. By defining the decoherence channel as such leads to following four Kraus operators:

$$\begin{aligned}
E_0 = A_0P_0 &= \begin{bmatrix} 1 & 0 \\ 0 & \sqrt{1-\gamma} \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & \sqrt{1-\lambda} \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & \sqrt{(1-\gamma)(1-\lambda)} \end{bmatrix}, \\
E_1 = A_0P_1 &= \begin{bmatrix} 1 & 0 \\ 0 & \sqrt{1-\gamma} \end{bmatrix} \begin{bmatrix} 0 & 0 \\ 0 & \sqrt{\lambda} \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ 0 & \sqrt{\lambda(1-\gamma)} \end{bmatrix}, \\
E_2 = A_1P_0 &= \begin{bmatrix} 0 & \sqrt{\gamma} \\ 0 & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & \sqrt{1-\lambda} \end{bmatrix} = \begin{bmatrix} 0 & \sqrt{\gamma(1-\lambda)} \\ 0 & 0 \end{bmatrix}, \\
E_3 = A_1P_1 &= \begin{bmatrix} 0 & \sqrt{\gamma} \\ 0 & 0 \end{bmatrix} \begin{bmatrix} 0 & 0 \\ 0 & \sqrt{\lambda} \end{bmatrix} = \begin{bmatrix} 0 & \sqrt{\gamma\lambda} \\ 0 & 0 \end{bmatrix}.
\end{aligned}$$

Now, notice that $E_2 = \sqrt{\frac{1-\lambda}{\lambda}}E_3$, i.e., the two Kraus operators are linearly dependent and it is therefore possible to reduce the number needed to describe the channel. That is, the set of Kraus operators $E = \{E_0, E_1, E_2, E_3\}$ just found describes the same channel as another set $F = \{F_0, F_1, F_2\}$, which cf. Theorem 3.4 can be found by finding a unitary transformation $U \in \mathrm{End}(\mathbb{C}^4)$ relating the two sets. Since E_0, E_1 are linearly independent, they are simply replicated in F, i.e., $F_0 = E_0, F_1 = E_1$, meaning that U has the form

$$U = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ * & * & * & * \\ * & * & * & * \end{bmatrix},$$

where the last two rows are yet to be found. Now, since F must be zero-padded once to get the same cardinality as E, it follows that some weighted sum of the elements of E must be the zero-matrix. One possibility of this $\sqrt{\lambda}E_2 - \sqrt{1-\lambda}E_3 = 0$, which implies that the fourth row of U is $\begin{bmatrix} 0 & 0 & \sqrt{\lambda} & -\sqrt{1-\lambda} \end{bmatrix}$. For the last row to be be orthonormal to this, and the first two rows, one can simply pick the vector $\begin{bmatrix} 0 & 0 & \sqrt{1-\lambda} & \sqrt{\lambda} \end{bmatrix}$. This gives the following unitary transformation U relating the two sets of Kraus operators:

$$U = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & \sqrt{1-\lambda} & \sqrt{\lambda} \\ 0 & 0 & \sqrt{\lambda} & -\sqrt{1-\lambda} \end{bmatrix}.$$

This implies that F_2 is given as

$$F_2 = \sqrt{1-\lambda}E_2 + \sqrt{\lambda}E_3 = \begin{bmatrix} 0 & \sqrt{\gamma} \\ 0 & 0 \end{bmatrix}.$$

Hence, instead of using both E_2, E_3 to describe some error that may happen due to decoherence, the simpler operator F_2 can merely be used. Therefore, the set F of Kraus operators is used henceforth. This implies that for $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$, then $\rho = |\psi\rangle\langle\psi|$ is affected by the decoherence channel as follows:

$$\begin{aligned}
\mathcal{E}_{APD}(\rho) &= F_0\rho F_0^\dagger + F_1\rho F_1^\dagger + F_2\rho F_2^\dagger \\
&= \begin{bmatrix} \rho_{00} & \rho_{01}\sqrt{(1-\gamma)(1-\lambda)} \\ \rho_{01}^*\sqrt{(1-\gamma)(1-\lambda)} & (1-\rho_{00})(1-\gamma)(1-\lambda) \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & (1-\rho_{00})\lambda(1-\gamma) \end{bmatrix} + \begin{bmatrix} (1-\rho_{00})\gamma & 0 \\ 0 & 0 \end{bmatrix} \\
&= \begin{bmatrix} \rho_{00} + (1-\rho_{00})\gamma & \rho_{01}\sqrt{(1-\gamma)(1-\lambda)} \\ \rho_{01}^*\sqrt{(1-\gamma)(1-\lambda)} & (1-\rho_{00})(1-\gamma) \end{bmatrix}.
\end{aligned}$$

Inserting the expressions for γ and λ yields

$$\mathcal{E}_D(\rho) = \begin{bmatrix} 1 - (1-\rho_{00})\mathrm{e}^{-t/T_1} & \rho_{01}\mathrm{e}^{-t/T_2} \\ \rho_{01}^*\mathrm{e}^{-t/T_2} & (1-\rho_{00})\mathrm{e}^{-t/T_1} \end{bmatrix},$$

which can be seen to decohere into $|0\rangle$ as t tends to infinity. This should come as no surprise since the amplitude damping channel increases the probability of the system being in its ground state at the cost of the excited state, while the off-diagonal entries tends to zero for the phase damping channel such that the superposition of the state is destroyed but it merely becomes an ensemble of states. By combining these channels into the decoherence channel, it is clear that the system tends towards its ground state.

Having provided decoherence channels that are more realistic in practice than the depolarising channel, it is time to discuss how such channels destroy quantum information. This is considered in the following section, which is based upon [NC10, ch. 9.2.2, 9.3].

3.4 Fidelity

As illustrated in Example 3.6, not all states are corrupted by channels in the same manner. Particularly, some states are heavily corrupted by the channel whereas other states may not even be corrupted at all. Hence, it would be convenient to define a measure in order to quantify how channels corrupt states. In order to do so is it necessary to first define a measure of distance between states, namely the fidelity.

> **Definition 3.7: Fidelity**
> The fidelity of two quantum states $\rho, \sigma \in \mathrm{End}(\mathscr{H})$, is defined as
>
> $$F(\rho,\sigma) = \left(\mathrm{Tr}\left(\sqrt{\sqrt{\rho}\sigma\sqrt{\rho}}\right)\right)^2 .$$
>
> [NC10, p. 409]

It should be noted that fidelity is sometimes also defined without the trace being squared, such as in [NC10], however the convention seems to define fidelity with the square, hence this is done in this thesis.

Since ρ is a density operator, it is positive semi-definite and thus have a unique positive semi-definite square root. This implies that $\sqrt{\rho}\sigma\sqrt{\rho}$ also is positive semi-definite, hence also has a well-defined square root. It is however not a metric since $F(\rho,\rho) = 1$ due to density operators having trace one. Due to the square in the fidelity, it follows that $F(\rho,\sigma) \geq 0$ for any states, where equality holds if ρ and σ are orthogonal. Hence, opposite than typical distance measures, the fidelity of two quantum states is one when the states are identical and zero when they are orthogonal. In other words, a low distance between states corresponds to a good fidelity near one. All in all, fidelity is a well-defined measure, however it is not obvious why this is a good measure of distance. Before considering this, the case where one of the states is generated by a single state vector, e.g., $\rho = |\psi\rangle\langle\psi|$, the expression for fidelity simplifies. To see this, notice that ρ in that case is idempotent due to being a projection, hence

$$\begin{aligned} F(\rho,\sigma) &= \mathrm{Tr}\left(\sqrt{\sqrt{\rho}\sigma\sqrt{\rho}}\right)^2 \overset{(a)}{=} \mathrm{Tr}(\sqrt{\rho\sigma\rho})^2 = \mathrm{Tr}\left(\sqrt{|\psi\rangle\langle\psi|\sigma|\psi\rangle\langle\psi|}\right)^2 \\ &= \langle\psi|\sigma|\psi\rangle\,\mathrm{Tr}\left(\sqrt{|\psi\rangle\langle\psi|}\right)^2 \overset{(b)}{=} \langle\psi|\sigma|\psi\rangle , \end{aligned}$$

where (a) follows from ρ being idempotent and (b) from density operators having trace one. Since ρ only depends on $|\psi\rangle$, the fidelity of such states are often notated $F(|\psi\rangle, \sigma)$. Furthermore, if $\sigma = |\varphi\rangle\langle\varphi|$, this further simplifies to $F(|\psi\rangle, |\varphi\rangle) = |\langle\psi|\varphi\rangle|^2$.

Now that fidelity has been defined, it can be used to measure how much channels corrupt states. Said differently, fidelity can be used to measure how well information is preserved in a channel. If the initial state $|\psi\rangle$ is sent through a channel $\mathcal{E}$, this preservation is measured by the $F(|\psi\rangle, \mathcal{E}(|\psi\rangle\langle\psi|))$. This is however still depending on the initial state, which may or may not be corrupted by the channel, hence it is of more interest to find the worst case scenario for the channel, i.e., which state is least preserved in the channel. This is the minimum fidelity of the channel, which is given as

$$F_{\min} = \min_{|\psi\rangle} F(|\psi\rangle, \mathcal{E}(|\psi\rangle\langle\psi|)).$$

This is naturally dependent on the channel. To illustrate how this is used, a simple example follows.

Example 3.8: Minimum Fidelity of the Bit Flip Channel
Let $|\psi\rangle$ be an arbitrary qubit subject to a bit flip channel, $\mathcal{E}_{BF}$, with crossover probability p. This implies that $\mathcal{E}_{BF}(|\psi\rangle\langle\psi|) = (1-p)|\psi\rangle\langle\psi| + pX|\psi\rangle\langle\psi|X$. Hence, the channel fidelity is

$$\begin{aligned} F(|\psi\rangle, \mathcal{E}_{BF}(|\psi\rangle\langle\psi|)) &= \langle\psi|\mathcal{E}_{BF}(|\psi\rangle\langle\psi|)|\psi\rangle = \langle\psi|((1-p)|\psi\rangle\langle\psi| + pX|\psi\rangle\langle\psi|X)|\psi\rangle \\ &= (1-p)\langle\psi|\psi\rangle\langle\psi|\psi\rangle + p\langle\psi|X|\psi\rangle\langle\psi|X|\psi\rangle = 1-p+p\langle\psi|X|\psi\rangle^2 . \end{aligned}$$

Since the last term is non-negative, the minimum channel fidelity is found by minimising this term over all $|\psi\rangle$. Since X is the bit flip operator, it easily follows that if $|\psi\rangle \in \{|0\rangle, |1\rangle\}$, then the inner product is zero. Thus,

$$F_{\min}(|\psi\rangle, \mathcal{E}_X(|\psi\rangle\langle\psi|)) = 1-p.$$

Thus, a low crossover probability for the channel will yield a good minimum fidelity for the system.

Some of the most fundamental principles in quantum information theory have now been presented. Although this chapter could be more comprehensive, the focus now turns to how quantum information can be preserved by the usage of coding theory.

4 | Introduction to Quantum Coding

So far, different types of errors that may corrupt a qubit during storage or transmission has been considered. The aim of the following chapter is then to introduce methods for encoding quantum states in order to prevent them from such errors. Particularly, coding schemes for the bit flip channel, phase flip channel, and the bit-phase flip channel are considered in this chapter, which is based on [NC10, ch. 10.1].

Recall the scenario presented Section 1.6, where Alice has some information that she wants to encode and transmit to Bob. This scenario now applies to the theory needed for the remainder of this thesis, hence Alice is henceforth uses as the encoder, and Bob as the decoder.

Similarly to how quantum information theory is a generalisation of its classical counterpart, quantum coding theory can be considered a generalisation of its classical counterpart. It is important to remark that there are important differences between the two fields of coding theory, which will become apparent by using classical coding theory as a foundation for quantum coding theory throughout this chapter. The following section gives a simple example of how the classical repetition code can be generalised to a quantum code for the bit flip channel.

4.1 Bit Flip Repetition Code

As it is one of the most basic, yet insightful, examples, consider the case where a bit and a qubit must be communicated through a binary symmetric channel, and a bit flip channel, respectively, both with crossover probability p.

Firstly, per definition, if a bit is sent through a binary symmetric channel without further consideration, the bit is erroneously delivered with probability p. However, in order to prevent the bit from noise, the most straightforward idea is to repeat the input bit, i.e., add redundancy. Repeating the bit once is equivalent to performing the mappings

$$0 \mapsto 00, \quad 1 \mapsto 11,$$

and then sending both copies through the channel. Suppose that 00 is received. If $p < 1/2$, it is more likely that no errors have occurred than two errors, hence it is assumed that 00 was sent, which corresponds to the input being 0. Now, if 01 is received, it is obvious that an error has occurred, however it is impossible to determine whether the first or second bit has been flipped, hence the bit string provides no information. In that case, it must be randomly guessed whether it corresponds to input 0 or 1, which naturally is correct half of the time assuming that the inputs are equally likely to be sent. Hence, this strategy can correct one error half of the times, which implies that the probability of error in the communication, i.e., Bob decodes wrongly, is

$$\frac{1}{2}\operatorname{Pr}(1 \text{ error}) + \operatorname{Pr}(2 \text{ errors}) = \frac{1}{2}(2p(1-p)) + p^2 = p.$$

Hence, repeating the bit once does not make communication more reliable, i.e., increase the probability of successful communication.

To overcome the problem of ambiguity of the outcomes 01 and 10, the bit can be repeated one additional time. That is, performing the mappings

$$0 \mapsto 000, \quad 1 \mapsto 111.$$

When doing so, there are no outputs that are ambiguous and the corresponding input can be inferred by majority voting. As an example, if 001 is received, then either both zeros have been flipped or the one has been flipped. If $p < 1/2$, it is more likely that one error occurred, hence the corresponding input must have been 0. With this strategy, it is therefore possible to correctly infer the input if up to one error happens. Hence, the probability of error in communication is

$$\operatorname{Pr}(2 \text{ errors}) + \operatorname{Pr}(3 \text{ errors}) = 3p^2(1-p) + p^3 = 3p^2 - 2p^3.$$

Whether or not this is an improvement to doing no repetition depends on p. If $3p^2 - 2p^3 < p$, which by some simple algebra can be seen to be satisfied when $p < 1/2$, the probability of error in communication is decreased by this kind of repetition. This does nonetheless come at the cost of having to send three times as many bits every time.

A coding scheme as simple as the repetition code does perhaps not provide much practical use, however it does provide a possible starting point for creating a quantum coding scheme for a qubit. Particularly, like a bit can simply be repeated, it would be desirable to do the same with an unknown qubit. However, this is generally not possible.

Theorem 4.1: The No-cloning Theorem
Let two quantum systems A and B have state space $\mathscr{H}_A, \mathscr{H}_B$, respectively, with $\mathscr{H}_A = \mathscr{H}_B$. Then it is impossible to copy an arbitrary state from A to B. That is, there exists no unitary transformation $U \in \operatorname{End}(\mathscr{H}_A \otimes \mathscr{H}_B)$ such that for every $|\psi\rangle \in \mathscr{H}_A, |\varphi\rangle \in \mathscr{H}_B$, then

$$U(|\psi\rangle|\varphi\rangle) = \mathrm{e}^{\mathrm{i}\theta}|\psi\rangle|\psi\rangle,$$

where θ depends solely on $|\psi\rangle$ and $|\varphi\rangle$. [NC10, p. 532]

Proof
The proof will be shown by contradiction. Assume that U can clone two arbitrary states $|\psi_1\rangle, |\psi_2\rangle \in \mathscr{H}$. That is, U satisfies both

$$U(|\psi_1\rangle \otimes |\varphi\rangle) = \mathrm{e}^{\mathrm{i}\theta_1}|\psi_1\rangle \otimes |\psi_1\rangle, \quad U(|\psi_2\rangle \otimes |\varphi\rangle) = \mathrm{e}^{\mathrm{i}\theta_2}|\psi_2\rangle \otimes |\psi_2\rangle.$$

Taking the inner product of the left-hand sides yields

$$(\langle\psi_1| \otimes \langle\varphi|)U^\dagger U(|\psi_2\rangle \otimes |\varphi\rangle) = (\langle\psi_1|\psi_2\rangle) \otimes (\langle\varphi|\varphi\rangle) = \langle\psi_1|\psi_2\rangle.$$

Doing the same with the right-hand sides yields

$$(\mathrm{e}^{-\mathrm{i}\theta_1}\langle\psi_1| \otimes \langle\psi_1|)(\mathrm{e}^{\mathrm{i}\theta_2}|\psi_2\rangle \otimes |\psi_2\rangle) = \mathrm{e}^{\mathrm{i}(\theta_2-\theta_1)}\langle\psi_1|\psi_2\rangle^2.$$

Combining the results and taking modulus implies that

$$|\langle\psi_1|\psi_2\rangle| = |\langle\psi_1|\psi_2\rangle|^2,$$

which only holds true when $|\langle\psi_1|\psi_2\rangle|$ is either zero or one. The two states, $|\psi_1\rangle, |\psi_2\rangle$, are therefore either orthogonal or linearly dependent, i.e., $|\psi_1\rangle = \mathrm{e}^{\mathrm{i}\theta}|\psi_2\rangle$ due to the unit length condition, by the Cauchy-Schwarz inequality, which is a contradiction to the assumption of them being arbitrary. ■

It should be noted that the no-cloning theorem states that there exists no U capable of copying every possible state, not that copying is impossible. As illustrated in the proof of the theorem, orthogonal states are capable of being copied by the same U. Particularly, classical bits can be copied, as they should, as illustrated in the following example.

Example 4.2: Copying Bits
Consider the case of copying classical bits, i.e., the states $|0\rangle$ and $|1\rangle$. For simplicity, assume that the space it must be copied in has initial state $|0\rangle$. The aim is then to determine $U \in \operatorname{End}(\mathbb{C}^2)$ that enables such copying, i.e., one satisfying

$$U|00\rangle = |00\rangle, \quad U|10\rangle = |11\rangle.$$

It can easily be verified that this can be done with the so-called CNOT gate given as

$$\mathrm{CNOT} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}.$$

This can be generalised to copying additional $|0\rangle$ and $|1\rangle$ states if needed.

As Example 4.2 illustrated, the repetition coding scheme is also possible in the quantum mechanical case, however slightly different. Although the unknown state $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$ cannot be copied cf. the no-cloning theorem, it is possible to copy the basis states as discussed in Example 4.2. Assume that Alice not only possesses $|\psi\rangle$, but also two ancillary qubits in their ground state such that she in total has the state

$$|\psi_0\rangle = |\psi\rangle \otimes |0\rangle^{\otimes 2} = \alpha|000\rangle + \beta|100\rangle .$$

By applying a CNOT gate to the first two qubits, , i.e., applying operators $(\text{CNOT} \otimes I)$, she obtains the state

$$|\psi_1\rangle = (\text{CNOT} \otimes I)|\psi_0\rangle = \text{CNOT}(\alpha|00\rangle + \beta|10\rangle) \otimes |0\rangle = \alpha|000\rangle + \beta|110\rangle .$$

She can do the same for the third qubit to obtain

$$|\psi_2\rangle = \alpha|000\rangle + \beta|111\rangle .$$

Thus, by applying the given CNOT gates, Alice obtains an entangled state, $|\psi_2\rangle$, that contains the same information as $|\psi\rangle$, however, it has been spread over three qubits rather than one. It is therefore said that $|\psi_2\rangle$ is a logical version of $|\psi\rangle$, where for example $|000\rangle$ is the logical version of $|0\rangle$. Logical versions are henceforth denoted with a subscript L, i.e.,

$$|\psi\rangle_L = \alpha|0\rangle_L + \beta|1\rangle_L = \alpha|000\rangle + \beta|111\rangle .$$

As previously discussed, this logical qubit contains the same information as one qubit, however spread across three physical qubits. Thus, Alice has performed an encoding of $|\psi\rangle$ that is analogue to the classical repetition code. The logical qubit can now be transmitted through a bit flip channel, which after performing the above encoding is the 3-fold bit flip channel that is assumed to act independently and identically on the three qubits, i.e., each qubit is essentially sent through its own bit flip channel. Its Kraus operators are given by the 3-fold tensor product of the Kraus operators defining the bit flip channel. Explicitly, these are given as

$$\{\sqrt{(1-p)^3}I, \sqrt{p(1-p)^2}X_i, \sqrt{p^2(1-p)}X_iX_j, \sqrt{p^3}X_1X_2X_3\}, \quad \text{for } i,j \in \{1,2,3\} \text{ such that } i \neq j,$$

where X_i is shorthand for a bit flip on the i'th qubit, e.g., $X_2 = I \otimes X \otimes I$, and tensor product are written with juxtaposition, e.g., $X_1X_2 = X \otimes X \otimes I$ (see the discussion below (6.1) for further details on the notation). Thus, sending the logical qubit through this channel, the outcome is an ensemble containing the following normalised states, i.e., the probabilities have been omitted as they should be clear from how many bit flips that have occurred:

$$\begin{aligned}
&\alpha|000\rangle + \beta|111\rangle , \quad \alpha|100\rangle + \beta|011\rangle , \quad \alpha|010\rangle + \beta|101\rangle , \quad \alpha|001\rangle + \beta|110\rangle , \\
&\alpha|111\rangle + \beta|000\rangle , \quad \alpha|011\rangle + \beta|100\rangle , \quad \alpha|101\rangle + \beta|010\rangle , \quad \alpha|110\rangle + \beta|001\rangle .
\end{aligned}$$

The first row corresponds to zero or one bit flip, while the second row corresponds to two or three bit flips in 'reverse' direction. Notice that this ordering cause the second row to be identical to the first row if the two amplitudes α, β are flipped. Since Bob does not know the amplitudes, he cannot determine distinguish between the two rows, e.g., whether none or three of the qubits have been bit flipped such that the state is in the first column. Said differently, he cannot determine whether the logical state transmitted was $|\psi\rangle_L = \alpha|0\rangle_L + \beta|1\rangle_L$ or $\bar{X}|\psi\rangle_L = \beta|0\rangle_L + \alpha|1\rangle_L$, where $\bar{X}$ is used as the bit flip operator on the logical state as a whole. This implies that if $p < 1/2$, Bob will always assume that at most one error has occurred, hence $|\psi\rangle_L$ is one of the states is the first row. By majority voting, he can then determine which of the qubits that has been bit flipped. However, if more than two of the qubits are bit flipped such that $\bar{X}$ has been applied to $|\psi\rangle_L$, he wrongly assumes that $\bar{X}|\psi\rangle_L$ is the correct state. This is in general a problem, although in the specific case where $|\psi\rangle = (|0\rangle + |1\rangle)/\sqrt{2}$, the bit flip has no impact cf. Example 3.6. Hence, this coding scheme is essentially equivalent to the classical repetition code where at most one error can be corrected.

The next problem is, however, that it generally is impossible to measure the outcome without changing it Postulate 2.11, which naturally destroys the whole idea of transmitting the state in the first place. Nonetheless, it turns out to be possible in this case by performing a so-called syndrome measurement, which can measure each of the outcomes in the first row without changing the outcome. This is done by using the choosing the observables to be the four operators that project onto the span of one of the states in the first row, i.e.,

$$\begin{aligned} P_0 &= |000\rangle\langle 000| + |111\rangle\langle 111|, \quad & P_1 &= |100\rangle\langle 100| + |011\rangle\langle 011|, \\ P_2 &= |010\rangle\langle 010| + |101\rangle\langle 101|, \quad & P_3 &= |001\rangle\langle 001| + |110\rangle\langle 110|, \end{aligned} \tag{4.1}$$

where P_0 corresponds to no errors and P_i corresponds to a bit flip of qubit i for $i = 1, 2, 3$. They are easily seen to satisfy the completeness equation, thus they are valid observables. These projectors are suitable in this setup due to the following reasoning: Say that the first qubit has been flipped during transmission such that $\mathcal{E}_X(|\psi\rangle_L \langle\psi|_L) = \alpha|100\rangle + \beta|011\rangle$. Performing the syndrome measurement then yields

$$\Pr(1) = 1, \quad \Pr(0) = \Pr(2) = \Pr(3) = 0,$$

meaning that the correct outcome is measured with certainty. Notice that this is a result of the projectors being orthogonal such that the states can be distinguished cf. Theorem 2.13. The state after the syndrome measurement is then $P_1\mathcal{E}_X(|\psi\rangle_L \langle\psi|_L)$, which is the projection of $\alpha|100\rangle + \beta|011\rangle$ onto the span of $|100\rangle + |011\rangle$, which naturally does not change it. Similar arguments holds for all outcomes in the first row. Thus, performing this syndrome measurement, it is possible to know with certainty which error that has occurred, given that at most one happens, without actually inferring anything about the amplitudes and thereby ruining the superposition of $|\psi\rangle_L$. After the error has been detected, it must naturally also be corrected. This can simply be done by flipping the qubit that has been corrupted.

To see whether this bit flip repetition coding scheme actually improves the probability of successful communication, the minimum channel fidelities with and without applying the coding scheme are compared. The minimum channel fidelity without the coding scheme is cf. Example 3.8 given as $F_{\min}(|\psi\rangle, \mathcal{E}_X(|\psi\rangle\langle\psi|)) = 1 - p$. With the coding scheme, there are two possibilities of outcome of the channel. If up to one error occurs, it is corrected to $|\psi\rangle_L$. Otherwise, it is wrongly flipped to $\bar{X}|\psi\rangle_L$. This implies that the outcome of the channel with the coding scheme, denoted Φ_X, is given as

$$\Phi_X(|\psi\rangle_L \langle\psi|_L) = ((1-p)^3 + 3p(1-p)^2)\,|\psi\rangle_L \langle\psi|_L + (3p^2(1-p) + p^3)\bar{X}\,|\psi\rangle_L \langle\psi|_L\,\bar{X}^\dagger.$$

This implies that the fidelity of the channel with the coding scheme is given as

$$\begin{aligned} F(|\psi\rangle_L, \Phi_X(|\psi\rangle_L \langle\psi|_L)) &= ((1-p)^3 + 3p(1-p)^2)\,\langle\psi|_L\,|\psi\rangle_L\,\langle\psi|_L\,|\psi\rangle_L \\ &\quad + (3p^2(1-p) + p^3)\,\langle\psi|_L\,\bar{X}\,|\psi\rangle_L\,\langle\psi|_L\,\bar{X}^\dagger\,|\psi\rangle_L \\ &= (1-p)^3 + 3p(1-p)^2 + (3p^2(1-p) + p^3)\big|\langle\psi|_L\,\bar{X}\,|\psi\rangle_L\big|^2 \\ &= (1-p)^3 + 3p(1-p)^2 + (3p^2(1-p) + p^3)|\alpha^*\beta + \alpha\beta^*|^2. \end{aligned}$$

Since only the last term depends on $|\psi\rangle_L$, and thus $|\psi\rangle$, and that it is non-negative, the minimum fidelity is obtained by minimising the last term. This is clearly obtained if $|\psi\rangle$ is either $|0\rangle$ or $|1\rangle$ such that one of the amplitudes is zero, hence

$$F_{\min}(|\psi\rangle_L, \Phi_X(|\psi\rangle_L \langle\psi|_L)) = (1-p)^3 + 3p(1-p)^2 = 1 - 3p^2 + 2p^3.$$

Comparing the two fidelities, which is visually done in Figure 4.1, gives that the bit flip repetition code improves that probability of successful communication given that $p < 1/2$, which is the same conclusion that was drawn for the classical repetition code over the binary symmetric channel. Hence, the seemingly large problems of creating a code scheme for quantum states, e,g, the no-cloning theorem, measurements generally destroy the state, and the fact that not all states are affected by the channel similarly, turns out not to make quantum error correction impossible.

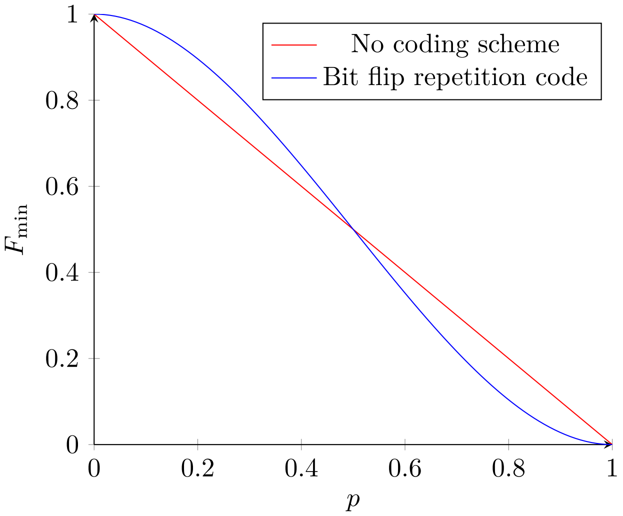


Figure 4.1: Minimum fidelity for the bit flip channel as a function of p with the bit flip repetition code (blue) and without any error correction scheme (red).

The foundation for quantum error correction has now been examined by expanding the repetition code for a binary symmetric channel to a bit flip repetition code for the bit flip channel. However, more types of error may corrupt quantum states, e.g., phase flips.

4.2 Phase Flip Repetition Code

Since there is no classical counterpart to the phase flip channel, creating a coding scheme for such errors can seemingly not be done with classical coding theory. However, if it is possible to find a change of basis of Z such that it turns into X, then the phase flip channel acts as a bit flip channel in that basis, and the coding scheme developed in the previous section can then be utilised. Hence, it would be desirable to determine an operator, which must be unitary in order to keep the unit length of quantum states, $U \in \operatorname{End}(\mathbb{C}^2)$ that satisfies $X = U^\dagger Z U$. Since U must be unitary, it can be written on the form

$$U = \begin{bmatrix} a & b \\ e^{i\theta} b^* & -e^{i\theta} a^* \end{bmatrix}, \quad a, b, \in \mathbb{C} \text{ s.t. } |a|^2 + |b|^2 = 1.$$

Using this, the requirement that $X = U^\dagger Z U$ can be written as

$$\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} = \begin{bmatrix} |a|^2 - |b|^2 & 2a^* b \\ 2ab^* & |b|^2 - |a|^2 \end{bmatrix}.$$

The diagonal entries of the equation implies that $|a| = |b|$, which from the requirement of U being unitary implies that $|a| = |b| = 1/\sqrt{2}$. The off-diagonal entries implies that $a^* b$ and ab^* are real and has amplitude $1/2$. Writing these two conditions out yield

$$\operatorname{Re}(a)\operatorname{Re}(b) + \operatorname{Im}(a)\operatorname{Im}(b) = \frac{1}{2}, \quad \operatorname{Re}(a)\operatorname{Im}(b) = \operatorname{Im}(a)\operatorname{Re}(b).$$

The simplest solutions to these equations is by letting $a = b = \frac{1}{\sqrt{2}}$. This implies that the unitary transformation is the Hadamard matrix given as

$$H = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix},$$

where the columns denote the two states $|+\rangle = (|0\rangle + |1\rangle)/\sqrt{2}$ and $|-\rangle = (|0\rangle - |1\rangle)/\sqrt{2}$, respectively. Hence, a change of basis from the $\{|0\rangle, |1\rangle\}$ basis to the $\{|+\rangle, |-\rangle\}$ basis transform the phase flip channel to a bit flip channel. Using this, a phase flip repetition code can be created by simply applying the

Hadamard transformation at appropriate places in the bit flip repetition code. Explicitly, the qubit $|\psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle$ is encoded into a logical qubit as in the bit flip repetition code to obtain $|\psi\rangle_L$, whereafter the Hadamard transformation is applied to each qubit to obtain a logical qubit, $|\psi'\rangle_L$ for the phase flip repetition code. That is,

$$|\psi'\rangle_L = H^{\otimes 3}\,|\psi\rangle_L = H^{\otimes 3}(\alpha\,|000\rangle + \beta 111) = \alpha\,|+++\rangle + \beta\,|---\rangle\,. \tag{4.2}$$

Hence, in this code $|+++\rangle$ is used as the logical ground state, while $|---\rangle$ is the logical excited state. This implies that $|\psi'\rangle_L$ can be sent through a phase flip channel such that it in the $\{|+++\rangle\,, |---\rangle\}$ basis acts as it is sent through a bit flip channel. Hence, the syndrome measurements used for the bit flip repetition code can be used again, however in the new basis. The projection operators defined in (4.1) are therefore reused by applying the Hadamard transformation on both sides, yielding

$$\begin{aligned}
\tilde{P}_0 &= H^{\otimes 3} P_0 H^{\otimes 3} = H^{\otimes 3}(|000\rangle\langle 000| + |111\rangle\langle 111|)H^{\otimes 3} = |+++\rangle\langle +++| + |---\rangle\langle ---|\,,\\
\tilde{P}_1 &= H^{\otimes 3} P_1 H^{\otimes 3} = H^{\otimes 3}(|100\rangle\langle 100| + |011\rangle\langle 011|)H^{\otimes 3} = |-++\rangle\langle -++| + |+--\rangle\langle +--|\,,\\
\tilde{P}_2 &= H^{\otimes 3} P_2 H^{\otimes 3} = H^{\otimes 3}(|010\rangle\langle 010| + |101\rangle\langle 101|)H^{\otimes 3} = |+-+\rangle\langle +-+| + |-+-\rangle\langle -+-|\,,\\
\tilde{P}_3 &= H^{\otimes 3} P_3 H^{\otimes 3} = H^{\otimes 3}(|001\rangle\langle 001| + |110\rangle\langle 110|)H^{\otimes 3} = |++-\rangle\langle ++-| + |--+\rangle\langle --+|\,.
\end{aligned}$$

With these syndrome measurements, a single phase flip can be detected, whereafter it can be corrected by simply flipping the given phase again. Since the phase flip repetition code is obtained from the bit flip repetition code by a suitable change of basis, they codes are said to be unitarily equivalent, which implies that their performances are equivalent, e.g., they have the same minimum fidelity.

Now that a coding scheme for both the bit flip channel and the phase flip channel has been examined, it may be possible to generalise these to the bit-phase flip channel.

4.3 The Shor Code

In order to create a coding scheme for the phase flip channel, it was related to the bit flip channel. The same idea is now applied to create a coding scheme for the bit-phase flip channel, which, as the name suggest, easily can be related to the bit flip and phase flip channels since

$$Y = \mathrm{i}ZX,$$

where the global phase naturally is irrelevant for the state. Thus, a bit-phase flip is simply a bit flip followed by a phase flip, two error types for which a coding scheme already have been derived. The simplest method to create a coding scheme for the bit-phase flip channel is therefore to simply concatenate the two coding schemes into one, something which was firstly done by Peter Shor in 1995 when he created the first quantum error-correcting code [Sho95]. This code is therefore, rightfully, known as the Shor code. That is, the Shor code, Φ_Y, is obtained by firstly encoding the state of information, $|\psi\rangle$, with the encoding used in the phase flip repetition code, Φ_Z, whereafter it can be further encoded with the bit flip repetition code encoding, Φ_X. Hence, Φ_Y is given as

$$\Phi_Y = (\Phi_X \circ \Phi_Z)(|\psi\rangle\langle\psi|),$$

in which case Φ_Z is called the outer code and Φ_X the inner code. Explicitly, $|\psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle$ is encoded with two ancillary qubits similarly to the phase flip repetition code given in (4.2), which yields the state

$$|\psi'\rangle_L = \alpha\,|+++\rangle + \beta\,|---\rangle\,.$$

Each of the three qubits in the logical qubit $|\psi'\rangle_L$ is then encoded with two ancillary qubits similarly to the bit flip repetition code. In order to see how this is done, $|\psi'\rangle_L$ is firstly written in the canonical basis, yielding

$$\begin{aligned}
|\psi'\rangle_L &= \alpha\left(\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)\right)^{\otimes 3} + \beta\left(\frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)\right)^{\otimes 3}\\
&= \frac{1}{\sqrt{2}}\big[\alpha(|000\rangle + |001\rangle + |010\rangle + |011\rangle + |100\rangle + |101\rangle + |110\rangle + |111\rangle)\\
&\qquad + \beta(|000\rangle - |001\rangle - |010\rangle + |011\rangle - |100\rangle + |101\rangle + |110\rangle - |111\rangle)\big].
\end{aligned}$$

By appending two ancillary qubits after each of these qubits, e.g., the term $\alpha\left|110\right\rangle/\sqrt{2}$ becomes $\alpha\left|100100000\right\rangle/\sqrt{2}$, the encoding used in the bit flip repetition code can be used on the first three qubits, the subsequent three qubits, and the last three qubits separately. Since this encoding simply copies the state of the first qubit to the other qubits, the state corresponding to $\alpha\left|110\right\rangle/\sqrt{2}$ becomes $\alpha\left|111111000\right\rangle/\sqrt{2}$. The full encoded state can then be written fairly compactly as

$$\left|\psi\right\rangle_L = \alpha\left(\frac{1}{\sqrt{2}}(\left|000\right\rangle + \left|111\right\rangle)\right)^{\otimes 3} + \beta\left(\frac{1}{\sqrt{2}}(\left|000\right\rangle - \left|111\right\rangle)\right)^{\otimes 3},$$

which is an entangled nine qubit state. Said differently, it requires nine qubits to encode a single qubit of information with the Shor code. Due to its construction, which is illustrated in Figure 4.2, it is nonetheless useful to consider it as three blocks of three qubits.

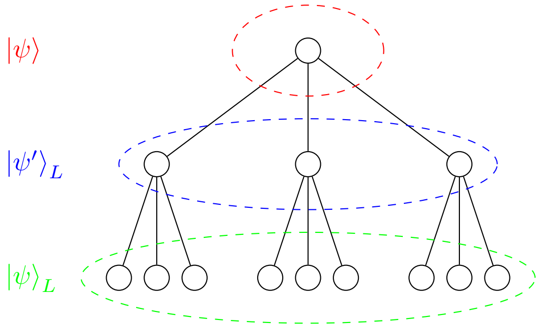

$\left|\psi\right\rangle$
$\left|\psi'\right\rangle_L$
$\left|\psi\right\rangle_L$

Figure 4.2: The concatenation code for the bit-phase flip channel using the phase flip code as outer code and the bit flip code as the inner code. Each node represents a qubit.

In order to illustrate how the Shor code can correct a bit-phase flip, it is firstly considered how it handles bit flips and phase flips separately. This is done by assuming that an error occurs on the first qubit, whereafter similar reasoning can be used to show that it holds for all single qubit errors.

Thus, firstly assume that the first qubit of $\left|\psi\right\rangle_L$ is corrupted by a bit flip, yielding the state

$$X_1\left|\psi\right\rangle_L = \frac{\alpha}{2\sqrt{2}}(\left|100\right\rangle + \left|011\right\rangle)\otimes(\left|000\right\rangle + \left|111\right\rangle)^{\otimes 2} + \frac{\beta}{2\sqrt{2}}(\left|100\right\rangle - \left|011\right\rangle)\otimes(\left|000\right\rangle - \left|111\right\rangle)^{\otimes 2}.$$

Intuitively, since $\left|\psi\right\rangle_L$ always contains either three subsequent zeros or ones, the bit flip can be detected, hence corrected by simply bit flipping the first qubit. More precisely, since the first three qubits, i.e., the first 3-qubit block in Figure 4.2, acts as a logical basis state in the bit flip repetition code, the error is correctable by similar reasoning as given in Section 4.1. This holds true for all of the three 3-qubit blocks, which acts as three independent bit flip repetition codes. It is therefore possible to correct up to three bit flips given that they occur in different blocks, e.g., bit flips on the first and fourth qubits can be corrected, while bit flips on the first and second qubits cannot. Thus, the Shor code outperforms the bit flip repetition with respect to correcting bit flips, however at the cost of a third of the rate.

Now assume that the first qubit of $\left|\psi\right\rangle_L$ is instead corrupted by a phase flip, which yields the state

$$\begin{aligned} Z_1\left|\psi\right\rangle_L &= \frac{\alpha}{2\sqrt{2}}(\left|000\right\rangle - \left|111\right\rangle)\otimes(\left|000\right\rangle + \left|111\right\rangle)^{\otimes 2} + \frac{\beta}{2\sqrt{2}}(\left|000\right\rangle + \left|111\right\rangle)\otimes(\left|000\right\rangle - \left|111\right\rangle)^{\otimes 2} \\ &= \alpha\left|-++\right\rangle + \beta\left|+--\right\rangle. \end{aligned}$$

The error is correctable as since each 3-qubit block acts as a qubit in the phase flip repetition code, which can correct single-qubit error as argued in Section 4.2. Notice that the corrupted state would be the same if the phase flip instead had corrupted either the second or third qubit, i.e., $Z_1\left|\psi\right\rangle_L = Z_2\left|\psi\right\rangle_L = Z_3\left|\psi\right\rangle_L$. This implies that the Shor code can correct up to three phase flips given that they occur in the same block, e.g., phase flips on the first and second qubits can be corrected, while phase flips on the first and fourth cannot. Once again, the Shor code has better performance than the phase flip repetition code with respect to correcting phase flips, but at the cost of a third of the rate.

Lastly, assume the first qubit of $|\psi\rangle_L$ is corrupted by a bit-phase flip such that the state becomes

$$Y_1 |\psi\rangle_L = \frac{\alpha}{2\sqrt{2}}(|100\rangle - |011\rangle) \otimes (|000\rangle + |111\rangle)^{\otimes 2} + \frac{\beta}{2\sqrt{2}}(|100\rangle + |011\rangle) \otimes (|000\rangle - |111\rangle)^{\otimes 2}.$$

By similar reasoning as above, the bit flip can be corrected first, whereafter the phase flip can be corrected. Thus, a single bit-phase flip is correctable with the Shor code. It is, however, not possible to correct more bit-phase flips as bit flips must occur in different blocks to be correctable while phase flips most occur within the same block.

Combining these results indicate that the Shor code can correct every single qubit Pauli error, which in turn imply that it can correct any single-qubit error caused by the depolarising channel. It can furthermore correct some error patterns where more than one qubit is corrected.

However, since states are unit vectors in some Hilbert space, errors are not limited to errors that simply flips the bit or phase (or both), but a continuum of errors. Nonetheless, the Shor code is in fact also capable of correcting such continuous errors, which is shown in the following subsection.

4.3.1 Correction of Continuous Errors

Assume that the qubit $|\psi\rangle = \alpha |0\rangle + \beta |1\rangle$ is encoding with the Shor code such that

$$|\psi\rangle_L = \alpha \left(\frac{1}{\sqrt{2}}(|000\rangle + |111\rangle) \right)^{\otimes 3} + \beta \left(\frac{1}{\sqrt{2}}(|000\rangle - |111\rangle) \right)^{\otimes 3}.$$

If an arbitrary error, E, corrupts the first qubit, say $|0\rangle \mapsto a|0\rangle + b|1\rangle$ and $|1\rangle \mapsto c|0\rangle + d|1\rangle$ for $a, b, c, d \in \mathbb{C}$, the corrupted state has the form

$$\begin{aligned} E|\psi\rangle_L = \frac{1}{2\sqrt{2}} \Big[&\alpha(a|000\rangle + b|100\rangle + c|011\rangle + d|111\rangle) \otimes (|000\rangle + |111\rangle)^{\otimes 2} \\ &+ \beta(a|000\rangle + b|100\rangle - c|011\rangle - d|111\rangle) \otimes (|000\rangle - |111\rangle)^{\otimes 2} \Big]. \end{aligned}$$

By cleverly grouping the terms in this state, it can be written as a sum of errors corresponding to either no error, a bit flip, a phase flip, or a bit-phase flip on the first qubit. Explicitly, the state can up to a normalisation factor be written as

$$\begin{aligned} E|\psi\rangle_L &= \frac{a+d}{2} \left[\alpha(|000\rangle + |111\rangle) \otimes (|000\rangle + |111\rangle)^{\otimes 2} + \beta(|000\rangle - |111\rangle) \otimes (|000\rangle - |111\rangle)^{\otimes 2} \right] \\ &+ \frac{b+c}{2} \left[\alpha(|100\rangle + |011\rangle) \otimes (|000\rangle + |111\rangle)^{\otimes 2} + \beta(|100\rangle - |011\rangle) \otimes (|000\rangle - |111\rangle)^{\otimes 2} \right] \\ &+ \frac{a-d}{2} \left[\alpha(|000\rangle - |111\rangle) \otimes (|000\rangle + |111\rangle)^{\otimes 2} + \beta(|000\rangle + |111\rangle) \otimes (|000\rangle - |111\rangle)^{\otimes 2} \right] \\ &+ \frac{b-c}{2} \left[\alpha(|100\rangle - |011\rangle) \otimes (|000\rangle + |111\rangle)^{\otimes 2} + \beta(|100\rangle + |011\rangle) \otimes (|000\rangle - |111\rangle)^{\otimes 2} \right] \\ &= \frac{1}{2} \left[(a+d)|\psi\rangle_L + (b+c)X_1|\psi\rangle_L + (a-d)Z_1|\psi\rangle_L + (b-c)Y_1|\psi\rangle_L \right]. \end{aligned}$$

Using the decoding procedure for the Shor code, a bit flip can first be detected. By performing a syndrome measurement on the first 3-qubit block with the projectors from the bit flip repetition code given in (4.1), the state is projected onto one of two possible subspaces; the one corresponding to no bit flip on the first qubit or the one corresponding to a bit flip on the first qubit (the other two possibilities are trivially zero is only the first qubit has been corrupted). Hence, although the occurred error, E, does not correspond to either no error or a bit flip, performing the syndrome measurement projects it into the case where it corresponds to one of these. If the outcome of the syndrome measurement is 0 such that the first qubit has not been bit flipped, the state becomes $P_0 E|\psi\rangle_L = \frac{1}{2}[(a+d)|\psi\rangle_L + (a-d)Z_1|\psi\rangle_L]$ up to a normalisation. Said differently, the syndrome measurement for detecting bit flips essentially corrects the part of the occurred error that corresponds to bit flips (amplitudes b, c in E). Analogously, if the outcome is 1, the state is mapped onto a subspace where a bit flip (including bit-phase flip) must have occurred, which already has been shown to be correctable.

Hence, depending on whether or not a bit flip happened, the state can be in either of two states. Without loss of generality, assume that no bit flip occurred such that the state at this point, up to a normalisation, is

$$\begin{aligned}
P_0E\left|\psi\right\rangle_L &= \frac{1}{2}\big[(a+d)\left|\psi\right\rangle_L + (a-d)Z_1\left|\psi\right\rangle_L\big] \\
&= \frac{a+d}{2}\left[\alpha(|000\rangle + |111\rangle)\otimes(|000\rangle + |111\rangle)^{\otimes 2} + \beta(|000\rangle - |111\rangle)\otimes(|000\rangle - |111\rangle)^{\otimes 2}\right] \\
&+ \frac{a-d}{2}\left[\alpha(|000\rangle - |111\rangle)\otimes(|000\rangle + |111\rangle)^{\otimes 2} + \beta(|000\rangle + |111\rangle)\otimes(|000\rangle - |111\rangle)^{\otimes 2}\right].
\end{aligned}$$

This state can now be used to detect a possible phase flip by another syndrome measurement corresponding to that used for the phase flip repetition code. By the same reasoning, it holds that the state is projected onto the state where either no phase flip has occurred or the first qubit has been phase flipped and Bob knows which one. Hence, both cases are correctable.

Without loss of generality, assume that the outcome of the syndrome was 1 such that the state is projected onto the case where the first qubit has been phase flipped. In that case, the state is up to a normalisation factor given as

$$\begin{aligned}
\tilde{P}_1P_0E\left|\psi\right\rangle_L &= \frac{1}{2}(a-d)Z_1\left|\psi\right\rangle_L \\
&= \frac{a-d}{2}\left[\alpha(|000\rangle - |111\rangle)\otimes(|000\rangle + |111\rangle)^{\otimes 2} + \beta(|000\rangle + |111\rangle)\otimes(|000\rangle - |111\rangle)^{\otimes 2}\right].
\end{aligned}$$

By normalising the state, it becomes

$$\tilde{P}_1P_0E\left|\psi\right\rangle_L = \frac{1}{2\sqrt{2}}\frac{a-d}{|a-d|}\left[\alpha(|000\rangle - |111\rangle)\otimes(|000\rangle + |111\rangle)^{\otimes 2} + \beta(|000\rangle + |111\rangle)\otimes(|000\rangle - |111\rangle)^{\otimes 2}\right],$$

which is equivalent to $\left|\psi\right\rangle_L$ since the global phase $\frac{a+d}{|a+d|}$ is unobservable. Hence, the arbitrary error that corrupted the first qubit is correctable. This shows that a continuum of errors can be corrected simply by creating coding schemes for a discrete set of possible errors, namely the Pauli errors.

It is actually possible to deduce the same result by simpler reasoning by considering the error as a result of some channel$\mathcal{E}$ that is described by the Kraus operators $\{E_i\}_i$. Hence, sending the encoded qubit, $\left|\psi\right\rangle_L$, through the channel, the state becomes

$$\mathcal{E}(\left|\psi\right\rangle_L\left\langle\psi\right|_L) = \sum_i E_i\left|\psi\right\rangle_L\left\langle\psi\right|_L E_i^\dagger.$$

By considering only the i-th term in the sum, the Kraus operators restricted to a single qubit can cf. Theorem B.7 be expanded as

$$E_i = a_iI + b_iX + c_iY + d_iZ.$$

The i-th of the corrupted state can therefore be written as as an ensemble of states corresponding to either no error, a bit flip, a phase flip, or a bit-phase flip. Thus, during syndrome measurement, the state collapses to either of these states, which all have been shown to be correctable. Since this holds for all i, an arbitrary error on a single qubit is correctable.

Some fairly simple coding schemes for the bit flip channel, the phase flip channel, and the bit-phase channel have been examined. The importance of these schemes is that they intuitively illustrate that seemingly catastrophic problems with encoding and decoding of quantum states actually are not fatal. More precisely, it has been shown that although quantum states generally cannot be copied cf. the no-cloning theorem, orthogonal states can be copied by creating entangled states. These are created such that it is possible to do measurements without destroying the state, but merely detecting the possible error that has occurred. In fact, such measurements ensures that an arbitrary (continuous) error is correctable by only correcting a discrete set of errors. With these considerations, more general theory for quantum codes can be examined.

5 | General Quantum Coding Theory

Based on the intuition gained in Chapter 4, the aim is to develop a general theory of quantum error-correcting codes, which will serve as the theoretical foundation for considering different types of codes. It is based on [NC10, ch. 10.3].

Before examining general properties of quantum codes, the basic ideas of them are briefly summarised in accordance to the discoveries made in Chapter 4. Even though the theory also holds more generally, it is only presented for qubits here as they are the building blocks for quantum computing.

Consider the case where Alice posses some quantum state of k qubits, $|\psi\rangle \in \mathscr{H}_k = (\mathbb{C}^2)^{\otimes k}$ described by the 2^k orthonormal basis vectors $\{|i\rangle\}_i$. In order to send this through some noisy channel, $\mathcal{E}$, such that it can be recovered by Bob in the presence of noise, it must be encoded. This is done in two steps. Firstly, redundant information is incorporated into the state by adding $n-k$ ancillary qubits prepared in the ground state, $|0\rangle$, such that Alice actually posses the state $|\psi'\rangle \in \mathscr{H}_n = \mathscr{H}_k \otimes \mathscr{H}_{n-k} = (\mathbb{C}^2)^{\otimes n}$ on the form

$$|\psi'\rangle = |\psi\rangle \otimes |0\rangle^{\otimes n-k} .$$

Since $|\psi'\rangle$ contains the same information as $|\psi\rangle$, it is the logical version of it, and is henceforth denoted $|\psi\rangle_L$. Thus, Alice posses the logical state $|\psi\rangle_L \in \mathscr{H}_n$ that is described by the 2^k orthonormal logical basis states $\{|i\rangle_L\}_i$. In other words, $|\psi\rangle_L$ is in a 2^k-dimensional subspace of $\mathscr{H}_n$. Since all the information of $|\psi\rangle_L$ is located in the first k qubits, similarly to $|\psi\rangle$ itself, it is a poorly encoding. In order to make the state more robust to noise, Alice spreads out the information to all n qubits by applying a unitary operator $U \in \operatorname{End}(\mathscr{H})$ that depends on $\mathcal{E}$. For example, if $\mathcal{E}$ is the bit flip channel, Alice can use $n-k$ CNOT gates in order to obtain a generalised version of the bit flip repetition code. The subspace spanned by $\{U|i\rangle_L\}_i$ is called a quantum code, C, such that $U|\psi\rangle_L \in C$. To ease the notation, U is henceforth omitted when referring to the encoded state $|\psi\rangle_L$ such that a logical state must be read as one that is fully encoded. Since C is a 2^k-dimensional subspace of $\mathscr{H}_n$, it is said to be an $[[n,k]]$ code, where the additional bracket is used to distinguish it from the classic notation.

After the encoding, the state $|\psi\rangle_L$ can be transmitted over $\mathcal{E}$ where it possibly is corrupted by some noise. Bob then aims to recover $|\psi\rangle_L$, something typically also done in two steps. In practice, $|\psi\rangle$ must be recovered, which after obtaining $|\psi\rangle_L$ simply can be done by applying $U^\dagger$ and then tracing out the ancillary qubits. It is therefore trivial to obtain $|\psi\rangle$ after obtaining $|\psi\rangle_L$, hence it is generally omitted from the decoding process in this thesis. The first step is to detect the error that possibly has occurred. This is done by a syndrome measurement, which is a projective measurement. It is therefore convenient to introduce the orthogonal projector onto C, henceforth denoted P_C. The syndrome measurement yields an error syndrome, which indicates which error that is most likely to have occurred. For example, a syndrome measurement for the codes described in Chapter 4 will always detect that at most one qubit has been corrupted, assuming that the probability of error is sufficiently low. The error syndrome does therefore not always detect the correct error. In any case, it indicates the error that most likely have occurred, which then must be corrected. This is done by applying some suitable operator, which generally only yields the correct state $|\psi\rangle_L$ if the error is correctly detected. The entire decoding process, i.e., detection and recovery, is typically combined into a single recovery process $\mathcal{R}$, which is assumed to be a quantum channel with Kraus operators $\{R_i\}_i$. The recovery process is therefore commonly known as a correction channel. The basic ides of the setup is illustrated in Figure 5.1 using the density operator notation.

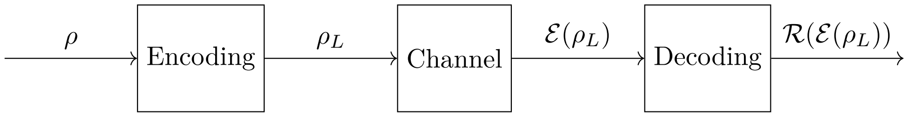


Figure 5.1: Block diagram of the communication system at a high level of abstraction.

The first problem in coding theory is then to determine when there for a given channel exists a correction channel that actually can correct the potential errors.

5.1 Existence of Correction Channel

In order to determine some conditions for the existence of a successful correction channel, it must firstly be clarified what is meant by successful in this regard. Intuitively, a correction channel, $\mathcal{R}$, is said to be successful if it can recover every state in the code with certainty. That is

$$\mathcal{R} \text{ is successful } \iff 1 = \Pr((\mathcal{R} \circ \mathcal{E})(\rho_L) = \rho_L) = \operatorname{Tr}\left(\sum_i R_i \mathcal{E}(\rho_L) R_i^\dagger\right),$$

which is equivalent to requiring that $\mathcal{R}$ is trace-preserving, i.e., $\{R_i\}_i$ satisfy the completeness equation. However, as illustrated in Chapter 4, it is generally impossible to correct all the possible errors of a channel due to the fact that error syndromes only detect the most likely error. Thus, only a subset of possible errors for $\mathcal{E}$ enables the existence of a successful correction channel. Denote such subset $\mathcal{E}_C$. For example, the bit flip repetition code can only correct up to one bit flip, while more bit flips will be wrongly 'corrected'. Thus, $\mathcal{E}_C = \{I, X_1, X_2, X_3\}$ up to their respective probabilities. Therefore, $\mathcal{E}_C$ describes a subset of the Kraus operators for $\mathcal{E}$, $\{E_i\}_i$, which means that it generally is not trace-preserving. This slackens the definition of a successful correction channel to

$$(\mathcal{R} \circ \mathcal{E}_C)(\rho_L) \propto \rho_L.$$

Now that the success of a correction channel has been defined, it is possible to determine some existence conditions for $\mathcal{R}$ that depends on the code C and the possible errors $\mathcal{E}_C$.

Theorem 5.1: Knill-Laflamme Conditions
Let C be a quantum code, P_C the projection onto it, and $\mathcal{E}_C$ some error operators $\{E_i\}_i$. A necessary and sufficient condition for the existence of a correction channel, $\mathcal{R}$, that can correct $\mathcal{E}_C$ on C is that there exists some Hermitian matrix H such that

$$P_C E_i^\dagger E_j P_C = h_{ij} P_C.$$

[NC10, p. 436]

Proof
Sufficiency is proven by a constructive proof, where the correction channel $\mathcal{R}$ contains a syndrome measurement and a recovery operator. So suppose the condition holds with errors $\{E_i\}_i$. Since H is assumed to be Hermitian, the spectral theorem gives that it is unitarily diagonalisable such that $D = U^\dagger H U$, where D is diagonal with non-negative real entries and U is unitary. This U can then be used in Theorem 3.4 to define another set of errors, $\{F_j\}_j$ on the form $F_j = \sum_i u_{ij} E_i$ that also describes $\mathcal{E}_C$. By using these errors on the condition, it holds that

$$\begin{aligned} P_C F_n^\dagger F_m P_C &= P_C \left(\sum_i u_{in} E_i\right)^\dagger \left(\sum_j u_{jm} E_j\right) P_C = \sum_i \sum_j u_{in}^* u_{jm} P_C E_i^\dagger E_j P_C \\ &= \sum_i \sum_j u_{in}^* u_{jm} h_{ij} P_C = \sum_i \sum_j u_{in}^* h_{ij} u_{jm} P_C = d_{nm} P_C. \end{aligned} \tag{5.1}$$

It can therefore without loss of generality be assumed that the errors describing $\mathcal{E}_C$ is chosen such that H is diagonal with non-negative real entries. The aim is now to determine the syndrome measurements. The polar decomposition yields that there for $F_k P_C$ exists some unitary U_k such that

$$F_k P_C = U_k \sqrt{(F_k P_C)^\dagger F_k P_C} = U_k \sqrt{P_C F_k^\dagger F_k P_C} \overset{(a)}{=} U_k \sqrt{d_{kk} P_C} = \sqrt{d_{kk}} U_k P_C. \tag{5.2}$$

where (a) follows from (5.1), and the last from the idempotency of P_C. This implies that the error F_k occurs with probability d_{kk} and acts as a rotation on the subspace C. Notice that d_{kk} may be zero for some k in which case these simply can be ignored. More precisely, for k such that $d_{kk} \neq 0$, F_k rotates C into the subspace given by the orthogonal projection $P_k = U_k P_C U_k^\dagger$. It now easily follows that

$$\begin{aligned} P_l P_k = P_l^\dagger P_k = U_l P_C U_l^\dagger U_k P_C U_k^\dagger &\overset{(a)}{=} U_l P_C \left(\frac{1}{\sqrt{d_{ll}}} F_l\right)^\dagger \left(\frac{1}{\sqrt{d_{kk}}} F_k\right) P_C U_k^\dagger \\ = \frac{1}{\sqrt{d_{ll} d_{kk}}} U_l P_C F_l^\dagger F_k P_C U_k^\dagger &\overset{(b)}{=} \frac{1}{\sqrt{d_{ll} d_{kk}}} U_l (d_{lk} P_C) U_k^\dagger = \begin{cases} P_k, & \text{if } k = l, \\ 0, & \text{if } k \neq l, \end{cases} \end{aligned}$$

where (a) follows from (5.2), (b) from (5.1), and the last equality from the definition of P_k and D being diagonal. This implies that $\{P_k\}_k$ are projections onto orthogonal subspaces such that a projective measurement with observables $\{P_k\}_k$ can distinguish states. Notice that since $I \in \mathcal{E}_C$, then $P_C \in \{P_k\}_k$. However, the projectors must add to the identity to satisfy the completeness equation, which simply can be done by defining an additional observable given as

$$P_I = I - \sum_k P_k$$

with $U_I = I$ such that all projectors still are orthogonal. Now that the syndrome measurements has been defined, the recovery operator must be found. Since the errors simply rotates C into another subspace, the recovery can simply be done by performing the reverse rotation, i.e., applying $U_k^\dagger$. Combining these steps into the correction channel $\mathcal{R}$, it is given as

$$\mathcal{R}(\cdot) = \sum_k U_k^\dagger P_k(\cdot)P_k U_k,$$

where $\sum_k P_k U_k (P_k U_k)^\dagger = \sum_k P_k U_k U_k^\dagger P_k = \sum_k P_k = I$ such that $\mathcal{R}$ is trace-preserving and therefore a valid correction channel. To see that it is successful:

$$\begin{aligned}(\mathcal{R} \circ \mathcal{E}_C)(\rho_L) &= \sum_{i,j} U_j^\dagger P_j F_i \rho_L F_i^\dagger P_j U_j \\ &= \sum_{i,j} U_j^\dagger U_j P_C U_j^\dagger F_i \rho_L F_i^\dagger U_j P_C U_j^\dagger U_j \\ &= \sum_{i,j} P_C \left(\frac{1}{\sqrt{d_{jj}}} F_j\right)^\dagger F_i \rho_L F_i^\dagger \left(\frac{1}{\sqrt{d_{jj}}} F_j\right) P_C \\ &\overset{(a)}{=} \sum_{i,j} \frac{1}{d_{jj}} P_C F_j^\dagger F_i P_C \rho_L P_C F_i^\dagger F_j P_C \\ &\overset{(b)}{=} \sum_{i,j} \frac{1}{d_{jj}} (\delta_{ij} d_{ji} P_C) \rho_L (\delta_{ij} d_{ij} P_C) \\ &= \sum_j d_{jj} P_C \rho_L P_C \\ &\overset{(a)}{\propto} \rho_L,\end{aligned} \tag{5.3}$$

where the first equations follows from the definitions of $\mathcal{R}, \mathcal{E}_C, P_j$, and U_j, (a) from $\rho_L \in C$ hence unaffected by P_C, and (b) from (5.1). This proves that if the condition is satisfied, then there exists some successful correction channel $\mathcal{R}$.

Conversely, to prove necessity, assume that there exists a correction channel $\mathcal{R}$ with Kraus operators $\{R_i\}_i$ capable of correcting all errors in $\mathcal{E}_C$, i.e.,

$$(\mathcal{R} \circ \mathcal{E}_C)(\rho_L) \propto \rho_L,$$

which is equivalent to saying that for all ρ, then

$$(\mathcal{R} \circ \mathcal{E}_C)(P_C \rho P_C) \propto P_C \rho P_C.$$

The latter expression can be written in terms of it Kraus operators as

$$\sum_{i,j} R_i E_j P_C \rho P_C E_j^\dagger R_i^\dagger = k P_C \rho P_C$$

for some constant $k \in \mathbb{C}$. Thus, the entire channel given by the above expression is thus characterised by both the Kraus operators $\{R_i E_j P_C\}_{i,j}$ from the left-hand side and the single Kraus operator $\sqrt{k} P_C$ from the right-hand side. This implies from Theorem 3.4 that there exists some unitary matrix U such that

$$(R_i E_j P_C)_n = u_{n1} \sqrt{k} P_C.$$

Since the scalars $\{u_{n1}\}_n$ only depends on i, j, this is equivalent to saying that there exists some scalars such that

$$R_i E_j P_C = k_{ij} P_C.$$

Since the correction channel $\mathcal{R}$ is trace-preserving, i.e., $\sum_i R_i^\dagger R_i = I$, implies that

$$P_C E_j^\dagger E_l P_C = P_C E_j^\dagger \left(\sum_i R_i^\dagger R_i\right) E_l P_C = \sum_i P_C E_j^\dagger R_i^\dagger R_i E_l P_C \sum_i = (k_{ij}^* P_C)(k_{il} P_C) = h_{jl} P_C,$$

where the last equality follows from defining H such that $h_{jl} = \sum_i k_{ij}^* k_{il}$. Hence, it is easily seen that H is Hermitian, which concludes the proof. ■

Before discussing the Knill-Laflamme conditions further, it should be noted that they immediately infer that errors can be discretised, a property already shown in Section 4.3.1, which is now proven more rigorously.

> **Corollary 5.2: Discretisation of Errors**
> Assume C is a code such that $\mathcal{E}_C$ with Kraus operators $\{E_i\}_i$ is a collection of correctable errors by the correction channel $\mathcal{R}$, which has been constructed as in the proof of the Knill-Laflamme conditions. Let another collection $\mathcal{F}_C$ with Kraus operators $\{F_j\}_j$ be in the span of $\mathcal{E}_C$, i.e., $F_j = \sum_i a_{ji} E_i$ for some complex matrix A. Then $\mathcal{R}$ also correct the errors $\mathcal{F}_C$ on C. [NC10, p. 438]

Proof
By similar reasoning as in the proof of the Knill-Laflamme conditions, it can without loss of generality be assumed that $\{E_i\}_i$ is chosen such that the Hermitian matrix in the Knill-Laflamme condition is diagonal, i.e.,

$$P_C E_i^\dagger E_j P_C = d_{ij} P_C$$

for some real-valued non-negative diagonal matrix D. In that case, the Kraus operators of $\mathcal{R}$ is given by $\{U_k^\dagger P_k\}_k$ as defined in the proof of the Knill-Laflamme conditions. By using that $\mathcal{F}_C$ is in the span of $\mathcal{E}_C$, it follows that

$$\begin{aligned}
(\mathcal{R} \circ \mathcal{F}_C)(\rho_L) &= \sum_{i,j} U_i^\dagger P_i F_j \rho_L F_j^\dagger P_i U_i \\
&= \sum_{i,j} U_i^\dagger P_i \left(\sum_k a_{jk} E_k\right) \rho_L \left(\sum_k a_{jk} E_k\right)^\dagger P_i U_i \\
&= \sum_{i,j,k} |a_{jk}|^2 U_i^\dagger P_i E_k \rho_L E_k^\dagger P_i U_i \\
&\overset{(a)}{=} \sum_{i,j,k} \frac{|a_{jk}|^2}{d_{ii}} (\delta_{ik} d_{ii} P_C) \rho_L (\delta_{ki} d_{ii} P_C) \\
&= \sum_{i,j} |a_{ji}|^2 d_{ii} P_C \rho_L P_C \\
&\propto \rho_L,
\end{aligned}$$

where (a) follows from similar steps as in (5.3). This concludes that $\mathcal{F}_C$ also is correctable by $\mathcal{R}$ as desired. ■

Corollary 5.2 implies two noteworthy things. Firstly, the errors $\{E_i\}_i$ in $\mathcal{E}_C$ can be normalised such that H is a binary matrix. That is, even though the elements of $\mathcal{E}_C$ generally are defined with some probability of occurrence, this probability factor is not necessary to consider when the Knill-Laflamme conditions are tested. Secondly, since the Pauli matrices (including the identity operator) spans the set of operators in $\mathrm{End}(\mathbb{C}^2)$, it is sufficient to consider these errors for a single qubit as previously discussed. As quantum errors in this thesis are assumed to be independent and identically distributed, it is for n qubits sufficient to only consider n-fold Pauli operators. These realisations simplifies the Knill-Laflamme conditions.

With these two simplifications, the Knill-Laflamme conditions can be discussed. For a given code, C, and a set of errors, $\mathcal{E}_C$, the Knill-Laflamme conditions simply yields an easy method for determining the existence of a correction channel $\mathcal{R}$ capable of correcting $\mathcal{E}_C$ on C. It is nonetheless rather time-consuming as it requires an exhaustive process. However, the process can be simplified when C is described by an orthonormal basis $\{|i\rangle_L\}_i$, which typically is the case, such that projection onto C is given as $P_C = \sum_i |i\rangle_L \langle i|_L$. This implies that the Knill-Laflamme conditions can equivalently be written as

$$\begin{aligned}
P_C E_i^\dagger E_j P_C &= h_{ij} P_C \\
\iff \sum_{n,m} |n\rangle_L \langle n|_L E_i^\dagger E_j |m\rangle_L \langle m|_L &= h_{ij} \sum_k |k\rangle_L \langle k|_L \\
\iff \langle n|_L \left(\sum_{n,m} |n\rangle_L \langle n|_L E_i^\dagger E_j |m\rangle_L \langle m|_L \right) |m\rangle_L &= h_{ij} \langle n|_L \left(\sum_k |k\rangle_L \langle k|_L \right) |m\rangle_L \\
\iff \langle n|_L |n\rangle_L \langle n|_L E_i^\dagger E_j |m\rangle_L \langle m|_L |m\rangle_L &= h_{ij} \delta_{nk} \langle n|_L |k\rangle_L \delta_{km} \langle k|_L |m\rangle_L \\
\iff \langle n|_L E_i^\dagger E_j |m\rangle_L &= h_{ij} \delta_{nm},
\end{aligned}$$

where the last two implications follows from the orthonormality of the basis. Thus, to test whether the Knill-Laflamme conditions are satisfied, it is sufficient to only consider the effects of errors on the basis states of C. Not only can this reformulation simplify the verification of the Knill-Laflamme conditions, but it is also geometrically informative, as the following three things can be concluded:

i) In order to satisfy the conditions, orthogonal states must be mapped to orthogonal states no matter which errors that occurs due to δ_{nm}. Thus, if one can find orthogonal states that is mapped to non-orthogonal states, the condition cannot be satisfied, which means that no correction channel exists for all errors in $\mathcal{E}_C$. This should however also be apparent from Theorem 2.13, which states that non-orthogonal cannot be distinguished.

ii) Since the errors $\{E_i\}_i$ without loss of generality can be chosen such that H is diagonal, it is sufficient, but naturally not necessary, to do so. If $\{E_i\}_i$ are chosen as such, then $\langle n|_L E_i^\dagger E_j |m\rangle_L = h_{ij}\delta_{ij}\delta_{nm}$, which implies that different errors must map the basis vectors to orthogonal states, i.e., $\langle n|_L E_i^\dagger E_j |n\rangle_L = 0$ for all $i \neq j$ and $|n_L\rangle$ in the orthonormal basis of C. In that case, a measurement will detect the error whereafter it can be corrected.

iii) In contrast to 2), if the errors $\{E_i\}_i$ are chosen such that H not is diagonal, then it is not necessary that different errors map the basis states to orthogonal states. An example of this is when two phase flips occur in the same block in the Shor code. In that case, e.g., Z_1 and Z_2 corrupt the state identically. This implies that one cannot distinguish which of the errors that have occurred, yet they are both correctable by applying either Z_1 or Z_2. However, Corollary 5.2 gives that it is possible to find errors in the span of $\{Z_1, Z_2\}$ that also are correctable by the Shor code. To find such errors, notice that Z_1 and Z_2 acting similarly on the code implies that H used in the Knill-Laflamme conditions is the operator of all ones in $\operatorname{End}(\mathbb{C}^2)$. Its eigenvalues are zero and two, with eigenvectors $|-\rangle$ and $|+\rangle$ respectively. Hence, H can be diagonalised by applying the Hadamard operator on both sides. From Corollary 5.2, this implies that it is possible to define errors $E_1 = \frac{1}{2}(Z_1 + Z_2), E_2 = \frac{1}{2}(Z_1 - Z_2)$ such that H is diagonal with respect to these. These can be written out explicitly by realising that

$$Z_1 = (Z \otimes I) \otimes I^{\otimes 7} = \begin{bmatrix} 1 & & & \\ & 1 & & \\ & & -1 & \\ & & & -1 \end{bmatrix} \otimes I^{\otimes 7} = \begin{bmatrix} I & & & \\ & I & & \\ & & -I & \\ & & & -I \end{bmatrix},$$

$$Z_2 = (I \otimes Z) \otimes I^{\otimes 7} = \begin{bmatrix} 1 & & & \\ & -1 & & \\ & & 1 & \\ & & & -1 \end{bmatrix} \otimes I^{\otimes 7} = \begin{bmatrix} I & & & \\ & -I & & \\ & & I & \\ & & & -I \end{bmatrix},$$

where $I \in \operatorname{End}(\mathbb{C}^{128})$ in the matrices on the right-hand side. Inserting these expressions into those of E_1, E_2 yields

$$E_1 = \begin{bmatrix} I & & & \\ & 0 & & \\ & & 0 & \\ & & & -I \end{bmatrix}, \quad E_2 = \begin{bmatrix} 0 & & & \\ & I & & \\ & & -I & \\ & & & 0 \end{bmatrix}.$$

Since the Shor code is defined such that the first three qubits out of the nine are either all zero or all one, only the first or last 64 columns of E_1 and E_2 act on the code. In other words, only the first and last block matrix on the diagonals of the four matrices acts on the code. By visual inspection, it can then be seen that E_1 acts similarly as Z_1 on the code, while E_2 acts as an annihilator. This implies for an encoded state $\left|\psi\right\rangle_L$ that

$$\left\langle\psi\right|_L E_1^\dagger E_1 \left|\psi\right\rangle_L = 1, \quad \left\langle\psi\right|_L E_2^\dagger E_2 \left|\psi\right\rangle_L = 0, \quad \left\langle\psi\right|_L E_1^\dagger E_2 \left|\psi\right\rangle_L = 0.$$

In other words, H has been diagonalised as promised, implying that a measurement will detect that the error E_1 has occurred (E_2 occurs with probability 0). It can after detection be recovered by applying either Z_1 or Z_2. Generally, this gives the condition that

$$\left\langle 0\right|_L E_i^\dagger E_i \left|0\right\rangle_L = \left\langle n\right|_L E_i^\dagger E_i \left|n\right\rangle_L$$

for all states $\left|n\right\rangle_L$ in the basis of C such that it is sufficient to consider how E_i affects the logical ground state.

Based on the above discussion, it follows that although different errors may corrupt the code C similarly and thus leading to the same error syndrome, different error syndromes must correspond to orthogonal subspaces in $\mathscr{H}_n$. They subspaces must not only be orthogonal, but the errors must also preserve the orthogonality of the basis states of C. If this is the case, the syndrome measurement will detect some error, which then can be recovered by a unitary operator. Using this, Figure 5.1 can be extended to include how the Knill-Laflamme conditions must be satisfied.

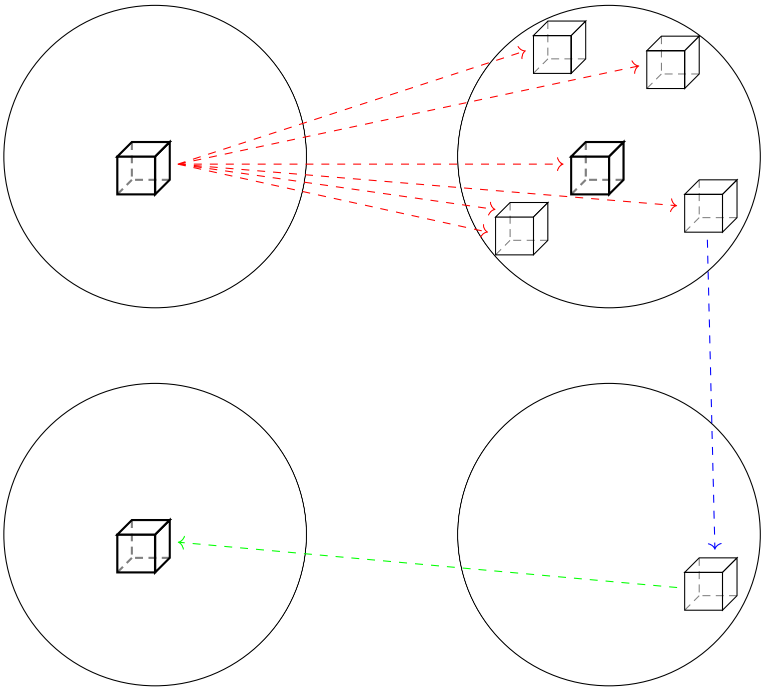


Figure 5.2: Visualisation of the Knill-Laflamme conditions for an $[[n, k]]$ code. The state is encoded into a 2^k-dimensional subspace (the bold square) of $\mathscr{H}_n$ (the circle). The state is then affected by some errors $\mathcal{E}_C$ (red arrows) that maps the code to other 2^k-dimensional subspaces (non-bold squares). A syndrome measurement (blue arrow) then determines an error syndrome, wherefrom a unitary operator (green arrow) can be applied to correct the error given that the Knill-Laflamme conditions are satisfied for $\mathcal{E}_C$.

Codes with the property that several errors act similarly on the code, i.e., the condition in 3), are called degenerate codes. This is also illustrated in Figure 5.2, where the bottom two red arrows (errors) map C to the same subspace. This property is equivalent to H, which were the case for Z_1 and Z_2 for the Shor code. There is no classical codes with this property, hence these codes can in some sense outperform the classical ones. However, this also implies that many classical bound on codes cannot be directly translated to degenerate codes, but only non-degenerate codes.

5.2 Bounds on Quantum Codes

Before considering bounds on quantum codes, one needs some notion of distance in C. The Hamming distance used for classical linear block codes cannot be used for quantum codes, however the idea can be generalised to define a distance measure for quantum codes. The idea of the Hamming distance as a measure is to determine the smallest number of bit flips one need to get from one codeword to another. In the quantum case, this can be generalised to determine the smallest number of Pauli operators needed to get from a codeword in C to another. This number is denoted d, and such codes are then said to be $[[n, k, d]]$ codes. Analogue to the classical case, an $[[n, k, d]]$ code can correct arbitrary errors on t qubits given that $d \geq 2t + 1$. With a fairly vague notion of distance for quantum codes, it is possible to consider how the parameters n, k, d must be related in order for the existence of a correction channel.

Now, assume that C is a non-degenerate code such that all correctable errors map C to orthogonal subspaces cf. the Knill-Laflamme conditions. If k and d, and thereby also t, are fixed, one obtains a lower bound on n. More precisely, since C is a 2^k-dimensional subspace of a 2^n-dimensional space, one can simply count how many orthogonal subspaces that fit into this space when each subspace corresponds to an error pattern of up to t errors. This leads to the quantum Hamming bound:

$$2^k \sum_{i=0}^{t} 3^i \binom{n}{i} \leq 2^n,$$

where each term in the summation accounts for how many error patterns on i qubits there exists with Pauli operators.

For example, if one wants to encode a single qubit such that any error on a single qubit can be corrected, the quantum Hamming bound with $k = 1, d = 3$ yields $2(1 + 3n) \leq 2^n$, which is satisfied for $n \geq 5$. Thus, at least five physical qubits are needed in order to encode a qubit against an arbitrary error on a single qubit with a non-degenerate code. It actually turns out, that not even a degenerate code can correct an arbitrary error with less than five qubits. This follows from the quantum Singleton bound for which a proof can be found in [NC10, ch. 12.4.3]. The bound states that for any quantum code, even degenerate ones, it holds that

$$n - k \geq 2(d - 1)$$

In the particular case where a single qubit of information is to be encoded such that an arbitrary error can be corrected, the bound states that $n - 1 \geq 2(3 - 1) = 4$, which implies that $n \geq 5$. Thus, there exists no quantum error-correcting code that can encode a qubit of information such that a single error can be corrected in fewer than five physical qubits.

Now that the fundamental concepts of quantum error-correcting codes have been presented, it is time to examine a particular class of such codes, namely stabilizer codes.

6 | Stabilizer Codes

One of the most important types of quantum codes are stabilizer codes. Such codes are based upon group theory, especially stabilizer group actions, hence it is briefly presented in Chapter C. The terminology is therefore assumed known in this chapter, which focus on the application of stabilizer groups in designing quantum error-correcting codes. The chapter is based on [NC10, ch. 10.5].

In order to create stabilizer codes is it necessary to see how stabilizer groups are related to quantum states and channels.

6.1 The Stabilizer Formalism

As illustrated in Example 3.6, the effect of a channel depends on the state prior to the channel. Some states may be completely corrupted while others remain the same. This is analogous to the situation where some elements in a set is stabilized by a group action, while others are changed in some way. Hence, the connection between channels and errors is closely related to that of stabilizer groups. More precisely, a group of Kraus operators acting upon a set of quantum states by some group action translates the problem of determining states that are indifferent to channels to that of finding stabilizer groups. The first problem is then to determine a group of Kraus operators. Since the Pauli matrices form a basis for complex matrices cf. Theorem B.7, it can particularly be used to generate every Kraus operator. Hence, the group generated by the Pauli matrices is one possible group under matrix multiplication. It is called the Pauli group and is denoted $\mathcal{G}_1$ as it acts on a single qubit. Simple calculations show that it is given by

$$\mathcal{G}_1 = \langle X, Y, Z \rangle = \{\pm I, \pm \mathrm{i} I, \pm X, \pm \mathrm{i} X, \pm Y, \pm \mathrm{i} Y, \pm Z, \pm \mathrm{i} Z\}.$$

Since the binary operation for the group is matrix multiplication, the group is non-abelian, which follows from the generators being anti-commuting cf. Theorem B.7. In fact, the properties analogue to those in Theorem B.7 hold for the elements of $\mathcal{G}_1$; i) all elements are unitary, ii) elements with a factor of ± 1 are Hermitian and have eigenvalues ± 1, while those with a factor $\pm \mathrm{i}$ are anti-Hermitian and have eigenvalues $\pm \mathrm{i}$, iii) elements either commute or anti-commute.

However, since more qubits are needed in the construction of codes, $\mathcal{G}_1$ merely serves as a foundation for groups that use the tensor product of Pauli matrices as the underlying set. The Pauli group $\mathcal{G}_n$ is defined as the group consisting of n-fold tensor products of Pauli matrices, i.e.,

$$\mathcal{G}_n = \{g_1 \otimes g_2 \otimes \cdots \otimes g_n \,|\, g_i \in \mathcal{G}_1\}. \tag{6.1}$$

It would be tedious to write all elements of $\mathcal{G}_n$ every time, hence some convenient notation is introduced. Firstly, the identity element in $\mathcal{G}_n$ is $I^{\otimes n}$, however it is merely denoted I. Secondly, instead of explicitly writing the tensor product in elements of $\mathcal{G}_n$, it is simply written with juxtaposition and subscript denoting the qubit that it acts upon. As an example, $X \otimes Z \in \mathcal{G}_2$ is simply denoted $X_1 Z_2$. Lastly, identity elements are omitted, but should be clear from context, e.g., $X \otimes Z \otimes I \in \mathcal{G}_3$ is also simply denoted $X_1 Z_2$. These notational conventions imply that $\mathcal{G}_n$ can be written more self-contained as

$$\mathcal{G}_n = \{c g_1 g_2 \cdots g_n \,|\, c \in \{\pm 1, \pm \mathrm{i}\}, g_i \in \{I, X, Y, Z\}\}.$$

Since the group $\mathcal{G}_n$ has n qubits that is acted upon by a Pauli matrix, or the identity, as well as the scalar c, its order is $|\mathcal{G}_n| = 4^{n+1}$.

Now that the Pauli group $\mathcal{G}_n$ has been defined, it is possible to define stabilizer groups corresponding to it. Although it should be clear at this point, $\mathcal{G}_n$ acts upon quantum states of n qubits, which has state space $(\mathbb{C}^2)^{\otimes n}$. Hence, it is possible to define a group action, particularly a stabilizer group, $A : \mathcal{G}_n \times (\mathbb{C}^2)^{\otimes n} \to (\mathbb{C}^2)^{\otimes n}$. So, let $\mathcal{S}$ be a subgroup of $\mathcal{G}_n$. It is then possible to define a set $V_{\mathcal{S}}$ consisting of all the vectors in $(\mathbb{C}^2)^{\otimes n}$ that are stabilized by all elements of $\mathcal{S}$. In other words, $\mathcal{S}$ is a stabilizer of the subspace $V_{\mathcal{S}}$. Since the stabilizer framework is applied to an operator on a state vector, elements of the stabilizer of $V_{\mathcal{S}}$ must be operators that has elements of $V_{\mathcal{S}}$ in its eigenspace associated with the eigenvalue one. Said differently, for s to be in $\mathcal{S}$, its $+1$-eigenspace must at least be $V_{\mathcal{S}}$. But since the subspace $V_{\mathcal{S}}$ is stabilized by all elements of $\mathcal{S}$, $V_{\mathcal{S}}$ is exactly the intersection of $+1$-eigenspaces of elements in $\mathcal{S}$. Hence, $V_{\mathcal{S}}$ can be defined purely in terms of linear algebra;

$$V_{\mathcal{S}} = \bigcap_{s \in \mathcal{S}} E_s, \quad E_s = \{v \,|\, sv = v\}.$$

A few remarks should be made regarding this definition of the stabilized subspace $V_{\mathcal{S}}$. Firstly, by describing $\mathcal{S}$ by its generators, it is sufficient to only take the intersection of the $+1$-eigenspaces of the generators. Secondly, in the end, it is only quantum states that are of relevance, hence v should be normalised in practice. This is nonetheless merely a practicality since $V_{\mathcal{S}}$ is a subspace. Lastly, even though quantum states are only defined up to some global phase, it is sufficient to only consider one particular phase, and the rest are naturally also included in $V_{\mathcal{S}}$.

To further clarify the stabilizer framework in the setting of quantum mechanics, an example follows.

> **Example 6.1: Stabilizer of a Bell State**
> Consider the case where the state $|\psi\rangle$ it the EPR pair $|\Phi^+\rangle$ such that
>
> $$|\psi\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle).$$
>
> Depending on the context, different aims may be considered. From one perspective could it be interesting to determine whether a given set of elements in $\mathcal{G}_2$ stabilizes $|\psi\rangle$ or not. From another perspective, it may be more interesting to determine the subgroup $\mathcal{S}$ that stabilizes $|\psi\rangle$. Since the first question is simply a verification process, it is not further discussed for now. Hence, the problem is to determine a non-trivial stabilizer of $|\psi\rangle$.
>
> Again, different approaches may be taken. It is typically easiest to deduce some elements that must be in $\mathcal{S}$, and then use that as a foundation. For this Bell states, is it intuitively easy to deduce four elements of $\mathcal{S}$. Firstly, $I \in \mathcal{S}$ is trivial. Secondly, the amplitudes of $|\psi\rangle$ are identical, hence flipping the amplitudes stabilizes $|\psi\rangle$. This is achieved by flipping both qubits, i.e., $X_1X_2 \in \mathcal{S}$. Thirdly, since there are two qubits, two phase flips stabilize $|\psi\rangle$, which is achieved by Z_1Z_2. Lastly, combining these implies that bit flipping and phase shifting both qubits also stabilizes $|\psi\rangle$. This is at first sight achieved by Y_1Y_2, however this introduces a global phase of -1, hence it instead follows that $(\mathrm{i}Y) \otimes (\mathrm{i}Y) = (-\mathrm{i}Y) \otimes (-\mathrm{i}Y) = -Y_1Y_2 \in \mathcal{S}$. Thus, simple intuition yields that $\mathcal{S} \supseteq \{I, X_1X_2, Z_1Z_2, -Y_1Y_2\}$. The subgroup criterion can then be used to determine whether this set is a subgroup of $\mathcal{G}_2$ or not. It is clearly nonempty, and it is also closed under multiplication, which is more apparent by noticing that $-Y_1Y_2 = X_1X_2Z_1Z_2 = Z_1Z_2X_1X_2$. Thus $\mathcal{S} = \{I, X_1X_2, Z_1Z_2, X_1X_2Z_1Z_2\}$. It is however more convenient to deal with generators such that $\mathcal{S} = \langle X_1X_2, Z_1Z_2\rangle$.
>
> In terms of $+1$-eigenspaces, is it easily shown that $E_{X_1X_2} = \mathrm{span}\{|\psi\rangle)\}, E_{Z_1Z_2} = \mathrm{span}\{|00\rangle, |11\rangle\}$, hence the intersection of eigenspaces is exactly $|\psi\rangle$. Thus, not only does $\mathcal{S}$ stabilize $|\psi\rangle$, but $|\psi\rangle_L$ is also the only state in $(\mathbb{C}^2)^{\otimes 2}$ that $\mathcal{S}$ stabilizes. It can furthermore be shown, e.g., by simply checking all elements of $\mathcal{G}_2$, that the subgroup that stabilizes $|\psi\rangle$ is $\mathcal{S}$ or one of its subgroups.

Now that $V_{\mathcal{S}}$ has been defined and Example 6.1 hopefully has improved the intuition for $V_{\mathcal{S}}$, it is worth considering whether there are any conditions that must be met in order for a subgroup, $\mathcal{S}$, of $\mathcal{G}_n$ to stabilize one or more quantum states. In other words, does every $\mathcal{S}$ stabilize a non-trivial subspace? This question is answered in the following subsection.

6.1.1 Conditions for Stabilizers of Quantum States

Since $V_{\mathcal{S}}$ is defined as the simultaneous $+1$-eigenspace of the elements of $\mathcal{S}$, it trivially holds that for $V_{\mathcal{S}}$ to be non-trivial, then all elements of $\mathcal{S}$ must have $+1$ as an eigenvalue and those eigenspaces must have some overlap. The first condition implies that it is necessary, but not sufficient, that all elements of $\mathcal{S}$ are Hermitian, while the second implies that all elements must commute. Furthermore, $-I$ naturally cannot be contained in $\mathcal{S}$ since it does not have $+1$ as an eigenvalue. This simple restriction does in itself imply that $\mathcal{S}$ cannot contain anti-Hermitian elements. This follows by contraposition as $\mathcal{S}$ is a subgroup of $\mathcal{G}_n$, and thereby closed under matrix multiplication, and the fact that anti-Hermitian elements of $\mathcal{G}_n$ square to $-I$. It turns out that these criteria, i.e., $\mathcal{S}$ being abelian and $-I \notin \mathcal{S}$, also are sufficient.

Theorem 6.2: Conditions for Stabilized Vector Space Being Non-trivial
The vector space $V_{\mathcal{S}}$ stabilized by $\mathcal{S}$ is non-trivial if and only if $\mathcal{S}$ is abelian and $-I \notin \mathcal{S}$.[NC10, p. 455]

It has already been argued that these conditions are necessary, however, showing that they are sufficient requires a bit more effort. Thus, different results are presented in the remaining part of this section with the aim of ultimately proving Theorem 6.2.

The first step is to introduce the check matrix, which is a quantum analogue to the parity-check matrix, which will become apparent in Section 6.2.4. Assuming that $\mathcal{S} = \langle g_1, \ldots, g_m \rangle$ is a subgroup of $\mathcal{G}_n$, the check matrix is a binary $m \times 2n$ matrix, denoted G. Each row corresponds to a generator of $\mathcal{S}$, while the columns, loosely speaking, indicates which Pauli matrix is associated with that generator. More precisely, the construction of the check matrix is as follows; consider g_i for $i = 1, \ldots, m$. There are then, without the multiplicative factor, four possible matrices it can have related to its j-th qubit, which determine the i, j-th and $i, n+j$-th elements of the check matrix. Firstly, if it is I, both elements are zero. Secondly, if it is X, the former element is one and the latter is zero. Thirdly, if it is Z, the former is zero, the latter is one. Lastly, if it is Y, both elements are one. In other words, by disregarding the multiplicative factor, one defines an isomorphism between $\mathcal{G}_1$ and $\mathbb{F}_2^2$ given by

$$I = \begin{bmatrix} 0 & 0 \end{bmatrix}, \quad X = \begin{bmatrix} 1 & 0 \end{bmatrix}, \quad Z = \begin{bmatrix} 0 & 1 \end{bmatrix}, \quad Y = \begin{bmatrix} 1 & 1 \end{bmatrix}, \tag{6.2}$$

which then can be generalised to $\mathcal{G}_n$ and $\mathbb{F}_2^{2n}$ by the above method. The construction of the check matrix can be illustrated by continuing on Example 6.1.

Example 6.3: Check Matrix for EPR Pair
The stabilizer of the EPR pair $|\Phi^+\rangle = (|00\rangle + |11\rangle)/\sqrt{2}$ is $\mathcal{S} = \langle X_1X_2, Z_1Z_2 \rangle$. For the first generator, X_1X_2, the first qubit is acted upon by X, hence $g_{11} = 1$ and $g_{13} = 0$. As the second qubit also is affected by X, $g_{12} = 1, g_{14} = 0$. The second generator, Z_1Z_2, instead has a Z on each qubit, hence $g_{21} = g_{22} = 0$ and $g_{23} = g_{24} = 1$ by similar reasoning. Explicitly, the check matrix has the form

$$G = \left[\begin{array}{cc|cc} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{array}\right],$$

where the vertical bar between the second and third column merely is inserted for visual clarification.

The check matrix can be used to determine whether the conditions in Theorem 6.2 are satisfied are not. To see this, let $r(g)$ denote the row in the check matrix corresponding to generator g. It is also denoted $(x|z)$ occasionally, when it is convenient to distinguish between the left-hand side and the right-hand side of the check matrix. The i'th entry of the row vector is denoted $r(g)_i$. Notice that all arithmetic with the check matrix is done modulo two. By introducing the block matrix

$$\Lambda = \begin{bmatrix} 0 & I \\ I & 0 \end{bmatrix}, \quad 0, I \in \operatorname{End}(\mathbb{F}_2^n),$$

it is possible to determine whether two generators g, g' commutes or not. This is done by considering the so-called symplectic inner product given as

$$r(g)\Lambda r(g')^\top = \sum_{i=1}^{n} \Big(r(g)_i r(g')_{n+i} + r(g)_{n+i} r(g')_i \Big) = \sum_{i=1}^{n} \Big((x_i|z_i') + (x_i'|z_i) \Big).$$

The i'th term of the sum is one if and only if there is one X and Z intersect between g and g' on the i'th qubit, but not Y on both. Explicitly, the term is one if and only if the generators g, g' satisfies the unordered set relation $\{g_i, g_i'\} \in \{\{X, Y\}, \{X, Z\}, \{Y, Z\}\}$. From this description is it clear that the i'th term is one if and only if g_i and g_i' anti-commutes. Hence, the summation itself is one if and only if there is an odd number of indices/qubits where g and g' anti-commutes, which is equivalent to saying that g and g' anti-commutes. To briefly summarise,

$$r(g)\Lambda r(g')^\top = \begin{cases} 0, & \text{if and only if } g \text{ and } g' \text{ commutes,} \\ 1, & \text{if and only if } g \text{ and } g' \text{ anti-commutes.} \end{cases} \tag{6.3}$$

Thus, $\mathcal{S} = \langle g_1, \ldots, g_m \rangle$ is abelian if and only if all $r(g_i)\Lambda r(g_j)^\top = 0$ for all $i, j = 1, \ldots, m$ satisfying $i \neq j$.

A relation between the check matrix and the commutativity of $\mathcal{S}$ has been derived. It is, however, also possible to give a characteristic of $g \in \mathcal{S}$ given the other condition in Theorem 6.2, i.e., that $-I \notin \mathcal{S}$. Particularly, every $g \in \mathcal{S}$ can be written as some product of elements in $\{g_1, \ldots, g_m\}$ by definition. In the check matrix notation, it is easily verified that the product becomes a summation, i.e.,

$$g = \prod_j g_j \iff r(g) = \sum_j r(g_j), \quad j \in \{1, \ldots, m\}.$$

Even though the generators may not be commutative with respect to multiplication, the corresponding check matrix rows are commutative with respect to addition. It therefore follows that

$$g^2 = \prod_j g_j^2 \iff r(g^2) = \sum_j r(g_j^2) = 2\sum_j r(g_j) = 0,$$

which means that g^2 is a multiplicative factor of the identity ($\pm I, \pm \mathrm{i} I$). But if $-I \notin \mathcal{S}$, then $\pm \mathrm{i} I \notin \mathcal{S}$ since $\mathcal{S}$ is closed under multiplication, meaning that $g^2 = I$. Thus, elements $g \in \mathcal{S}$ are involutory. To summarise,

$$-I \notin \mathcal{S} \implies g = g^{-1} \quad \forall\, g \in \mathcal{S}. \tag{6.4}$$

This realisation can be used to determine a connection between independence of generators and linear independency of rows in the check matrix.

> **Theorem 6.4: Independence of Generators and Rows of the Check Matrix**
> Let $\mathcal{S} = \langle g_1, \ldots, g_m \rangle$ such that $-I \notin \mathcal{S}$. The generators $\{g_1, \ldots, g_m\}$ are independent if and only if the rows of the corresponding check matrix are linearly independent. [NC10, p. 457]

Proof
The proof is shown by contraposition. So assume that the rows of the check matrix are linearly dependent. That is

$$\sum_{j=1}^m a_j r(g_j) = 0, \quad a_j \in \mathbb{F}_2$$

such that $a_i = 1$ for some i. Since $-I \notin \mathcal{S}$ by assumption, (6.4) implies that $g^2 = I$, hence the right hand-side above corresponds to $r(I)$. Thus, linear dependency of the rows of the check matrix is equivalent to

$$\prod_j g_j^{a_j} = I.$$

But for i with $a_i = 1$, the above equation can be multiplied by g_i on both sides yielding

$$g_i \prod_j g_j^{a_j} = g_i \implies g_i = \prod_{j \neq i} g_j^{a_j},$$

which means that the generators are not independent as g_i can be written as the product of other generators. Thus, the rows of the check matrix are linearly dependent if and only if the generators are not independent, hence the contrapositive completes the proof. ■

This relation is used in the following important result.

Theorem 6.5: Commuting or Anti-commuting Element in the Pauli Group
Consider $\mathcal{G}_n$. For some $k \in \mathbb{N}$, let $m = n - k$. Let $\mathcal{S} = \langle g_1, \dots, g_m \rangle$ be generated by independent generators such that $-I \notin \mathcal{S}$. For any $i = 1, \dots, m$, there exists a $g \in \mathcal{G}_n$ such that $g g_i g^{-1} = -g_i$ and $g g_j g^{-1} = g_j$ for all $j \neq i$. [NC10, p. 458]

Proof
Since the generators of $\mathcal{S}$ are assumed to be independent, the assumptions in Theorem 6.4 are satisfied, hence the rows in the corresponding $(m \times 2n)$-dimensional check matrix, G, are linearly independent. Thus, there exists a $2n$-dimensional vector, x, such that $G\Lambda x = e_i$, where e_i is the m-dimensional canonical basis vector with a one in the i-th entry. Now, let $g \in \mathcal{G}_n$ be such that $r(g) = x^\top$. By construction of x, for every $j \in \{1, \dots, m\}$ then

$$r(g_j)\Lambda r(g)^\top = r(g_j)\Lambda x^\top = \begin{cases} 1, & \text{if } j = i, \\ 0, & \text{if } j \neq i, \end{cases}$$

where i is the one defining e_i. From (6.3), this is equivalent to saying that g and g_j anti-commutes when $j = i$ and commutes if $j \neq i$. That is,

$$g g_j = \begin{cases} -g_j g, & \text{if } j = i, \\ g_j g, & \text{if } j \neq i \end{cases} \iff g g_j g^{-1} = \begin{cases} -g_j, & \text{if } j = i, \\ g_j, & \text{if } j \neq i, \end{cases}$$

which completes the proof. ■

One important implication of this result is that it can be used to determine the dimension of $V_{\mathcal{S}}$.

Theorem 6.6: Dimension of Stabilized Space
Let $\mathcal{S} = \langle g_1, \dots, g_{n-k} \rangle$ be generated by independent and commuting elements from $\mathcal{G}_n$ such that $-I \notin \mathcal{S}$. Then $V_{\mathcal{S}}$ is a 2^k-dimensional vector space. [NC10, p. 458]

Proof
Let x be a binary vector of length $n - k$. Define

$$P_{\mathcal{S}}(x) = \prod_{j=1}^{n-k} \frac{I + (-1)^{x_j} g_j}{2}.$$

Notice that the i'th factor in the product is the Frobenius covariant related to g_i. Thus, if $x_i = 0$, the i'th factor is the projection onto the $+1$-eigenspace of g_i, while it for $x_i = 1$ is the projection onto the -1-eigenspace of g_i. Since the generators are assumed to commute, then so does their Frobenius covariants, hence $P_{\mathcal{S}}(x)$ is a projection onto the intersection of the corresponding eigenspaces, e.g., when $x = 0$, then

$$P_{\mathcal{S}}(0, \dots, 0) = \frac{1}{2^{n-k}} \prod_{j=1}^{n-k} (I + g_j),$$

which is the projection onto the mutual $+1$-eigenspace of the commuters, i.e., the projection onto $V_{\mathcal{S}}$.

Now, starting with $x = 0$, hence $P_{\mathcal{S}}(0, \dots, 0)$, it is possible to introduce ones into x. Since the assumptions in Theorem 6.5 are satisfied, for every generator g_j there exists a $g \in \mathcal{G}_n$ such that g anti-commutes with g_j, but commutes with all the other generators. By both left and right multiplying one such g, say the one related to g_1, onto $P_{\mathcal{S}}(0, \dots, 0)$, this only changes the sign of terms with g_1, i.e.,

$$g P_{\mathcal{S}}(0, \dots, 0) g^{-1} = P_{\mathcal{S}}(1, 0, \dots, 0).$$

Since g_1 was arbitrarily chosen, a single one can be introduced into x at any position. Products of different $g \in \mathcal{G}_n$ can then be used to introducing several ones. In other words, for every x there exists some $g \in \mathcal{G}_n$ such that $g P_{\mathcal{S}}(0, \dots, 0) g^{-1} = P_{\mathcal{S}}(x)$. But since g is unitary due to being in $\mathcal{G}_n$, this is equivalent to saying that $P_{\mathcal{S}}(x)$ has the same dimension as $V_{\mathcal{S}}$.

Furthermore for x and x' being distinct, it follows that

$$\begin{aligned}
P_{\mathcal{S}}(x)P_{\mathcal{S}}(x') &= \left(\prod_{i=1}^{n-k}\frac{I+(-1)^{x_i}g_i}{2}\right)\left(\prod_{j=1}^{n-k}\frac{I+(-1)^{x'_j}g_j}{2}\right) \overset{(a)}{=} \prod_{i=1}^{n-k}\left[\left(\frac{I+(-1)^{x_i}g_i}{2}\right)\left(\frac{I+(-1)^{x'_i}g_i}{2}\right)\right] \\
&= \prod_{i=1}^{n-k}\frac{1}{4}\left[I+(-1)^{x_i}g_i+(-1)^{x'_i}g_i+(-1)^{x_i+x'_i}g_i^2\right] \\
&\overset{(b)}{=} \prod_{i=1}^{n-k}\frac{1}{4}\left[(1+(-1)^{x_i+x'_i})I+2(-1)^{x_i}\delta(x_i,x'_i)g_i\right] = \prod_{i=1}^{n-k}\delta(x_i,x'_i)\left[\frac{I+(-1)^{x_i}g_i}{2}\right] \\
&= \begin{cases} P_{\mathcal{S}}(x), & \text{if } x = x', \\ 0, & \text{if } x \neq x', \end{cases},
\end{aligned}$$

where (a) follows from the generators being commutative by assumption such that the order of multiplication can be changed, and (b) from g_i being involutory. This implies that the projectors $\{P_{\mathcal{S}}(x)\}_x$ are orthogonal.

Lastly, by construction of $P_{\mathcal{S}}(x)$ it follows that

$$\sum_x P_{\mathcal{S}}(x) = \sum_x \prod_{j=1}^{n-k}\frac{I+(-1)^{x_j}g_j}{2} = \prod_{j=1}^{n-k}\sum_{x_j\in\{0,1\}}\frac{I+(-1)^{x_j}g_j}{2} = \prod_{j=1}^{n-k} I = I,$$

where the penultimate equality follows from terms related to g_j cancelling out while the identities sum up for changing x_j. The right-hand side is the 2^n-dimensional identity matrix, hence it is a projection onto a 2^n-dimensional space. The left-hand side is a sum of 2^{n-k} projections of the same size of $V_{\mathcal{S}}$. This implies that the dimension of $V_{\mathcal{S}}$ must be 2^k, which concludes the proof. ■

With all of these results, it is trivial to give the proof of Theorem 6.2, however it is explicitly given for completeness.

Proof (Theorem 6.2)
It is by contradiction shown that the two conditions are necessary. So let $V_{\mathcal{S}}$ be non-trivial such that it contains $|\psi\rangle$, and let $A, B \in \mathcal{S}$. Firstly, assume that A and B do not commute, which is equivalent to saying they anti-commute since they are elements of $\mathcal{S}$. Then $|\psi\rangle = AB\,|\psi\rangle = -BA\,|\psi\rangle = -\,|\psi\rangle$, which only holds true in the trivial case (the zero-vector), which is a contradiction to $V_{\mathcal{S}}$ being non-trivial. Thus, A, B must commute. Secondly, assume that $-I \in \mathcal{S}$ such that $-I\,|\psi\rangle = |\psi\rangle$. However, this is once again a contradiction to $V_{\mathcal{S}}$ being non-trivial.

The fact that the conditions are sufficient follows directly from Theorem 6.6. ■

Now that the main result has been proven, it is henceforth assumed that $\mathcal{S}$ is defined by independent and commuting generators such that $-I \notin \mathcal{S}$, unless stated otherwise. It will now be examined how the stabilizer formalism is useful in quantum coding theory.

6.2 Construction of Stabilizer Codes

As described in Chapter 5, a $[[n,k]]$ code C is a 2^k-dimensional subspace of $(\mathbb{C}^2)^{\otimes n}$. Such a subspace can by Theorem 6.6 be obtained by defining an abelian subgroup, $\mathcal{S}$, of $\mathcal{G}_n$ that has $n-k$ independent generators and $-I \notin S$. In that case, the subspace stabilized by $\mathcal{S}$, $V_{\mathcal{S}}$, is 2^k-dimensional. Hence, a $[[n,k]]$ code is in the stabilizer formalism defined as the subspace stabilied by $\mathcal{S} = \langle g_1, \dots, g_{n-k}\rangle$, $V_{\mathcal{S}}$, if the generators are independent and form an abelian group such that $-I \notin \mathcal{S}$. It is for the remainder of this chapter assumed that a $[[n,k]]$ code is considered unless stated otherwise.

All of the codes introduced in Chapter 4 are examples of stabilizer codes. For example, the bit flip repetition code is defined such that $V_{\mathcal{S}} = \text{span}\{|000\rangle, |111\rangle\}$, which implies that elements of $\mathcal{S}$ must contain an even number of phase flips, and nothing else. Hence $\mathcal{S} = \{I, Z_1Z_2, Z_1Z_3, Z_2Z_3\} = \langle Z_1Z_2, Z_2Z_3\rangle$.

The question is now how the three blocks in Figure 5.1, that is encoding, errors, and decoding, are formulated for stabilizer codes. This is theoretically considered in the following sections, whereafter it is combined into an example in Section 6.3. Since measurements are needed in order to formulate all of the three blocks, measurements in the stabilizer formalism are firstly considered.

6.2.1 Stabilizer Measurements

Consider a state $|\psi\rangle \in V_{\mathcal{S}}$ with $\mathcal{S} = \langle g_1, \dots, g_{n-k}\rangle$ for $g_i \in \mathcal{G}_n$. Furthermore, consider an element $g \in \mathcal{G}_n$, which without loss of generality can be assumed to have no multiplicative factor such that its eigenvalues are ± 1. Then, g is a Hermitian operator that can be used as the only observable in a projective measurement. Now, there are two cases; g either commutes with all of the generators of $\mathcal{S}$ or it anti-commutes with at least one of the them. The two cases are considered separately.

If g commutes with all generators, then $g|\psi\rangle = gg_i|\psi\rangle = g_ig|\psi\rangle$ for all generators g_i, where the first equality follows from $g_i \in \mathcal{S}$ such that it stabilizes $|\psi\rangle$. This implies that $g|\psi\rangle$ also is stabilized by $\mathcal{S}$, meaning that it is a multiple of $|\psi\rangle$. However, since g is assumed to have eigenvalues ± 1, this is equivalent to stating that $g|\psi\rangle = \pm|\psi\rangle$, which implies that $\pm g \in \mathcal{S}$. That is, either g or $-g$ stabilizes $|\psi\rangle$. Say $g \in \mathcal{S}$. Then, the probability of measuring $+1$ is

$$\Pr(+1) = \langle\psi|g|\psi\rangle = \langle\psi|\psi\rangle = 1.$$

The state after the measurement is therefore $g|\psi\rangle = |\psi\rangle$. Analogously, if $-g \in \mathcal{S}$, then $\Pr(-1) = 1$ and $-g|\psi\rangle = |\psi\rangle$. Thus, measuring $g \in \mathcal{S}$ has no effect on $|\psi\rangle \in V_{\mathcal{S}}$. Particularly, the generators of $\mathcal{S}$ can be measured without changing the state such that the outcome is either $+1$ or -1 with certainty.

If g anti-commutes with at least one generator, the generators can always be parameterised such that g only anti-commutes with one of the generators by the following reasoning. Say g anti-commutes with g_1 and g_2, then it commutes with g_1g_2. Thus, the generator g_2 can be replaced by g_1g_2 without changing $\mathcal{S}$ or the independency of generators. This can be done recursively to construct a generating set where only a single element anti-commutes with g. Say this element is g_1. To see how this affects the measurement of g, firstly decompose g into its Frobenius covariants, which by Section B.3 is

$$g = g_+ - g_- = \left(\frac{I+g}{2}\right) - \left(\frac{I-g}{2}\right).$$

Using this decomposition, the probability of measuring $+1$ can with the density operator formalism seen to be

$$\begin{aligned}\Pr(+1) = \operatorname{Tr}(g_+|\psi\rangle\langle\psi|) = \operatorname{Tr}\left(\frac{I+g}{2}|\psi\rangle\langle\psi|\right) &\overset{(a)}{=} \operatorname{Tr}\left(\frac{I+g}{2}g_1|\psi\rangle\langle\psi|\right) \overset{(b)}{=} \operatorname{Tr}\left(g_1\frac{I-g}{2}|\psi\rangle\langle\psi|\right)\\ \overset{(c)}{=} \operatorname{Tr}\left(\frac{I-g}{2}|\psi\rangle\langle\psi|g_1\right) &\overset{(d)}{=} \operatorname{Tr}\left(\frac{I-g}{2}|\psi\rangle\langle\psi|\right) = \operatorname{Tr}(g_-|\psi\rangle\langle\psi|) = \Pr(-1),\end{aligned}$$

where (a) follows from $g_1 \in \mathcal{S}$, (b) from g and g_1 anti-commute, (c) from the cyclic property of trace, and (d) from g_1 begin Hermitian as well as $g_1 \in \mathcal{S}$. This implies that $\Pr(+1) = \Pr(-1) = \frac{1}{2}$. After measurement, the state becomes

$$|\psi_\pm\rangle = \frac{g_\pm|\psi\rangle}{\sqrt{\Pr(\pm 1)}} = \frac{I \pm g}{2\sqrt{1/2}}|\psi\rangle = \frac{1}{\sqrt{2}}(|\psi\rangle \pm g|\psi\rangle),$$

where the sign depends on the outcome of the measurement. Say the outcome is $+1$. Applying g to $|\psi_+\rangle$ then yields

$$g|\psi_+\rangle = \frac{1}{\sqrt{2}}(g|\psi\rangle + g^2|\psi\rangle) = \frac{1}{\sqrt{2}}(g|\psi\rangle + |\psi\rangle) = |\psi_+\rangle.$$

Thus, g stabilizes $|\psi_+\rangle$. Since g commutes with $g_2, \dots, g_m$, these also stabilize $|\psi_+\rangle$, hence the stabilizer of $|\psi_+\rangle$ has generators $\langle g, g_2, \dots, g_m\rangle$. It follows analogously that if the measurement outcome is -1, then the stabilizer of $|\psi_-\rangle$ has generators $\langle -g, g_2, \dots, g_m\rangle$. Thus, $\pm g$ replaces g_1 as a stabilizer when $|\psi\rangle$ is transformed to $|\psi_\pm\rangle$. It should be explicitly noted that whether it is completely random whether $+g$ or $-g$ replaces g_1 as a stabilizer.

The general idea of measuring g is to introduce $\pm g$ into the stabilizer of $|\psi\rangle$ if it not already is there, hence such measurement is known as a stabilizer measurement. This is done by projecting the state into the $+1$-eigenspace of $\pm g$. Stabilizer measurements are, however, only a particular case of quantum 'evolution' that may change a system. When considering evolution as described in Section 2.2, the system is evolved according to some unitary operator $U \in \operatorname{End}((\mathbb{C}^2)^{\otimes n})$. If $|\psi\rangle$ evolves according to U, it holds that for all $g_i \in \mathcal{S}$ that $U|\psi\rangle = Ug_i|\psi\rangle$. The unitarity of U then furthermore implies that $Ug_i|\psi\rangle = Ug_iU^\dagger U|\psi\rangle$, which combined implies that $U|\psi\rangle$ is stabilized by $Ug_iU^\dagger$. Since this holds for all $g_i \in \mathcal{S}$ and all $|\psi\rangle \in V_{\mathcal{S}}$, the stabilizer of $UV_{\mathcal{S}}$ is $\mathcal{S}' = \langle Ug_1U^\dagger, \dots, Ug_{n-k}U^\dagger\rangle$. These results are used repeatedly when discussion the encoding process of a stabilizer code.

6.2.2 Encoding

As previously discussed, the encoding of a quantum code is done by applying a unitary operation to the logical basis states of $V_{\mathcal{S}}$. The first problem is therefore to determine such logical basis states. In the canonical basis, that is $|x_1 \cdots x_n\rangle$ must be encoded into $|x_1 \cdots x_n\rangle_L$ for $x_i \in \{0,1\}$. Generally, any 2^k orthonormal vectors in $V_{\mathcal{S}}$ can be used as logical basis states, however it is convenient to determine them systematically. This is done in two steps; 1) the check matrix of $\langle g_1, \dots, g_{n-k}\rangle$ is converted to standard form, 2) k additional linearly independent rows that also commute with all the generators, are found. After the logical basis states have been determined, $|\psi\rangle$ must be encoded. Depending on the usage of the encoded state, this adds either one or two extra steps to the above approach; 3) stabilizer measurements is used to encode a known state, 4) if the state is unknown, then an additional measurement is required. It should be noted that even though step 3) and 4) are fairly simple, they are not unitary. If one requires a unitary encoding, see [Got97, ch. 4.2]. The four steps are now discussed.

Let G be the check matrix of $\langle g_1, \dots, g_{n-k}\rangle$, and let G_X and G_Z denote the submatrix related to X and Z elements of G, respectively. Then, G is given as

$$G = \left[\begin{array}{c|c} \overbrace{\;G_X\;}^{n} & \overbrace{\;G_Z\;}^{n} \end{array}\right] \}\, n-k,$$

where the braces are used to illustrate the dimensions of the submatrices. As previously discussed, addition of rows in the check matrix corresponds to multiplication of the associated generators. Since all generators are assumed to be independent, it is always possible to use $g_i g_j$ for $i \neq j$ as generator of $\mathcal{S}$ instead of either g_i or g_j without altering the independency of the generators. Thus, addition of two rows of G is an allowed elementary operation that does not change the linear independency of rows. Furthermore, swapping rows of G or swapping the same column indices in both G_X and G_Z corresponds to re-indexing generators or qubits, respectively, which naturally also is allowed. Since arithmetic on G is done modulo 2, these operations corresponds to performing Gaussian elimination (including column swaps) on G. This can firstly be done on the matrix G_X, yielding the check matrix

$$G = \left[\begin{array}{cc|cc} \overbrace{\quad I \quad}^{r} & \overbrace{\quad A \quad}^{n-r} & \overbrace{\quad B \quad}^{r} & \overbrace{\quad C \quad}^{n-r} \\ 0 & 0 & D & E \end{array}\right] \begin{array}{l} \}\, r \\ \}\, n-k-r \end{array},$$

where r is the rank of G_X. Similarly, Gaussian elimination can be performed on E to obtain

$$G = \left[\begin{array}{ccc|ccc} \overbrace{\quad I \quad}^{r} & \overbrace{\quad A_1 \quad}^{s} & \overbrace{\quad A_2 \quad}^{n-r-s} & \overbrace{\quad B \quad}^{r} & \overbrace{\quad C_1 \quad}^{s} & \overbrace{\quad C_2 \quad}^{n-r-s} \\ 0 & 0 & 0 & D_1 & I & F \\ 0 & 0 & 0 & D_2 & 0 & 0 \end{array}\right] \begin{array}{l} \}\, r \\ \}\, s \\ \}\, n-k-r-s \end{array},$$

where s is the rank of E. Since all generators must commute, all pairs of rows, $(r(g), r(g'))$ must cf. (6.3) satisfy that their symplectic inner product is zero. Particularly, the first r generators, given by the block matrix G_r, commutes with the last $m = (n-k-r-s)$ generators, given by G_m if and only if $G_r \Lambda G_m^\top = 0 \in \operatorname{Hom}(\mathbb{C}^r, \mathbb{C}^m)$. Writing out the product gives

$$G_r \Lambda G_m^\top = \left[\begin{array}{ccc|ccc} I & A_1 & A_2 & B & C_1 & C_2 \end{array}\right] \begin{bmatrix} 0 & I \\ I & 0 \end{bmatrix} \left[\begin{array}{c} 0^\top \\ 0^\top \\ 0^\top \\ \hline D_2^\top \\ 0^\top \\ 0^\top \end{array}\right] = \left[\begin{array}{ccc|ccc} I & A_1 & A_2 & B & C_1 & C_2 \end{array}\right] \left[\begin{array}{c} D_2^\top \\ 0^\top \\ 0^\top \\ \hline 0^\top \\ 0^\top \\ 0^\top \end{array}\right] = D_2^\top,$$

which imply that $D_2 = 0$ in order for the generators to commute. Hence, the last block row of G is all zeros, which corresponds to all of the last m generators being I. Naturally, I cannot be used as a generator of $\mathcal{S}$ that is independent from the other generators, hence E must have full rank such that there are no zero rows in the bottom of G. That is, $s = n-k-r$, such that the check matrix has the simpler form

$$G = \left[\begin{array}{ccc|ccc} \overbrace{I}^{r} & \overbrace{A_1}^{n-k-r} & \overbrace{A_2}^{k} & \overbrace{B}^{r} & \overbrace{C_1}^{n-k-r} & \overbrace{C_2}^{k} \\ 0 & 0 & 0 & D_1 & I & F \end{array}\right] \begin{array}{l} \}r \\ \}n-k-r \end{array},$$

where s is omitted in order to use as few variables as needed. Lastly, C_1 can be transformed to $0 \in \mathrm{Hom}(\mathbb{C}^{n-k-r}, \mathbb{C}^r)$ by taking appropriate sums of rows, which yield the following standard form of G

$$G = \left[\begin{array}{ccc|ccc} \overbrace{I}^{r} & \overbrace{A_1}^{n-k-r} & \overbrace{A_2}^{k} & \overbrace{B}^{r} & \overbrace{0}^{n-k-r} & \overbrace{C}^{k} \\ 0 & 0 & 0 & D & I & E \end{array}\right] \begin{array}{l} \}\, r \\ \}\, n-k-r \end{array}, \tag{6.5}$$

where some submatrices have been re-labelled for a cleaner representation, e.g., B is not necessarily the same in both matrices. Now that G is represented in standard form, the next part is to find k additional rows corresponding to elements of $\mathcal{G}_n$ that are independent of all generators of $\mathcal{S}$ yet still commutes with all of them. Naturally, they must also mutually be independent and commute. One method to do so, is to find rows that commute with those of G, and then determine whether they are independent as well. Let the check matrix of the new rows be denoted G_k and be decomposed into six block matrices as

$$\left[\begin{array}{ccc|ccc} F_1 & F_2 & F_3 & F_4 & F_5 & F_6 \end{array}\right],$$

where each element has k rows and $r, n-k-r, k, r, n-k-r, k$ columns, respectively. For the rows of G_k to commute with the rows of G, then $G\Lambda G_k^\top = 0 \in \mathrm{Hom}(\mathbb{C}^{n-k}, \mathbb{C}^k)$ must be satisfied. Writing out this symplectic inner product yields

$$G\Lambda G_k^\top = \left[\begin{array}{ccc|ccc} I & A_1 & A_2 & B & 0 & C \\ 0 & 0 & 0 & D & I & E \end{array}\right] \left[\begin{array}{c|c} 0 & I \\ I & 0 \end{array}\right] \left[\begin{array}{c} F_1^\top \\ F_2^\top \\ F_3^\top \\ \hline F_4^\top \\ F_5^\top \\ F_6^\top \end{array}\right] = \begin{bmatrix} F_4^\top + A_1 F_5^\top + A_2 F_6^\top + B_1 F_1^\top + C F_3^\top \\ D F_1^\top + F_2^\top + E F_3^\top \end{bmatrix}.$$

To obtain the zero-matrix, one can simply set $F_1 = F_2 = F_3 = 0$ to obtain zero in the second entry, which then leads $F_4 = A_2^\top, F_5 = 0, F_6 = I$ to solve the first entry. Thus, one possible solution to $G\Lambda G_k^\top = 0$ is

$$G_k = \left[\begin{array}{ccc|ccc} 0 & 0 & 0 & A_2^\top & 0 & I \end{array}\right]. \tag{6.6}$$

By construction, these elements commute with all generators, while they are easily seen to mutually commute due to only having non-zero elements on the right-hand side of the vertical bar. It is furthermore trivially seen that appending G_k to G does not destroy the linearly independency of the rows. Explicitly, since the first column of G contains I in the first r rows and 0 in the last $(n-k-r)$ while G_k also contains 0, the generators of G_k cannot be written in terms of the first r generators of G. Similar reasoning applied to the fifth column indicates that neither can the generators of G_k be written in terms of the last $(n-k-r)$ generators of G. Hence G_k is a valid solution.

Another solution could be to set $F_1 = 0, F_2 = E^\top, F_3 = I$, which then leads the first entry to be solved by $F_4 = C^\top, F_5 = F_6 = 0$. Thus, another solution is

$$G_k' = \left[\begin{array}{ccc|ccc} 0 & E^\top & I & C^\top & 0 & 0 \end{array}\right], \tag{6.7}$$

which by similar arguments as above can be seen to satisfy the imposed conditions.

Now, let the Pauli operators corresponding to the i-th rows of G_k and G_k' be denoted $\bar{Z}_i$ and $\bar{X}_i$, respectively. Computing the symplectic inner product of G_k and G_k' yields

$$G_k\Lambda(G_k')^\top = \left[\begin{array}{ccc|ccc} 0 & 0 & 0 & A_2^\top & 0 & I \end{array}\right] \left[\begin{array}{c} C \\ 0 \\ 0 \\ \hline 0 \\ E_2 \\ I \end{array}\right] = I,$$

meaning that $\bar{Z}_i$ commutes with $\bar{X}_j$ unless $i = j$, in which case they anti-commute.

The logical basis states in the canonical basis, $\{|x_1 \cdots x_k\rangle_L\}$ for $x_i \in \{0,1\}$, is in the systematic form defined to be stabilized by the generators

$$\langle g_1, \ldots, g_{n-k}, (-1)^{x_1}\bar{Z}_1, \ldots, (-1)^{x_k}\bar{Z}_k\rangle . \tag{6.8}$$

Hence, the logical ground state, $(|0\rangle^{\otimes n})_L$ is defined to be the state stabilized by $\langle g_1, \ldots, g_{n-k}, \bar{Z}_1, \ldots, \bar{Z}_k\rangle$. Since there are n generators, $(|0\rangle^{\otimes n})_L$ is a 1-dimensional subspace of $\mathcal{G}_n$ cf. Theorem 6.6 such that it actually corresponds to a valid state. This construction of the logical basis states implies that applying $\bar{Z}_i$ to any of the logical basis states has no effect on the stabilizers of the logical basis states as $\bar{Z}_i$ commutes with all generators in Equation (6.8). However, since the logical basis states with $x_i = 1$ is stabilized by $-\bar{Z}_i$, it follows that the sign of such logical basis states are flipped when applying $\bar{Z}_i$. For example, if $k = 1$, then $\bar{Z}\,|0\rangle_L = |0\rangle_L$ and $\bar{Z}\,|1\rangle_L = -\,|1\rangle_L$, hence $\bar{Z}$ acts as a phase flip operator on the logical basis states, which is indicated by the bar in $\bar{Z}$ similarly to the notation used in Chapter 4. Generally, $\bar{Z}_i$ acts as the phase flip operator on the i-th logical qubit.

Since $\bar{X}_i$ commutes with all of the generators in (6.8) except $\bar{Z}_i$, applying $\bar{X}_i$ to the logical basis states bit flips the i-th logical qubit due to flipping the sign of the generator corresponding to $\bar{Z}_i$. By again considering the case where $k = 1$, then $\bar{X}\,|0\rangle_L$ is stabilized by $\langle \bar{X}g_1\bar{X}, \ldots, \bar{X}g_{n-k}\bar{X}, \bar{X}\bar{Z}\bar{X}\rangle = \langle g_1, \ldots, g_{n-k}, -\bar{Z}\rangle$ due to the commutation relations, which implies that $\bar{X}\,|0\rangle_L = |1\rangle_L$. Analogously, $\bar{X}\,|1\rangle_L = |0\rangle_L$, hence $\bar{X}$ acts as the logical bit flip operator. Generally, $\bar{X}_i$ act as the bit flip operator on the i-th logical qubit. Now that that the logical basis states have been defined, the next step is how to use them for encoding.

Assume first that Alice wants to encode a known state of k qubits into n qubits. One approach to do so is to initialise all of the n qubits in the ground state $|0\rangle^{\otimes n}$, and transform it into the logical ground state, $(|0\rangle^{\otimes k})_L$, which is stabilized by $\langle g_1, \ldots, g_{n-k}, \bar{Z}_1, \ldots, \bar{Z}_k\rangle$, by performing stabilizer measurements.

Thus, Alice initialises $|\psi_0\rangle = |0\rangle^{\otimes n}$, which is stabilized by the generators $\langle Z_1, \ldots, Z_n\rangle$. Alice can then use the first generators, g_1, as an observable for a stabilizer measurement, which results in the state $|\psi_1\rangle$ that is stabilized by $\langle \pm g_1, Z_2', \ldots, Z_n'\rangle$, where Z' indicate that the elements might have been multiplied by Z_1 such that at most one generator of $|\psi_0\rangle$ anti-commutes with g_1. Performing stabilizer measurements for all generators in $\langle g_1, \ldots, g_{n-k}, \bar{Z}_1, \ldots, \bar{Z}_k\rangle$ yields a state $|\psi_n\rangle$ that is stabilized by $\langle \pm g_1, \ldots, \pm g_{n-k}, \pm\bar{Z}_1, \ldots, \pm\bar{Z}_k\rangle$. To obtain $|0\rangle_L$, all signs of the generators must be $+1$. By Theorem 6.5, each sign can be turned into $+1$ by multiplying the generators by appropriate elements of $\mathcal{G}_n$. By doing so, Alice obtains a state that is stabilized by $\langle g_1, \ldots, g_{n-k}, \bar{Z}_1, \ldots, \bar{Z}_k\rangle$, which by definition is $(|0\rangle^{\otimes k})_L$. After she has obtained $(|0\rangle^{\otimes k})_L$, she can obtain the other logical basis states by applying appropriate$\bar{X}_i$ operators. However, this procedure only works if Alice knows the state that must be encoded. If not, then an additional measurement is needed instead of applying the appropriate $\bar{X}$.

For simplicity, it is only shown how to encode for $k = 1$, where the state to be encoded is $|\psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle$. Now, consider preparing the state

$$|\psi_0\rangle = |\psi\rangle \otimes |0\rangle_L = \alpha\,|0\rangle \otimes |0\rangle_L + \beta\,|1\rangle \otimes |0\rangle_L\,,$$

which easily can be prepared, e.g., by the method described above. Although $|0\rangle_L$ consists of n physical qubits such that $|\psi_0\rangle$ consists of $(n+1)$, $|\psi_0\rangle$ can be considered as a two-qubit state since the last n qubits is a logical qubit. Applying the CNOT gave to the state then yields

$$|\psi_1\rangle = \text{CNOT}\,|\psi_0\rangle = \begin{bmatrix} I & 0 \\ 0 & \bar{X} \end{bmatrix} |\psi_0\rangle = \alpha\,|0\rangle \otimes |0\rangle_L + \beta\,|1\rangle \otimes |1\rangle_L\,,$$

where $\bar{X}$ is used to indicate that the second qubit of $|\psi_0\rangle$ is a logical qubit. By applying the Hadamard operators (logical Hadamard on the second qubit), the state becomes

$$\begin{aligned} |\psi_2\rangle &= (H \otimes \bar{H})\,|\psi_1\rangle = \alpha\,|+\rangle \otimes |+\rangle_L + \beta\,|-\rangle \otimes |-\rangle_L \\ &= \frac{\alpha+\beta}{\sqrt{2}}\Big(\,|0\rangle \otimes |0\rangle_L + |1\rangle \otimes |1\rangle_L\Big) + \frac{\alpha-\beta}{\sqrt{2}}\Big(\,|0\rangle \otimes |1\rangle_L + |1\rangle \otimes |0\rangle_L\Big). \end{aligned}$$

Applying the CNOT gate once again yields

$$\begin{aligned} |\psi_3\rangle = \text{CNOT}\,|\psi_2\rangle &= \frac{\alpha+\beta}{\sqrt{2}}\Big(\,|0\rangle \otimes |0\rangle_L + |1\rangle \otimes |0\rangle_L\Big) + \frac{\alpha-\beta}{\sqrt{2}}\Big(\,|0\rangle \otimes |1\rangle_L + |1\rangle \otimes |1\rangle_L\Big) \\ &= \alpha\,|+\rangle \otimes |+\rangle_L + \beta\,|+\rangle \otimes |-\rangle_L\,. \end{aligned}$$

Lastly, applying the Hadamard operators again yields

$$|\psi_4\rangle = (H \otimes \bar{H}) |\psi_3\rangle = \alpha |0\rangle \otimes |0\rangle_L + \beta |0\rangle \otimes |1\rangle_L = |0\rangle \otimes |\psi\rangle_L ,$$

which shown that the information of $|\psi\rangle$ has been encoded into $|\psi\rangle_L$. The additional qubit in the ground state can merely be traced out.

It has now been shown how to encode a state into a stabilizer code, hence it can be analysed how errors can be corrected for such codes.

6.2.3 Existence of Correction Channel

The existence of a correction channel can be determined by formulating the Knill-Laflamme conditions in the stabilizer formalism. This has already been partly done implicitly in Section 6.2.2 by introducing the logical Pauli operators $\{\bar{X}_i\}_i$ and $\{\bar{Z}_i\}_i$. More rigorously, since $V_{\mathcal{S}}$ is a subspace of $(\mathbb{C}^2)^{\otimes n}$, the elements of $\mathcal{G}_n$ are exactly the (discretised) errors that may corrupt the encoded state $|\psi\rangle_L$, which by Corollary 5.2 is sufficient to consider. In fact, it is sufficient to only consider the elements of $\mathcal{G}_n$ with no multiplicative factor, i.e., it has factor $+1$, as the other elements only differ from these by introducing a global phase that is irrelevant. By definition, the stabilisers of $V_{\mathcal{S}}$ acts as the identity operator on $V_{\mathcal{S}}$, while the logical Pauli operators acts as non-trivial Pauli operators on $V_{\mathcal{S}}$, meaning that it takes an element from $V_{\mathcal{S}}$ to a different element in $V_{\mathcal{S}}$. By construction of the logical basis states in the standard form, the logical Pauli operators commute with all generators of $\mathcal{S}$, thus they are elements of the centraliser of $\mathcal{S}$, $C(\mathcal{S})$. In the literature, they are often said to be elements of the normaliser of $\mathcal{S}$, however, the two subgroups are equivalent in this framework due to the following theorem.

Theorem 6.7: Equivalence of Centraliser and Normaliser
Let $\mathcal{A}$ be a subgroup of $\mathcal{G}_n$ such that $-I \notin \mathcal{A}$. Then $C(\mathcal{A}) = N(\mathcal{A})$. [NC10, p. 466]

Proof
Since $C(\mathcal{A}) \subseteq N(\mathcal{A})$ always holds, it is sufficient to show that $N(\mathcal{A}) \subseteq C(\mathcal{A})$. Thus, let $g_i \in \mathcal{A}$ be given and assume that $g \in N(\mathcal{A})$. Since elements of $\mathcal{G}_n$ either commute or anti-commute, it follows that

$$g g_i g^{-1} = \pm g_i g g^{-1} = \pm g_i.$$

However, since $g_i \in \mathcal{A}$ and $-I \notin \mathcal{A}$ by assumption, then $-g_i \notin \mathcal{A}$. Hence, $g \in N(\mathcal{A})$ implies that $g g_i g^{-1} = g_i$. Since this holds for all $g_i \in \mathcal{A}$, $N(\mathcal{A}) \subseteq C(\mathcal{A})$, which completes the proof. ■

Since $\mathcal{S}$ is a subgroup of $\mathcal{G}_n$ not containing $-I$, the result in Theorem 6.7 applies to $\mathcal{S}$. Since $\mathcal{S}$ furthermore is abelian, it also holds that $\mathcal{S} \subseteq N(\mathcal{S})$. This is important since it implies that any error, $E \in \mathcal{G}_n$, falls in either of the following three subsets; $\mathcal{S}, N(\mathcal{S}) \setminus \mathcal{S}$, or $\mathcal{G}_n \setminus N(\mathcal{S})$. The effect of these three cases on $V_{\mathcal{S}}$ are now considered individually in an intuitive manner:

i) If $E \in \mathcal{S}$, then it acts as an identity operator on $V_{\mathcal{S}}$, thus E is not corrupting the state $|\psi\rangle_L$.

ii) If $E \in N(\mathcal{S}) \setminus \mathcal{S}$, then it non-trivially takes elements $V_{\mathcal{S}}$ to $V_{\mathcal{S}}$, e.g., $\bar{X}|\psi\rangle_L = \bar{X}(\alpha |0\rangle_L + \beta |1\rangle_L) = \alpha |1\rangle_L + \beta |0\rangle_L$, which corrupts $|\psi\rangle_L$ in a manner that cannot be detected as it still belongs to $V_{\mathcal{S}}$.

iii) If $E \in \mathcal{G}_n \setminus N(\mathcal{S})$, then it anti-commutes with at least one generator of $\mathcal{S}$, thus $V_{\mathcal{S}}$ is sent to a subspace that is orthogonal to $V_{\mathcal{S}}$. This error should then be correctable.

This discussion is now made more rigorous by formulating the Knill-Laflamme conditions, which state that $P_C E_i^\dagger E_j P_C = h_{ij} P_C$ for some Hermitian H, in the stabilizer formalism.

Theorem 6.8: Stabilizer Knill-Laflamme Conditions
Let $\mathcal{S}$ be the stabilizer for a code $V_{\mathcal{S}}$. Assume $\mathcal{E}_C \subset \mathcal{G}_n$ is described by Kraus operators $\{E_i\}_i$ satisfying $E_i^\dagger E_j \notin N(\mathcal{S}) \setminus \mathcal{S}$ for all i, j. Then there exists a correction channel for $\mathcal{E}_C$ on $V_{\mathcal{S}}$. [NC10, p. 466]

Proof
Let $\mathcal{S} = \langle g_1, \dots, g_{n-k} \rangle$ such that the projection onto $V_{\mathcal{S}}$ is given by

$$P_{\mathcal{S}} = \frac{1}{2^{n-k}} \prod_{i=1}^{n-k} (I + g_i).$$

Under the assumption that $E_i^\dagger E_j \notin N(\mathcal{S}) \setminus \mathcal{S}$, then $E_i^\dagger E_j \in \mathcal{S}$ or $E_i^\dagger E_j \in \mathcal{G}_n \setminus N(\mathcal{S})$.

If $E_i^\dagger E_j \in \mathcal{S}$, there exists scalars $a_l \in \{0,1\}$ such that $E_i^\dagger E_j = \prod_{l=1}^{n-k} g_l^{a_l}$. Hence, applying $E_i^\dagger E_j$ to $P_\mathcal{S}$ yields

$$E_i^\dagger E_j P_\mathcal{S} = \left(\prod_{l=1}^{n-k} g_l^{a_l}\right)\left(\frac{1}{2^{n-k}}\prod_{m=1}^{n-k}(I+g_m)\right) \overset{(a)}{=} \frac{1}{2^{n-k}}\prod_{l=1}^{n-k} g_l^{a_l}(I+g_l) \overset{(a)}{=} \frac{1}{2^{n-k}}\prod_{l=1}^{n-k}(I+g_l) = P_\mathcal{S},$$

where (a) follows from re-ordering the factors in the products which can be done since $\mathcal{S}$ is abelian, and (b) from the i-th factor of the product trivially being $I + g_i$ if $a_i = 0$ and from $g_i^2 = I$ if $a_i = 1$. Thus, $P_\mathcal{S} E_i^\dagger E_j P_\mathcal{S} = P_\mathcal{S}^2 = P_\mathcal{S}$, which trivially satisfies the Knill-Laflamme conditions with $H = I$.

If $E_i^\dagger E_j \in \mathcal{G}_n \setminus N(\mathcal{S})$, then $E_i^\dagger E_j$ anti-commutes with at least one generator of $\mathcal{S}$, say g_1. Applying $E_i^\dagger E_j$ to $P_\mathcal{S}$ yields in this case

$$E_i^\dagger E_j P_\mathcal{S} = \frac{1}{2^{n-k}} E_i^\dagger E_j \prod_{l=1}^{n-k}(I+g_l) = \frac{1}{2^{n-k}}(I-g_1)E_i^\dagger E_j \prod_{l=2}^{n-k}(I+g_l).$$

Thus,

$$\begin{aligned} P_\mathcal{S} E_i^\dagger E_j P_\mathcal{S} &= \frac{1}{2^{n-k}}\prod_{l=1}^{n-k}(I+g_l)\left(\frac{1}{2^{n-k}}(I-g_1)E_i^\dagger E_j \prod_{m=2}^{n-k}(I+g_m)\right) \\ &\overset{(a)}{=} \left(\frac{1}{2^{n-k}}\right)^2 \prod_{l=2}^{n-k}(I+g_l)(I+g_1)(I-g_1)E_i^\dagger E_j \prod_{m=2}^{n-k}(I+g_m), \end{aligned}$$

where (a) follows from the Frobenius covariates commute due to the generators commute. Since

$$(I+g_1)(I-g_1) = I - g_1^2 = I - I = 0,$$

it follows that $P_\mathcal{S} E_i^\dagger E_j P_\mathcal{S} = 0$, which also trivially satisfies the Knill-Laflamme conditions with $H = 0$. This concludes the proof. ■

It should be explicitly noted that the correctable errors either let $|\psi_L\rangle$ remain in the mutual +1-eigenspace of $\mathcal{S}$, or it sendst $|\psi_L\rangle$ to some orthogonal subspace that is a mixture of ±1-eigenspaces, where the sign for each generator depends on the error corrupting $|\psi_L\rangle$.

Based on the stabilizer Knill-Laflamme conditions, it is possible to define a distance for the stabilizer code $V_\mathcal{S}$. Let the weight of elements of $\mathcal{G}_n$ be the number of non-identity elements in its tensor product, e.g., I has weight zero, while $X_1 X_2$ has weight two. The distance of $V_\mathcal{S}$, d, can then be defined to be the minimum weight of an element in $N(\mathcal{S}) \setminus \mathcal{S}$.

Now that the existence of a correction channel for stabilizer codes have been considered, the decoding process can be discussed.

6.2.4 Decoding

The first problem of decoding is to detect the occurred error. Since the correctable errors, $\mathcal{E}_C$, lie in some ±1-eigenspace of the generators of $\mathcal{S}$, the detection can simply be done by measuring all of the generators one by one with stabilizer measurements. This deterministically yields an error syndrome of ±1s, which indicate the occurred error, e.g., by a look-up table. More precisely, say the error $E \in \mathcal{E}_C$ occurs. Then the i-th bit in the error syndrome is +1 if E and g_i commute such that E leaves $|\psi\rangle_L$ in the +1-eigenspace of g_i, or it is −1 if E and g_i anti-commute such that E takes $|\psi\rangle_L$ to the −1-eigenspace of g_i. In neither of the cases does the measurement change the state since it already belongs to the given eigenspace. Explicitly, the probability of measuring +1 for generator g_i is

$$\begin{aligned} \Pr(+1) &= \langle\psi|_L E^\dagger (g_i)_+ E |\psi\rangle_L = \frac{1}{2}\langle\psi|_L E^\dagger (I+g_i) E |\psi\rangle_L = \frac{1}{2}(\langle\psi|_L E^\dagger E |\psi\rangle_L \pm \langle\psi|_L E^\dagger g_i E |\psi\rangle_L) \\ &= \begin{cases} 1, & \text{if } E \text{ and } g_i \text{ commute,} \\ 0, & \text{if } E \text{ and } g_i \text{ anti-commute.} \end{cases} \end{aligned}$$

The probability for measuring -1 follows analogously, however with the opposite conclusion. After detecting the occurred error, it can be corrected simply by applying the appropriate Pauli operator. If the code is non-degenerate, there is only one suitable operator, namely $E^\dagger$. If it is degenerate, Bob cannot distinguish between the possible errors, however applying one of them, e.g., $E^\dagger$, will correct the error.

Now that the theory of stabilizer codes has been presented, the entire coding process is illustrated for an important type of stabilizer code.

6.3 Steane Code

The framework for stabilizer codes is quite different from the classical setting, hence a completely new method for finding stabilizer codes is seemingly needed. However, it turns out that a class of stabilizer codes can be constructed from classical codes that satisfy a given property. The Calderbank-Shor-Steane (CSS) codes are examples of this as they simply are constructed by taking two classical codes, C_1 and C_2, and then putting their parity check matrices, H_1, H_2, into the check matrix, G, for the stabilizer code as follows

$$G = \left[\begin{array}{c|c} H_1 & 0 \\ 0 & H_2 \end{array}\right].$$

The idea of constructing G in this manner is to let one code correct for bit flips, while the other correct phase flips. Naturally, the length of the two codes must be the same for G to be a well-defined check matrix. They may however have varying dimension and distance, say C_1 is an $[n, k_1]$ code and C_2 an $[n, k_2]$ code. For G to define a stabilizer code, all rows must be independent. Clearly, the first $(n - k_1)$ are independent as they are constructed by H_1, which has rank $(n - k_1)$. Similarly, the last $(n - k_2)$ are independent. Thus, it must only be checked that the first $(n - k_1)$ rows are independent of the last $(n - k_2)$, which is when the corresponding symplectic inner product is zero. That is, when

$$0 = \left[\begin{array}{c|c} H_1 & 0 \end{array}\right] \Lambda \left[\begin{array}{c} 0^\top \\ \hline H_2^\top \end{array}\right] = \left[\begin{array}{c|c} H_1 & 0 \end{array}\right] \left[\begin{array}{c} H_2^\top \\ \hline 0^\top \end{array}\right] = H_1 H_2^\top .$$

Thus, $H_2^\top$ must be in the null-space of H_1, which is equivalent to $C_2^\perp \subseteq C_1$, where $C_2^\perp$ denotes the dual code of C_2. This implies that to construct an CSS code, it is sufficient to determine some dual-containing code, C. One such example is the $[7, 4, 3]$ Hamming code, which when used to construct an CSS code yields an $[[7, 1, 3]]$ stabilizer code typically known as the Steane code. By denoting the parity check matrix of the Hamming code H, the check matrix of the Steane code has the following form:

$$G = \left[\begin{array}{c|c} H & 0 \\ 0 & H \end{array}\right] = \left[\begin{array}{ccccccc|ccccccc} 1&0&0&1&1&0&1 & & & & & & & \\ 0&1&0&1&0&1&1 & & & &0& & & \\ 0&0&1&0&1&1&1 & & & & & & & \\ & & & & & & & 1&0&0&1&1&0&1 \\ & & &0& & & & 0&1&0&1&0&1&1 \\ & & & & & & & 0&0&1&0&1&1&1 \end{array}\right].$$

To use this for encoding, it must firstly be put in standard form, which is easily done as the left hand side already is in its standard form. The full standard form can be obtained by adding the fourth row to the fifth, then the fifth to the sixth, then the sixth to the fifth, and lastly permuting the last three columns once to the left. This yields the check matrix

$$G = \left[\begin{array}{c|c} H & 0 \\ 0 & H \end{array}\right] = \left[\begin{array}{ccccccc|ccccccc} 1&0&0&1&0&1&1 & & & & & & & \\ 0&1&0&1&1&1&0 & & & &0& & & \\ 0&0&1&0&1&1&1 & & & & & & & \\ & & & & & & & 0&1&1&1&0&0&1 \\ & & &0& & & & 1&1&0&0&1&0&1 \\ & & & & & & & 1&1&1&0&0&1&0 \end{array}\right],$$

where each of the coloured boxes correspond to one of the non-zero submatrices of the standard form of G. Notice that the submatrices B and C both are included in the zero block matrix on the right-hand side of G. It is now possible to directly read off the generators for the stabilizer code as well as the logical Pauli operators, where it must be remembered that the last three qubits were permuted when G was put in standard form:

$$g_1 = X_1X_4X_5X_7, \quad g_2 = X_2X_4X_6X_7, \quad g_3 = X_3X_5X_6X_7$$
$$g_4 = Z_2Z_3Z_4Z_5, \quad g_5 = Z_1Z_2Z_5Z_6, \quad g_6 = Z_1Z_2Z_3Z_7,$$

and

$$\bar{Z} = \begin{bmatrix} 0 & 0 & 0 & \big| & A_2^\top & 0 & I \end{bmatrix} = \begin{bmatrix} 0 & \cdots & 0 & \big| & 1 & 0 & 1 & 0 & 0 & 0 & 1 \end{bmatrix} = Z_1Z_3Z_5,$$
$$\bar{X} = \begin{bmatrix} 0 & E^\top & I & \big| & C^\top & 0 & 0 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 1 & 1 & 0 & 1 & \big| & 0 & \cdots & 0 \end{bmatrix} = X_4X_5X_6.$$

These generators are now used for encoding $|0\rangle_L$ by stabilizer measurements, hence $|\psi_0\rangle = |0\rangle^{\otimes 7}$ is initially prepared. This state is stabilized by $\tilde{\mathcal{S}} = \langle Z_1, \dots, Z_7\rangle$.

Firstly, g_1 is measured. Since it anti-commutes with Z_1, Z_4, Z_5, Z_7, the latter three of these generators is replaced by Z_1Z_i for $i \in \{4, 5, 7\}$ such that g_1 only anti-commutes with Z_1. Measuring g_1 then yields ±1 with equal probability. Say $+1$ is measured. The state $|\psi_0\rangle$ then changes to

$$|\psi_1\rangle = \frac{1}{\sqrt{2}}(I + g_1)\,|\psi_0\rangle = \frac{1}{\sqrt{2}}(|0000000\rangle + |1001101\rangle),$$

which is stabilized by $\mathcal{S}' = \langle g_1, Z_2, Z_3, Z_1Z_4, Z_1Z_5, Z_6, Z_1Z_7\rangle$.

Secondly, g_2 is measured similarly by first replacing all but one of the generators of $\mathcal{S}'$ that anti-commute with g_2, i.e., Z_1Z_4, Z_6, Z_1Z_7 are multiplied with Z_2. Measuring g_2 then yields ±1 with equal probability. Say it yields -1 such that $|\psi_1\rangle$ changes to

$$|\psi_2\rangle = \frac{1}{\sqrt{2}}(I - g_2)\,|\psi_1\rangle = \frac{1}{2}(|0000000\rangle + |1001101\rangle - |0101011\rangle - |1100110\rangle),$$

which is stabilized by $\mathcal{S}' = \langle g_1, -g_2, Z_3, Z_1Z_2Z_4, Z_1Z_5, Z_2Z_6, Z_1Z_2Z_7\rangle$.

Thirdly, measuring g_3 follows similarly. Say the outcome is -1. Then $|\psi_2\rangle$ evolves to

$$\begin{aligned}|\psi_3\rangle = \frac{1}{\sqrt{2}}(I - g_3)\,|\psi_2\rangle = \frac{1}{2\sqrt{2}}\big[\,&|0000000\rangle + |1001101\rangle - |0101011\rangle - |1100110\rangle \\ &- |0010111\rangle - |1011010\rangle + |0111100\rangle + |1110001\rangle\,\big],\end{aligned}$$

which is stabilized by $\mathcal{S}' = \langle g_1, -g_2, -g_3, Z_1Z_2Z_4, Z_1Z_3Z_5, Z_2Z_3Z_6, Z_1Z_2Z_3Z_7\rangle$.

Next, g_4 must be measured. However, since it commutes with all generators of $\mathcal{S}'$, it stabilizes $|\psi_3\rangle$. Particularly, it can replace $Z_1Z_2Z_4$ as a generator by multiplying it with $Z_1Z_3Z_5$. Similarly, g_5 can replace $Z_2Z_3Z_6$ by multiplying it with $Z_1Z_3Z_5$. The last two generators of $\mathcal{S}'$ are exactly g_6 and $\bar{Z}$. Thus, $\mathcal{S}' = \langle g_1, -g_2, -g_3, g_4, g_5, g_6, \bar{Z}\rangle$. This implies that measuring $g_4, g_5, g_6, \bar{Z}$ has no impact on $|\psi_3\rangle$.

The last step before obtaining $|0\rangle_L$ is to fix all the signs of the generators of $\mathcal{S}'$, which is done by determining the two elements in $\mathcal{G}_7$ that anti-commute with either g_2 or g_3, while commuting with all the other generators. It is easily seen that Z_2 satisfy the property for g_2, while Z_3 does for g_3. Hence, one obtain $|0\rangle_L$ by applying Z_2Z_3 to $|\psi_3\rangle$, which yields

$$\begin{aligned}|0_L\rangle = Z_2Z_3\,|\psi_3\rangle = \frac{1}{2\sqrt{2}}\big[\,&|0000000\rangle + |1001101\rangle + |0101011\rangle + |1100110\rangle \\ &+ |0010111\rangle + |1011010\rangle + |0111100\rangle + |1110001\rangle\big],\end{aligned}$$

which naturally is stabilized by $\mathcal{S} = \langle g_1, \dots, g_6, \bar{Z}\rangle$. It should be noted that this is not form typically given for the Steane code, e.g., in equation 10.78 of [NC10, p. 453], which merely is a result of writing G on a different form and thereby obtaining different generators for $\mathcal{S}$ as seen in Figure 10.6 in [NC10, p. 456]. In fact, the code derived in this section only differ from the form given in [NC10, p. 456] by having swapped the third and fourth qubit.

After having prepared $|0\rangle_L$, Alice can encode some unknown state $|\psi\rangle$ into $|\psi\rangle_L = \alpha\,|0\rangle_L + \beta\,|1\rangle_L$, however this is not explicitly done here since the procedure is straight-forward as described in Section 6.2.2.

Now that $|0\rangle_L$ has been encoded, it would be desirably to determine which errors it can correct. The elements of $N(\mathcal{S})$ must have an even number of Z overlaps with the X positions of g_1, g_2, g_3 in order to commute, and vice versa for g_4, g_5, g_6. The element with the smallest non-zero weight satisfying this has weight three, e.g., $\bar{X}$ or $\bar{Z}$. Since neither of these are in $\mathcal{S}$, it can be concluded that the Steane code is an $[[7, 1, 3]]$ code, hence it can correct an arbitrary single qubit error. An example of this is now considered.

Assume that a bit-phase flip has occurred on the forth qubit, i.e, Bob possesses the state $Z_4X_4\,|\psi\rangle_L$. The error can be detected by stabilizer measurements. Firstly, g_1 is measured. Since g_1 and Z_4X_4 anti-commute, the measurement yields -1 with certainty, yet it does not change the state. Similarly, measuring g_2 and g_4 yields -1. Since neither of g_3, g_5, g_6 has a non-identity element on the fourth qubit, they all commute with Z_4X_4, hence the measurements of these generators yield $+1$. Combining this, Bob obtains the error syndrome $(-1, -1, +1, -1, +1, +1)$. Since the first three generators detect phase flips and the last three detect bit flips, it can be deduced that a bit-phase flip has occurred. Furthermore, the position of the corrupted qubit can be determined based on which measurements that have yielded -1. Particularly, it cannot be in any other position than the fourth since neither the measurement of g_5 nor g_6 yielded -1. Thus, it can with certainty be deduced that the occurred error was a bit-phase flip on the fourth qubit, which naturally can be corrected by simply applying Z_4X_4 to $|\psi\rangle_L$.

The general theory of stabilizer codes have now been presented, culminating in a thorough example of encoding/decoding an important class of stabilizer codes. These CSS codes are constructed from classical codes, hence they are in principle quite easy to construct. However, the dual-containing requirement limits the power substantially. In the following chapter, the stabilizer codes are therefore generalised to another type of quantum code, which weakens this requirement.

7 | Entanglement-Assisted Codes

Even though stabilizer codes can be constructed from classical codes, the dual-containing requirement puts a limit on the classical codes that can be used to construct such codes. It turns out that this condition can be removed if Alice and Bob posses some suitable amount of EPR pairs among them. Since the types of codes created under this framework utilise the pre-existing entanglement, they are known as entanglement-assisted quantum error-correcting codes, henceforth simply EA codes. This chapter is primarily based on [BDH06] and [LB12].

It should be noted that the theory of EA codes is not limited to being a generalisation of stabilizer codes, however, it is the only framework considered in this thesis. It is commonly known as the entanglement-assisted stabilizer coding framework, however stabilizer is omitted henceforth such that it is referred to as the EA coding framework. This framework is now presented.

7.1 The Entanglement-Assisted Framework

In order to construct a $[[n, k]]$ stabilizer code, one has to determine an abelian subgroup of $\mathcal{G}_n$ with $n - k$ independent generators such that $-I$ is not included in the subgroup as described in Section 6.2. If one instead considers a non-abelian subgroup, $\mathcal{S}$, of $\mathcal{G}_n$, then it it still possible to construct a quantum code, not a stabilizer but an EA code, given that Alice and Bob share some suitable amount of pre-existing EPR pairs. The framework used to construct such EA codes is presented in this section based on a particular example originally given in [BDH06]. This is done to not only provide an example of an EA code, but also to make the description of the framework less abstract. The foundation for this example is the subgroup defined in Example 7.1.

Example 7.1: Subgroup for Entanglement-Assisted Code
Consider the subgroup $\mathcal{S}$ of $\mathcal{G}_4$ generated by the generators

$$g_1 = Z_1X_2Z_3, \quad g_2 = Z_1Z_2Z_4, \quad g_3 = X_1Y_2X_3, \quad g_4 = X_1X_2X_4. \tag{7.1}$$

Since Pauli matrices anti-commute with each other cf. Theorem B.7, a pair of generators of $\mathcal{S}$ anti-commute if they different Pauli matrices on a odd number of qubits. This is the case for all pairs of $\mathcal{S}$ except (g_2, g_3), which has different Pauli matrices on two qubits. Hence, all pairs of generators of $\mathcal{S}$ anti-commute except (g_2, g_3), which commute. This implies that $\mathcal{S}$ is a non-abelian subgroup. Thus, $V_{\mathcal{S}}$ is a trivial subspace cf. Theorem 6.2, which in turn implies that $\mathcal{S}$ does not define a stabilizer code.

Throughout the remainder of this section, everything presented as an example is related to Example 7.1, while everything not presented as an example holds generally, i.e., about a general non-abelian subgroup, $\mathcal{S}$, of $\mathcal{G}_n$.

Since $\mathcal{S}$ is assumed to be non-abelian, it does not generate a stabilizer code. The idea of the EA coding framework is then to utilise shared EPR pairs between Alice and Bob as additional qubits that is appended to $\mathcal{S}$ such that it becomes abelian, thus generating a stabilizer code, which is called an EA code since it requires shared entanglement between Alice and Bob. In other words, instead of considering $\mathcal{S}$ as a non-abelian subgroup of $\mathcal{G}_n$, it is considered an abelian subgroup of $\mathcal{G}_{n+c}$ for some suitable number c that describes how many EPR pairs Alice and Bob must share in order for $\mathcal{S}$ to be abelian in $\mathcal{G}_{n+c}$. The first task is therefore to determine c. In practice, this is done by firstly relating $\mathcal{S}$ to a simpler subgroup, $\mathcal{B}$, of $\mathcal{G}_n$, from which it is easier to determine an abelian subgroup. However, before determining a suitable $\mathcal{B}$, it is convenient to choose the generators of $\mathcal{S}$ such that they satisfy some given commutation relations.

Theorem 7.2: Decomposition of Generators
Let $\mathcal{S}$ be a subgroup of $\mathcal{G}_n$ with 2^m distinct elements up to the multiplicative factor. Then there exists a set of independent generators $A = \{\hat{Z}_1, \dots, \hat{Z}_l, \hat{X}_1, \dots, , \hat{X}_{m-l}\}$ with $m/2 \leq l \leq m$ satisfying

> that $\hat{Z}_i$ commute with all generators except $\hat{X}_i$, and vice versa, for all i. Let $A_I = \{\hat{Z}_{m-l+1}, \dots, \hat{Z}_l\}$ be generators of the 'isotropic' group $\mathcal{S}_I$, and $A_S = \{\hat{Z}_1, \dots, \hat{Z}_{m-l}, \hat{X}_1, \dots, \hat{X}_{m-l}\}$ be generators of the 'symplectic' group $\mathcal{S}_S$. Then $A = A_I \cup A_S$ generate $\mathcal{S}$, which by abusive of notation is denoted $\mathcal{S} = \langle \mathcal{S}_I, \mathcal{S}_S \rangle$. [BDH06, p. 437]

Before proving the result, it should be explicitly noted that the symplectic generators come in pairs.

Proof
The result is proven by describing a constructive algorithm that decomposes the initial generators of $\mathcal{S}$, i.e., $g = \{g_1, \dots, g_m\}$, into generators of $\mathcal{S}_I$ or $\mathcal{S}_S$. This is done by iteratively determining pairs of symplectic generators to add to A_S when possible.

The initialisation of the algorithm is: $g = \{g_1, \dots, g_m\}, A_S = \emptyset, A_I = \emptyset, k = 0$, where k is the iteration counter. The algorithm then contains the following steps:

i) Find any anti-commuting pair of generatios in g, say g_i and g_j. If necessary, swap indices of g_i and g_{2k+1} as well as g_j and g_{2k+2} such that g_{2k+1} and g_{2k+2} is anti-commuting. If no anti-commuting pair exists in g, then g is a set of isotropic generators. In that case, denote the elements of g as $\hat{Z}_{k+1}, \dots, \hat{Z}_{m-k}$, and let $A_I = \{\hat{Z}_{k+1}, \dots, \hat{Z}_{m-k}\}$. Afterwards, stop the algorithm.

ii) For g_{2k+1} and g_{2k+2} to be a symplectic pair, they must commute with all the other generators in g. For all generators $g_n \in g$ with $n \neq 2k+1, 2k+2$, g_n satisfy one of the following four commutation relations; 1) g_n commutes with both g_{2k+1} and g_{2k+2}, 2) g_n commutes with g_{2k+1} but anti-commutes with g_{2k+2}, 3) g_n anti-commutes with g_{2k+1} but commutes with g_{2k+2}, or 4) g_n anti-commutes with both g_{2k+1} and g_{2k+2}. Depending on which case holds true, g_n will commute with both g_{2k+1} and g_{2k+2} by applying the following mapping:

$$g_n \mapsto g_n = \begin{cases} g_n, & \text{if } g_n \text{ satisfy commutation relation 1,} \\ g_n g_{2k+1}, & \text{if } g_n \text{ satisfy commutation relation 2,} \\ g_n g_{2k+2}, & \text{if } g_n \text{ satisfy commutation relation 3,} \\ g_n g_{2k+1} g_{2k+2}, & \text{if } g_n \text{ satisfy commutation relation 4.} \end{cases}$$

Since this mapping only changes a generator of $\mathcal{S}$, but not $\mathcal{S}$ itself, it merely enables one to choose the generators of $\mathcal{S}$ such that they satisfy some desirable commutation properties, namely that g_n commute with both g_{2k+1} and g_{2k+2}. After the above mapping has been applied to all generators g_n, then g_{2k+1} and g_{2k+2} constitute a symplectic pair.

iii) Re-label the symplectic pair g_{2k+1} and g_{2k+2} as $\hat{Z}_{k+1}$ and $\hat{X}_{k+1}$, respectively. Then remove these generators from g, while appending them to A_S, i.e.,

$$g \mapsto g \setminus \{\hat{Z}_{k+1}, \hat{X}_{k+1}\}, \quad A_S \mapsto A_S \cup \{\hat{Z}_{k+1}, \hat{X}_{k+1}\}.$$

iv) Increase the iterations counter with one. If $g = \emptyset$, stop the algorithm. Otherwise, repeat from step 1.

The algorithm converges as it terminates when g no longer contains a pair of anti-commuting generators. Particularly, if the algorithm terminates after k iterations, k symplectic pairs and $m - 2k$ isotropic generators has been found. In the simplest case, the initial g consists only of commuting generators, in which case the algorithm terminates when $k = 0$. In the other extreme, the algorithm may continue until all generators have been put into a symplectic pair (if m is odd, an additional isotropic generator will remain). There are at most $m/2$ of such pairs, hence the algorithm stops when $k \leq \frac{m}{2}$. By construction, $l = m - k$, hence it follows that $\frac{m}{2} \leq l \leq m$, since $0 \leq k \leq \frac{m}{2}$, which concludes the proof. ■

It is noteworthy that the algorithm described in the proof based on group theoretic terminology could be described as a Gram-Schmidt process with respect to the symplectic inner product by using the vector representation of the Pauli matrices defined in (6.2) due to the commutation property given in (6.3). Thus, the algorithm is henceforth referred to as the symplectic Gram-Schmidt process.

The symplectic Gram-Schmidt process will now be applied to the generators of $\mathcal{S}$ defined in Example 7.1 in order to decompose it into isotropic and symplectic generators.

Example 7.3: Decomposing Generators of Subgroup
By using the generators given in (7.1), the initialisation of the symplectic Gram-Schmidt process is

$$g = \{g_1, g_2, g_3, g_4\} = \{Z_1X_2Z_3, Z_1Z_2Z_4, X_1Y_2X_3, X_1X_2X_4\}, A_S = A_I = \emptyset, k = 0.$$

One decomposition of g is then as follows. Since g_1 and g_2 anti-commute, they can be used as a symplectic pair without re-indexing the generators of g. To be a symplectic pair, they must both commute with g_3 and g_4. However, only g_2 and g_3 commute initially. Thus, g_3 is multiplied with g_2, while g_4 is multiplied with both g_1 and g_2, yielding the mappings

$$g_3 \mapsto g_3 = g_3g_2 = Y_1X_2X_3Z_4, \quad g_4 \mapsto g_4 = g_4g_1g_2 = X_1Z_2Z_3Y_4,$$

where it should be noted that the multiplicative factor has been ignored. Since g_1 and g_2 now both commute with g_3 and g_4, they constitute a symplectic pair, thus they are denoted $\hat{Z}_1, \hat{X}_1$, respectively. They are then removed from g and appended to A_S such that

$$g \mapsto g \setminus \{\hat{Z}_1, \hat{X}_1\} = \{g_3, g_4\} = \{Y_1X_2X_3Z_4, X_1Z_2Z_3Y_4\}, \quad A_S \mapsto \{\hat{Z}_1, \hat{X}_1\} = \{Z_1X_2Z_3, Z_1Z_2Z_4\}.$$

Since $g \neq \emptyset$, a new iteration is conducted with $k = 1$. This time, g contains two commuting generators, g_3, g_4, hence they are isotropic generators and are denoted $\hat{Z}_2, \hat{Z}_3$, respectively. Thus, the decomposition yields $A_S = \{\hat{Z}_1, \hat{X}_1\}, A_I = \{\hat{Z}_2, \hat{Z}_3\}$ with

$$\hat{Z}_1 = Z_1X_2Z_3, \quad \hat{X}_1 = Z_1Z_2Z_4, \quad \hat{Z}_2 = Y_1X_2X_3Z_4, \quad \hat{Z}_3 = X_1Z_2Z_3Y_4 \tag{7.2}$$

such that $\mathcal{S}_S = \langle \hat{Z}_1, \hat{X}_1 \rangle, \mathcal{S}_I = \langle \hat{Z}_2, \hat{Z}_3 \rangle$, and $\mathcal{S} = \langle \mathcal{S}_I, \mathcal{S}_S \rangle$.

After the generators of $\mathcal{S}$ have been decomposed into isotropic and symplectic generators with the symplectic Gram-Schmidt process, it is easier to relate $\mathcal{S}$ to a simple subgroup $\mathcal{B}$ of $\mathcal{G}_n$. The simpler group $\mathcal{B}$ is chosen such that its generators can be decomposed similarly as those of $\mathcal{S}$. That is, the number of generators of $\mathcal{B}_I$ is equal to that of $\mathcal{S}_I$ such that an isomorphism between the two groups $\mathcal{B}_I, \mathcal{S}_I$ can be defined. Similarly, an isomorphism between $\mathcal{B}_S$ and $\mathcal{S}_S$ is defined. Combining these, one simply defines an isomorphism between $\mathcal{B}$ and $\mathcal{S}$ such that the commutation relations of both groups are similar, i.e., $\mathcal{B}$ and $\mathcal{S}$ has equally many symplectic pairs and equally many isotropic generators. In that case $\mathcal{B}$ and $\mathcal{S}$ are said to be isomorphic, and the following theorem describes the relation between the generators of the two groups.

Theorem 7.4: Unitary Mapping between Isomorphic Groups
Let $\mathcal{B}$ and $\mathcal{S}$ be isomorphic subgroups of $\mathcal{G}_n$. Then there exists a unitary operator $U \in \operatorname{End}\left((\mathbb{C}^2)^{\otimes n}\right)$ such that for all $b \in \mathcal{B}$, there exists an $s \in \mathcal{S}$ satisfying $b = UsU^\dagger$ up to a multiplicative factor.

[BDH06, p. 437]

Proof
The result is proven by describing a constructive method for determining a unitary operator achieving the desired mapping.

Let the generators of $\mathcal{S}$ be decomposed into symplectic and isotropic generators such that it has the form $\{\hat{Z}_1, \dots, \hat{Z}_l, \hat{X}_1, \dots, \hat{X}_{m-l}\}$ for $m/2 \leq l \leq m$ cf. Theorem 7.2. Since $\mathcal{B}$ is isomorphic to $\mathcal{S}$, it has a similar decomposition, which is denoted $\{Z'_1, \dots, Z'_l, X'_1, \dots, X'_{m-l}\}$. These two sets of generators are now expanded to a set of generators that generate $\mathcal{G}_n$. For the generators of $\mathcal{S}$, this is achieved by the following two steps:

i) Append generators $\hat{X}_{m-l+1}, \dots, \hat{X}_l$ to the set of generators for $\mathcal{S}$ such that it consists of l symplectic pairs. Naturally, $\hat{X}_i$ for $i = m - l + 1, \dots, l$ must commute with all the other generators except $\hat{Z}_i$.

ii) Since $|\mathcal{G}_n| = 4^n$ when the multiplicative factor is ignored, it can by Theorem C.10 be generated by $\log_2(|4^n|) = 2n$ generators. Thus, $n - l$ additional symplectic pairs, $(\hat{Z}_{l+1}, \hat{X}_{l+1}), \dots, (\hat{Z}_n, \hat{X}_n)$ is appended to the set of generators for S. Again, they must satisfy the appropriate commutation relations with the other generators.

Analogously, the set of generators for $\mathcal{B}$ is expanded to $\{Z'_1, \ldots, Z'_n, X'_1, \ldots, X'_n\}$. Thus, two sets of generators for $\mathcal{G}_n$ up to a multiplicative factor is obtained. Since the multiplicative factor for the elements is irrelevant, it can without loss of generality be assumed that all generators are Hermitian, hence having ± 1 as eigenvalues.

Now, define $|\hat{0}\rangle$ to be in the simultaneous +1-eigenspace of $\{\hat{Z}_k\}_k$. Furthermore, for $i \in \{0,1\}^n$ with $i = i_1, \ldots, i_n$ where $i_1, \ldots, i_n \in \{0,1\}$, define $|\hat{i}\rangle = \hat{X}_1^{i_1} \cdots \hat{X}_n^{i_n} |\hat{0}\rangle$. For any $k = 1, \ldots, n$ and any $i \in \{0,1\}^n$, it then follows that

$$\begin{aligned} \hat{Z}_k |\hat{i}\rangle &= \hat{Z}_k \hat{X}_1^{i_1} \cdots \hat{X}_n^{i_n} |\hat{0}\rangle \overset{(a)}{=} (-1)^{\delta(i_k,1)} \hat{X}_1^{i_1} \cdots \hat{X}_n^{i_n} \hat{Z}_k |\hat{0}\rangle \overset{(b)}{=} (-1)^{\delta(i_k,1)} \hat{X}_1^{i_1} \cdots \hat{X}_n^{i_n} |\hat{0}\rangle \\ &= (-1)^{\delta(i_k,1)} |\hat{i}\rangle , \end{aligned} \tag{7.3}$$

where (a) follows from $\hat{Z}_k$ commuting with all generators in $\{\hat{X}_j\}_j$ except $\hat{X}_k$, and (b) from $|\hat{0}\rangle$ being in the +1-eigenspace of $\hat{Z}_k$ by construction. Hence, $|\hat{i}\rangle$ is in the +1-eigenspace of $\hat{Z}_k$ if the k-th bit of i is zero, and in the −1-eigenspace if it is one. Since $\hat{Z}_k$ is unitary, this implies that $|\hat{i}\rangle$ is orthogonal to $|\hat{0}\rangle$ for any $i \in \{0,1\}^n$ (except the zero bit string which corresponds to $|\hat{0}\rangle$). Since k and i was arbitrarily chosen, it follows that $\{|\hat{i}\rangle\}_i$ constitute an orthonormal basis for $(\mathbb{C}^2)^{\otimes n}$. Similarly, it follows that

$$\hat{X}_k |\hat{i}\rangle = \hat{X}_k \hat{X}_1^{i_1} \cdots \hat{X}_n^{i_n} |\hat{0}\rangle \overset{(a)}{=} \hat{X}_1^{i_1} \cdots \hat{X}_{k-1}^{i_{k-1}} \hat{X}_k^{i_k+1} \hat{X}_{k+1}^{i_{k+1}} \cdots \hat{X}_n^{i_n} |\hat{0}\rangle \overset{(b)}{=} |\widehat{(i+k)}\rangle , \tag{7.4}$$

where (a) follows from $\hat{X}_k$ commuting with all other generators in the set $\{\hat{X}_j\}_j$, while the state after (b) should be read as the state with bit string $(i+k)$ mod 2, where k simply is the bit string of all zeros except a one at the k-th bit (this is abusive notation, since k then both denote a single index in $\hat{X}_k$ and a bit string with a single one positioned at the k-th bit, however it should be clear from context which is used).

Analogously, define an orthonormal basis $\{|i'\rangle\}_i$ from the set of generators $\{Z'_1, \ldots, Z'_n, X'_1, \ldots, X'_n\}$ where results analogously to (7.3) and (7.4) naturally hold.

Based on these two orthonormal bases, one can define a unitary operator U as

$$U = \sum_{i \in \{0,1\}^n} |i'\rangle \langle \hat{i}| .$$

Its effect on elements of $\mathcal{S}$ is now examined, for which is suffice to only consider its generators. The effect on generators in $\{\hat{Z}_k\}_k$ is:

$$\begin{aligned} U\hat{Z}_k U^\dagger &= \sum_{i,j \in \{0,1\}^n} |i'\rangle \langle \hat{i}|\hat{Z}_k|\hat{j}\rangle \langle j'| \overset{(a)}{=} \sum_{i,j \in \{0,1\}^n} (-1)^{\delta(j_k,1)} |i'\rangle \langle \hat{i}|\hat{j}\rangle \langle j'| \overset{(b)}{=} \sum_{i \in \{0,1\}^n} (-1)^{\delta(i_k,1)} |i'\rangle \langle i'| \\ &\overset{(c)}{=} \sum_{i \in \{0,1\}^n} Z'_k |i'\rangle \langle i'| \overset{(d)}{=} Z'_k, \end{aligned}$$

where (a) and (c) follows from (7.3), while (b) and (d) from $\{|\hat{i}\rangle\}_i$ and $\{|i'\rangle\}_i$ constituting orthonormal bases.

Similarly, applying U to a generator from $\{\hat{X}_k\}_k$ yields

$$\begin{aligned} U\hat{X}_k U^\dagger &= \sum_{i,j \in \{0,1\}^n} |i'\rangle \langle \hat{i}|\hat{X}_k|\hat{j}\rangle \langle j'| \overset{(a)}{=} \sum_{i,j \in \{0,1\}^n} |i'\rangle \langle \hat{i}|\widehat{(j+k)}\rangle \langle j'| \overset{(b)}{=} \sum_{i \in \{0,1\}^n} |i'\rangle \langle (i+k)'| \\ &\overset{(c)}{=} \sum_{i \in \{0,1\}^n} |i'\rangle \langle i'| \, (X'_k)^\dagger \overset{(d)}{=} (X'_k)^\dagger \overset{(e)}{=} X'_k, \end{aligned}$$

where (a) and (c) follows from (7.4), (b) and (d) once again follow from $\{|\hat{i}\rangle\}_i$ and $\{|i'\rangle\}_i$ constituting orthonormal bases, and (e) from X'_k being Hermitian.

Since these two results hold for any k, it particularly holds that the generators of $\mathcal{S}$, $\{\hat{Z}_1, \ldots, \hat{Z}_l, \hat{X}_1, \ldots, \hat{X}_{m-l}\}$, maps directly to the generators of $\mathcal{B}$, $\{Z'_1, \ldots, Z'_l, X'_1, \ldots, X'_{m-l}\}$ by applying U appropriately, which completes the proof. ■

Said in other words, Theorem 7.4 simply states that since $\mathcal{S}$ and $\mathcal{B}$ are isomorphic, they are unitarily equivalent up to a multiplicative factor. The first problem is therefore to find a subgroup of $\mathcal{G}_n$ that is isomorphic to $\mathcal{S}$. Before discussing this for a general subgroup, it is done for the one defined in Example 7.1.

Example 7.5: Isomorphic Subgroups
By Example 7.3, the generators of $\mathcal{S}$ can be decomposed into

$$\hat{Z}_1 = Z_1X_2Z_3, \quad \hat{X}_1 = Z_1Z_2Z_4, \quad \hat{Z}_2 = Y_1X_2X_3Z_4, \quad \hat{Z}_3 = X_1Z_2Z_3Y_4,$$

such that $\mathcal{S}_S = \langle \hat{Z}_1, \hat{X}_1 \rangle$ and $\mathcal{S}_I = \langle \hat{Z}_2, \hat{Z}_3 \rangle$ as stated in (7.2). That is, $\mathcal{S}$ contains a symplectic pair of generators and two isotropic generators. The problem is therefore to find four generators of $\mathcal{G}_4$ such that they all commute except one pair, which anti-commute. It is in fact trivial to find such a group since the commuting generators can be chosen such that they act non-trivially on different qubits. Perhaps the simplest generators satisfying this is

$$g'_1 = Z_1, \quad g'_2 = X_1, \quad g'_3 = Z_2, \quad g'_4 = Z_3. \tag{7.5}$$

Let $\mathcal{B}$ denote the subgroup generated by these generators. Then $\mathcal{B}_S = \langle Z_1, X_1 \rangle$ and $\mathcal{B}_I = \langle Z_2, Z_3 \rangle$, hence $\mathcal{B}$ and $\mathcal{S}$ are isomorphic.

By using the same idea as described in Example 7.5, it is trivial to determine an isomorphic subgroup to any subgroup $\mathcal{S}$ of $\mathcal{G}_n$. Assume that $\mathcal{S}$ is defined by $m \leq n$ generators that has been decomposed with the symplectic Gram-Schmidt process such that $\mathcal{S}$ has c symplectic pairs and $s = m - 2c$ isotropic generators. That is, $\mathcal{S}$ can be defined on the following form

$$\mathcal{S} = \langle \hat{Z}_1, \ldots, \hat{Z}_{c+s}, \hat{X}_1, \ldots, \hat{X}_c \rangle . \tag{7.6}$$

A simple example of a subgroup $\mathcal{B}$ of $\mathcal{G}_n$ that is isomorphic to $\mathcal{S}$ is defined as

$$\mathcal{B} = \langle Z_1, \ldots, Z_{c+s}, X_1, \ldots, X_c \rangle , \tag{7.7}$$

where the first c Z operators are part of a symplectic pair with the corresponding X operator, while the last s Z operators are isotropic generators. Notice that the prime has been removed from the generators of $\mathcal{B}$ compared to the proof of Theorem 7.4 for clarity since $Z'_1 = Z_1, X'_1 = X_1$, and so on in this particular case. In fact, the notation for the generators of $\mathcal{S}$ used in Theorem 7.2 and (7.6) is chosen to indicate the isomorphism between $\hat{Z}_1$ and Z_1, $\hat{X}_1$ and X_1, and so on.

To gain a better understanding of the unitary equivalence of $\mathcal{S}$ and $\mathcal{B}$, the unitary operator, U, that cf. Theorem 7.4 relates the two groups is examined. This is done by determining U for the two isomorphic subgroups given in Example 7.5.

Example 7.6: Unitary Equivalence of the Isomorphic Subgroups
The method for determining U that relates $\mathcal{S}$ and $\mathcal{B}$ is the one described in the proof of Theorem 7.4. Thus, the two set of generators for $\mathcal{S}$ and $\mathcal{B}$, i.e.,

$$\{\hat{Z}_1, \hat{Z}_2, \hat{Z}_3, \hat{X}_1\} = \{Z_1X_2Z_3, Y_1X_2X_3Z_4, X_1Z_2Z_3Y_4, Z_1Z_2Z_4\}, \tag{7.8}$$

$$\{Z_1, Z_2, Z_3, X_1\}, \tag{7.9}$$

respectively, must first be expanded to two sets of generators for $\mathcal{G}_4$ up to a multiplicative factor, whereafter two orthonormal bases for $(\mathbb{C}^2)^{\otimes 4}$ can be found. Since this is easiest for $\mathcal{B}$, this is done first to illustrate the idea.

First, the two isotropic generators of $\mathcal{B}$, Z_2 and Z_3, must be paired with an anti-commuting generator. Clearly, X_2 and X_3 satisfies this, respectively. Then, a fourth symplectic pair must be determined, which again is easily found to be Z_4, X_4 since no other generators act on the forth qubit. Thus, the set of generators of $\mathcal{B}$ given in (7.9) can be expanded to the following set of generators which generate $\mathcal{G}_4$ up to a multiplicative factor:

$$\{Z_1, Z_2, Z_3, Z_4, X_1, X_2, X_3, X_4\}.$$

Since the $Z\left|0\right\rangle = \left|0\right\rangle$ as shown in Example 2.10, it follows that the simultaneous +1-eigenvector of $\{Z_i\}_i$ is $\left|0000\right\rangle$. Applying, say X_2, to this state simply yields $\left|0100\right\rangle$. Thus, the orthonormal basis found by expanding the set of generators for $\mathcal{B}$ is simply $\{\left|i\right\rangle\}_i$ for $i \in \{0,1\}^4$, i.e., the canonical basis states for $(\mathbb{C}^2)^{\otimes 4}$. This also illustrate why choosing $\mathcal{B}$ as the group generated by the generators given in (7.9) is convenient.

The same procedure is now applied to the generators of $\mathcal{S}$, however, it requires substantially more effort. To find anti-commuting generators to the two isotropic generators of $\mathcal{S}$, the check matrix corresponding to (7.8) is utilised. It is given as

$$G = \begin{bmatrix} r(\hat{Z}_1) \\ r(\hat{Z}_2) \\ r(\hat{Z}_3) \\ r(\hat{X}_1) \end{bmatrix} = \left[\begin{array}{cccc|cccc} 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 \end{array}\right].$$

The aim is to find two generators, one which commutes with all but the second row, and one which commutes with all but the third row. By Theorem 6.5, such an element exists, and can be obtained by simply finding the generators in $\{\hat{X}_i\}_i$ that satisfies

$$G\Lambda r(\hat{X}_i)^\top = e_i,$$

where $i = 2, 3$ depending on the isotropic generators being considered. Inserting G into this expression yields

$$e_i = G\Lambda r(\hat{X}_i)^\top = \left[\begin{array}{cccc|cccc} 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 1 & 0 & 0 & 1 \\ 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \end{array}\right] r(\hat{X}_i)^\top.$$

Since the second last column of $G\Lambda$ is e_2 as seen above, it easily follows that $G\Lambda r(\hat{X}_2)^\top = e_2$ is satisfied by

$$r(\hat{X}_2) = \left[\begin{array}{cccc|cccc} 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{array}\right] \implies \hat{X}_2 = Z_3.$$

Similarly, the last column of $G\Lambda = e_3$, hence it follows that $G\Lambda r(\hat{X}_3)^\top = e_3$ is satisfied by

$$r(\hat{X}_3) = \left[\begin{array}{cccc|cccc} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{array}\right] \implies \hat{X}_3 = Z_4.$$

In principle, one should extend G with $r(\hat{X}_2)$ before determining $\hat{X}_3$ to make sure they also commute, however, this is easily satisfied in this case since $\hat{X}_2$ and $\hat{X}_3$ act non-trivially on different qubits. The next step is to find an additional symplectic pair, where the check matrix again is beneficial. This time, it is extended to also include $\hat{X}_2, \hat{X}_3$. It is thus given by

$$G = \left[\begin{array}{cccc|cccc} 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{array}\right] \implies G\Lambda = \left[\begin{array}{cccc|cccc} 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 1 & 0 & 0 & 1 \\ 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{array}\right]$$

Firstly, $\hat{Z}_4$ is determined by finding a vector that commutes with all rows of G, i.e., $G\Lambda r(\hat{Z}_4)^\top = 0$ must be satisfied. From the bottom two rows, it can e.g., be seen that the Pauli operator on the third and fourth qubit must be either the identity or Z. However, it is easily verifiable that summing up the first, second, sixth, and eighth columns of $G\Lambda$ yields the zero vector, which means that a solution is determined by

$$r(\hat{Z}_4) = \left[\begin{array}{cccc|cccc} 1 & 1 & 0 & 0 & 0 & 1 & 0 & 1 \end{array}\right] \implies \hat{Z}_4 = X_1 Y_2 Z_4.$$

To find $\hat{X}_4$, the check matrix can be expanded to include $r(\hat{Z}_4)$ in the fourth row and then solving $G\Lambda r(\hat{X}_4)^\top = e_4$. However, by using $G\Lambda$ as it is, one can also determine solve for $\hat{X}_4$ satisfying $G\Lambda r(\hat{X}_4) = 0$ such that $\hat{X}_4$ commute with all generators but $\hat{Z}_4$, and thereafter checking whether it anti-commutes with $\hat{Z}_4$. Since the fifth, seventh and eight columns of $G\Lambda$ sum to 0, it follows that $G\Lambda r(\hat{X}_4)^\top = 0$ is satisfied by

$$r(\hat{X}_4) = \left[\begin{array}{cccc|cccc} 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 \end{array}\right] \implies \hat{X}_4 = Z_1 Z_3 Z_4.$$

Thus, $\hat{X}_4$ certainly commutes with all generators except $\hat{Z}_4$. Since $\hat{X}_4$ and $\hat{Z}_4$ only have anti-commuting operators on the first qubit, they in fact anti-commute. Thus, the set of generators of $\mathcal{S}$ given in (7.8) can be expanded to the following set of generators that generate $\mathcal{G}_4$ up to the multiplicative factor:

$$\{\hat{Z}_1, \ldots, \hat{Z}_4, \hat{X}_1, \ldots, \hat{X}_4\} = \{Z_1X_2Z_3, Y_1X_2X_3Z_4, X_1Z_2Z_3Y_4, X_1Y_2Z_4, Z_1Z_2Z_4, Z_3, Z_4, Z_1Z_3Z_4\}.$$

To determine the simultaneous +1-eigenvector of $\{\hat{Z}_i\}_i$, the effect of the operators on $|0000\rangle$ is examined. Firstly,

$$\hat{Z}_1 |0000\rangle = Z_1X_2Z_3 |0000\rangle = |0100\rangle,$$

from which it can be concluded that

$$\hat{Z}_1 \left(\frac{1}{\sqrt{2}} (|0000\rangle + |0100\rangle \right) = \frac{1}{\sqrt{2}} (|0000\rangle + |0100\rangle),$$

implying that $(|0000\rangle + |0100\rangle)/\sqrt{2}$ is in the +1-eigenspace of $\hat{Z}_1$. Similarly,

$$\hat{Z}_2 |0000\rangle = i |1110\rangle, \quad \hat{Z}_3 |0000\rangle = i |1001\rangle, \quad \hat{Z}_4 |0000\rangle = i |1100\rangle.$$

By considering the effect of $\{\hat{Z}_i\}_i$ on all canonical basis states and taking the intersection of those +1-eigenspaces, it follows that the simultaneous +1-eigenvector of $\{\hat{Z}_i\}_i$ is given as

$$\begin{aligned} |\widehat{(0000)}\rangle = \frac{1}{4} \Big[& |0000\rangle + |0001\rangle + |0010\rangle + |0011\rangle + |0100\rangle + |0101\rangle - |0110\rangle - |0111\rangle \\ & i(-|1000\rangle + |1001\rangle + |1010\rangle - |1011\rangle + |1100\rangle - |1101\rangle + |1110\rangle - |1111\rangle) \Big]. \end{aligned}$$

Since every generator in $\{\hat{X}_i\}_i$ only consists of Z operators, it is easy to see their effects on $|\hat{0}\rangle$ since it flips the sign of basis states if and only if it has an odd number of ones in places where $\hat{X}_i$ has a Z operator. As an example,

$$\begin{aligned} |\widehat{(1010)}\rangle &= \hat{X}_1 \hat{X}_3 |\hat{0}\rangle = Z_1Z_2Z_4Z_4 |\hat{0}\rangle = Z_1Z_2 |\hat{0}\rangle \\ &= \frac{1}{4} \Big[|0000\rangle + |0001\rangle + |0010\rangle + |0011\rangle - |0100\rangle - |0101\rangle + |0110\rangle + |0111\rangle \\ &\qquad i(|1000\rangle - |1001\rangle - |1010\rangle + |1011\rangle + |1100\rangle - |1101\rangle + |1110\rangle - |1111\rangle) \Big]. \end{aligned}$$

Considering all such states yields the orthonormal basis $\{\hat{\imath}\}_i$ for $i \in \{0,1\}^4$, which clearly is a less trivial orthonormal basis compared to the canonical basis obtained by expanding the generators of $\mathcal{B}$.

Now that the two bases has been found, the unitary operator U can be defined by

$$U = \sum_{i \in \{0,1\}^4} |i\rangle \langle \hat{\imath}|.$$

Since $\{|i\rangle\}_i$ is the canonical basis, U is obtained by simply stacking the basis states of $\{|\hat{\imath}\rangle\}_i$ on top of each other as rows in the appropriate order. Explicitly, U is in this case given as

$$U = \frac{1}{4}\begin{bmatrix}
1 & 1 & 1 & 1 & 1 & 1 & -1 & -1 & -i & i & i & -i & i & -i & i & -i \\
1 & -1 & -1 & 1 & 1 & -1 & 1 & -1 & i & i & i & i & -i & -i & i & i \\
1 & -1 & 1 & -1 & 1 & -1 & -1 & 1 & -i & -i & i & i & i & i & i & i \\
1 & 1 & -1 & -1 & 1 & 1 & 1 & 1 & i & -i & i & -i & -i & i & i & -i \\
1 & 1 & -1 & -1 & 1 & 1 & 1 & 1 & -i & i & -i & i & i & -i & -i & i \\
1 & -1 & 1 & -1 & 1 & -1 & -1 & 1 & i & i & -i & -i & -i & -i & -i & -i \\
1 & -1 & -1 & 1 & 1 & -1 & 1 & -1 & -i & -i & -i & -i & i & i & -i & -i \\
1 & 1 & 1 & 1 & 1 & 1 & -1 & -1 & i & -i & -i & i & -i & i & -i & i \\
1 & -1 & 1 & -1 & -1 & 1 & 1 & -1 & i & i & -i & -i & i & i & i & i \\
1 & 1 & -1 & -1 & -1 & -1 & -1 & -1 & -i & i & -i & i & -i & i & i & -i \\
1 & 1 & 1 & 1 & -1 & -1 & 1 & 1 & i & -i & -i & i & i & -i & i & -i \\
1 & -1 & -1 & 1 & -1 & 1 & -1 & 1 & -i & -i & -i & -i & -i & -i & i & i \\
1 & -1 & -1 & 1 & -1 & 1 & -1 & 1 & i & i & i & i & i & i & -i & -i \\
1 & 1 & 1 & 1 & -1 & -1 & 1 & 1 & -i & i & i & -i & -i & i & -i & i \\
1 & 1 & -1 & -1 & -1 & -1 & -1 & -1 & i & -i & i & -i & i & -i & -i & i \\
1 & -1 & 1 & -1 & -1 & 1 & 1 & -1 & -i & -i & i & i & -i & -i & -i & -i
\end{bmatrix}. \tag{7.10}$$

It then follows from the proof of Theorem 7.4 that $Z_i = U\hat{Z}_iU^\dagger$ and $X_i = U\hat{X}_iU^\dagger$ for $i = 1, \ldots, 4$ up to a multiplicative factor.

For general subgroups $\mathcal{S}$ and $\mathcal{B}$ of $\mathcal{G}_n$ that are isomorphic and have decompositions as in (7.8) and (7.9), the method for determining the unitary operator relating $\mathcal{S}$ and $\mathcal{B}$ described in Example 7.6 can be used. The simple group $\mathcal{B}$ can be extended to $\{Z_1, \ldots, Z_n, X_1, \ldots, X_n\}$, wherefrom it with the same reasoning as in Example 7.6 follows that the corresponding orthonormal basis is the canonical basis of $(\mathbb{C}^2)^{\otimes n}$. It is, however, hard to say anything in general about the orthonormal basis, $\{|\hat{i}\rangle\}_i$, generated from extending $\mathcal{S}$. In the end, the rows of U correspond exactly to the states in the basis $\{|\hat{i}\rangle\}_i$. This again illustrates why it is convenient to choose the generators of $\mathcal{B}$ as in (7.9).

Having related $\mathcal{S}$ to the simple group $\mathcal{B}$, one can begin to construct a stabilizer code. Since $\mathcal{B}$ is a simpler group than $\mathcal{S}$, a stabilizer code is firstly constructed from $\mathcal{B}$. Since this group is non-abelian due to it being isomorphic to the non-abelian subgroup $\mathcal{S}$, it does however not initially generate a stabilizer code. The idea of the EA coding framework is then, as previously claimed, that this hurdle can be overcome by expanding $\mathcal{B}$ to a subgroup, $\mathcal{B}^e$, of $\mathcal{G}_{n+c}$ such that $\mathcal{B}^e$ is abelian. By doing so, $\mathcal{B}^e$ can be used to construct a stabilizer code. Before discussing the idea generally, it is firstly considered for $\mathcal{B}$ defined in Example 7.5.

Example 7.7: Stabilizer Code From Expanded Isomorphic Subgroup
The generators of $\mathcal{B}$ defined in (7.5) are $\{Z_1, X_1, Z_2, Z_3\}$, where $\mathcal{B}_S = \langle Z_1, X_1\rangle$ and $\mathcal{B}_I = \langle Z_2, Z_3\rangle$. The anti-commutativity of the symplectic pair can be fixed by applying appropriate operators to these generators on qubits that are not affected by any generators. In this case, this can be achieved by considered the generators $\{Z_1Z_4, X_1X_4, Z_2, Z_3\}$. By letting $\mathcal{B}$ be defined by these generators, $\mathcal{B}$ is an abelian subgroup of $\mathcal{G}_4$. It will, nonetheless, no longer be isomorphic to $\mathcal{S}$, which ruin the whole idea of $\mathcal{B}$. Instead, assume that $\mathcal{B}$ can be extended to $\mathcal{B}^e$ by having five qubits available. The above idea can then be used to define the abelian subgroup $\mathcal{B}^e$ of $\mathcal{G}_5$ by the generators

$$Z_1^e = Z_1Z_5, \quad X_1^e = X_1X_5, \quad Z_2^e = Z_2, \quad Z_3^e = Z_3. \tag{7.11}$$

In this case, $\mathcal{B}^e$ is easily seen to be an abelian subgroup of $\mathcal{G}_5$ since each pair of generators have anti-commuting Pauli operators on either zero or two qubits. If $\mathcal{B}^e$ is restricted to the first four qubits, then $\mathcal{B}$ is obtained, which then naturally still is isomorphic to $\mathcal{S}$. Hence, the idea of the generators of $\mathcal{B}^e$ is to simply 'copy' the Pauli operator on the first qubit to the fifth in order to resolve the anti-commutative of $\mathcal{B}$.

Since all generators of $\mathcal{B}^e$ defined in (7.11) commute and are Hermitian, they define a stabilizer code cf. Theorem 6.6, namely a $[[5, 1]]$ stabilizer code. By following the procedure in Section 6.2.2, the code can be constructed in standard form. The check matrix corresponding to (7.11) is

$$G = \begin{bmatrix} r(X_1^e) \\ r(Z_1^e) \\ r(Z_2^e) \\ r(3_1^e) \end{bmatrix} = \left[\begin{array}{ccccc|ccccc} 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \end{array}\right].$$

By permuting the last four columns in each block of G once to the right, which corresponds to re-labelling the corresponding qubits, yields G in standard form as described in (6.5):

$$G = \left[\begin{array}{ccccc|ccccc} 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{array}\right]$$

The two logical Pauli operators $\bar{Z}$ and $\bar{X}$ can be obtained from G by using (6.6) and (6.7), respectively. Doing so yields

$$r(\bar{Z}) = \left[\begin{array}{ccc|ccc} 0 & 0 & 0 & A_2^\top & 0 & 1 \end{array}\right] = \left[\begin{array}{ccccc|ccccc} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{array}\right],$$

which implies that $\bar{Z} = Z_4$ since the fifth column of each block of G corresponds to the forth qubit due to permutation performed to put G into standard form. Similarly

$$r(\bar{X}) = \left[\begin{array}{ccc|ccc} 0 & E^\top & 1 & 0 & 0 & 0 \end{array}\right] = \left[\begin{array}{ccccc|ccccc} 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \end{array}\right],$$

which for the same reasoning as above corresponds to $\bar{X} = X_4$.

The standard form of the logical ground state of the stabilizer code generated by $\mathcal{B}^e$ is then the simultaneous +1-eigenspace of the operators $\{Z_1^e, X_1^e, Z_2^e, Z_3^e, \bar{Z}\} = \{Z_1Z_5, X_1X_5, Z_2, Z_3, Z_4\}$. The first generator forces the sign of the first and fifth qubit to be identical, while the second forces the amplitude of the first and fifth qubit to be identical. The last three forces the second, third, and forth qubits to be zero, respectively. Combining these considerations implies that

$$|0\rangle_L = \frac{1}{\sqrt{2}}(|00000\rangle + |10001\rangle), \quad |1\rangle_L = \frac{1}{\sqrt{2}}(|00010\rangle + |10011\rangle), \tag{7.12}$$

where the logical excited state is obtained by simply flipping the forth qubit of the ground state according to $\bar{X}$. Using this to encode a qubit on the form $|\psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle$ yields a logical state on the form

$$\begin{aligned} |\psi\rangle_L = \alpha\,|0\rangle_L + \beta\,|1\rangle_L &= \frac{1}{\sqrt{2}}(\alpha\,|00000\rangle + \beta\,|00010\rangle + \alpha\,|10001\rangle + \beta\,|10011\rangle) \\ &= \frac{1}{\sqrt{2}}(|000\psi 0\rangle + |100\psi 1\rangle). \end{aligned} \tag{7.13}$$

By inspection, it is seen that the first and last bit of the computational basis states defining $|\psi\rangle_L$ must be the same, which is consistent with the discussion made to determine the form of the logical ground state. By re-indexing the qubits such that all qubits is permuted one to the right, the logical state can be written as

$$|\psi\rangle_L = \frac{1}{\sqrt{2}}(|0000\psi\rangle + |1100\psi\rangle) = |\Phi^+\rangle \otimes |00\rangle \otimes |\psi\rangle\,. \tag{7.14}$$

Thus, the encoding of $|\psi\rangle$ into $|\psi\rangle_L$ requires a single EPR pair $|\Phi^+\rangle$ along with the two ancillary qubits.

It is now briefly discussed how $|\psi\rangle_L$ is prepared given that the system is initialised in $|\psi_0\rangle = |000\psi 0\rangle$. Firstly, the Hadamard gate is applied to the first qubit, i.e., $H \otimes I^{\otimes 4}$ is applied, which yields

$$|\psi_1\rangle = (H \otimes I^{\otimes 4})\,|\psi_0\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \otimes |00\psi 0\rangle = \frac{1}{\sqrt{2}}(|000\psi 0\rangle + |100\psi 0\rangle).$$

> Secondly, a CNOT gate is applied on the first and last qubit. That is, the fifth qubit is bit flipped if the first qubit is one, otherwise nothing happens. This CNOT gate can be written as $(|0\rangle\langle 0| \otimes I^{\otimes 4} + (|1\rangle\langle 1|) \otimes I^{\otimes 3} \otimes X$. Applying this gate yields
>
> $$|\psi_2\rangle = \text{CNOT}\,|\psi_1\rangle = \frac{1}{\sqrt{2}}(|000\psi 0\rangle + |000\psi 1\rangle),$$
>
> which is equal to the form of $|\psi\rangle_L$ given in (7.13). In other words, applying the Hadamard and CNOT gate to an initialised ground state as described above enables one to prepare the EPR pair $|\Phi^+\rangle$.

The idea used to expand $\mathcal{B}$ to $\mathcal{B}^e$ in Example 7.7 also holds generally. Assuming that $\mathcal{B}$ has c symplectic pairs and s isotropic generators as in (7.7), then c additional qubits is required in $\mathcal{B}^e$ in order to resolve the anti-commutativity in $\mathcal{B}$. The Pauli operators on these c additional qubits for any given generator of $\mathcal{B}^e$ is found by 'copying' the first c Pauli operators of that particular generator in $\mathcal{B}$. By doing so, $\mathcal{B}^e$ becomes an abelian subgroup of $\mathcal{G}_{n+c}$ that is defined to have generators

$$\{Z_1 Z_{n+1}, \dots, Z_c Z_{n+c}, Z_{c+1}, \dots, Z_{c+s}, X_1 X_{n+1}, \dots, X_c X_{n+c}\}, \tag{7.15}$$

where again the first c Z operators are part of a symplectic pair, while the last s are isotropic generators. Since $\mathcal{B}^e$ is a subgroup of $\mathcal{G}_{n+c}$ defined by $2c+s$ generators, it can be used to construct a $[[n+c, n-c-s]]$ stabilizer code. In order to find the stabilizer code in standard form, the logical operators $\{\bar{Z}_1, \dots, \bar{Z}_{n-c-s}, \bar{X}_1, \dots, \bar{X}_{n-c-s}\}$ are needed. They are obtained by putting the check matrix corresponding to the generators of $\mathcal{B}^e$ given in (7.15) in standard form. The fewest possible operations needed in order to do so is if the check matrix is written such that the first c rows corresponds to the X generators in the symplectic pairs, the next s corresponds to the isotropic generators, and the last c rows to the Z operators of the symplectic pairs. Writing the check matrix in this form yields

$$G = \begin{array}{cccc|cccc|l} \overbrace{\quad}^{c} & \overbrace{\quad}^{s} & \overbrace{\quad}^{n-c-s} & \overbrace{\quad}^{c} & \overbrace{\quad}^{c} & \overbrace{\quad}^{s} & \overbrace{\quad}^{n-c-s} & \overbrace{\quad}^{c} & \\ I & 0 & 0 & I & 0 & 0 & 0 & 0 & \}\ c \\ 0 & 0 & 0 & 0 & 0 & I & 0 & 0 & \}\ s. \\ 0 & 0 & 0 & 0 & I & 0 & 0 & I & \}\ c \end{array}$$

By swapping the last two block columns, which corresponds to permuting the last $n-s$ columns c places to the right, G will be in standard form. Explicitly,

$$G = \begin{array}{cccc|cccc|l} \overbrace{\quad}^{c} & \overbrace{\quad}^{s} & \overbrace{\quad}^{c} & \overbrace{\quad}^{n-c-s} & \overbrace{\quad}^{c} & \overbrace{\quad}^{s} & \overbrace{\quad}^{c} & \overbrace{\quad}^{n-c-s} & \\ I & 0 & I & 0 & 0 & 0 & 0 & 0 & \}\ c \\ 0 & 0 & 0 & 0 & 0 & I & 0 & 0 & \}\ s. \\ 0 & 0 & 0 & 0 & I & 0 & I & 0 & \}\ c \end{array}$$

By comparing this with the standard form of the check matrix given in (6.5), it follows that $A_1 = D^\top = \begin{bmatrix} 0 & I \end{bmatrix}$ for $0 \in \text{Hom}(\mathbb{C}^c, \mathbb{C}^s)$ and $I \in \text{End}(\mathbb{C}^c)$, $A_2 = B = C = E = 0$. This implies that each of the $n-c-s$ logical operators in $\{\bar{Z}_i\}_i$ can be found as one of the rows of

$$\left[\begin{array}{ccc|ccc} 0 & 0 & 0 & A_2^\top & 0 & I \end{array}\right] = \left[\begin{array}{ccc|ccc} 0 & 0 & 0 & 0 & 0 & I \end{array}\right], \quad I \in \text{End}(\mathbb{C}^{n-c-s}).$$

This would imply that $\hat{Z}_i = Z_{n+c-(n-c-s)+i} = Z_{2c+s+i}$, however since the last $n-s$ columns has been permuted c places to the right, it actually follows that $\bar{Z}_i = Z_{c+s+i}$. Thus, the set of logical operators is $\{\bar{Z}_1, \dots, \bar{Z}_{n-c-s}\} = \{Z_{c+s+1}, \dots, Z_n\}$. Similarly, the logical operators in $\{\bar{X}_i\}_i$ are the rows of

$$\left[\begin{array}{ccc|ccc} 0 & E^\top & I & C^\top & 0 & 0 \end{array}\right] = \left[\begin{array}{ccc|ccc} 0 & 0 & I & 0 & 0 & 0 \end{array}\right], \quad I \in \text{End}(\mathbb{C}^{n-c-s}),$$

which due to the permutations of columns yields $\bar{X}_i = X_{c+s+i}$. Thus, $\{\bar{X}_1, \dots, \bar{X}_{n-c-s}\} = \{X_{c+s+1}, \dots, X_n\}$.

The logical ground state for the code is then found as the simultaneous $+1$-eigenspace of the generators of $\mathcal{B}^e$ and the logical Z operators, i.e., of $\{Z_1 Z_{n+1}, \dots, Z_c Z_{n+c}, Z_{c+1}, \dots, Z_n, X_1 X_{n+1}, \dots, X_c X_{n+c}\}$, where the first c generators comes from symplectic pairs of $\mathcal{B}$, the next s from the isotropic generators, the next $n-c-s$ from the logical Z operators, and the last c from symplectic pairs again. Since qubit $c+1$ to qubit n only is affected by a Z operator, all of these qubits must be in their ground state. Simultaneously, the first c generators combined with the last c generators imply that qubit i must be equal to qubit $n+i$ for $i=1,\dots,c$. Since there are c pairs of qubits that must satisfy this constraint, there are 2^c possible ways to put zeros or ones on these qubits, which imply that the amplitude of all of these states must be $1/\sqrt{2^c}$. Written explicitly, the ground state of the code generated by $\mathcal{B}^e$ is

$$|0\rangle_L = \frac{1}{\sqrt{2^c}}\left(\sum_{i\in\{0,1\}^c} |i\rangle \otimes |0\rangle^{\otimes(n-c)} \otimes |i\rangle\right).$$

Using that the logical X operators are $\{X_{c+s+1},\dots,X_n\}$, all of the logical basis states can in their decimal representation be written on the form

$$|k\rangle_L = \frac{1}{\sqrt{2^c}}\left(\sum_{i\in\{0,1\}^c} |i\rangle \otimes |0\rangle^{\otimes s} \otimes |k\rangle \otimes |i\rangle\right), \quad k=0,\dots,2^{n-c-s}-1.$$

This representation is convenient since it enables one to find the general formula for the encoded state of a general state consisting of $n-c-s$ qubits. In decimal representation, such a state has the form

$$|\psi\rangle = \sum_{k=0}^{2^{n-c-s}-1} \alpha_k |k\rangle .$$

This implies that the encoded state of $|\psi\rangle$ must have the form

$$\begin{aligned}|\psi\rangle_L &= \sum_{k=0}^{2^{n-c-s}-1} \alpha_k |k\rangle_L = \frac{1}{\sqrt{2^c}} \sum_{k=0}^{2^{n-c-s}-1} \alpha_k \sum_{i\in\{0,1\}^c} |i\rangle \otimes |0\rangle^{\otimes s} \otimes |k\rangle \otimes |i\rangle \\ &= \frac{1}{\sqrt{2^c}} \sum_{i\in\{0,1\}^c} |i\rangle \otimes |0\rangle^{\otimes s} \otimes \left(\sum_{k=0}^{2^{n-c-s}-1} \alpha_k |k\rangle\right) \otimes |i\rangle = \frac{1}{\sqrt{2^c}} \sum_{i\in\{0,1\}^c} |i\rangle \otimes |0\rangle^{\otimes s} \otimes |\psi\rangle \otimes |i\rangle .\end{aligned}$$

The above expression for the encoded state can be simplified by re-indexing the qubits such that the indices come in the order $1, n+1, 2, n+2, \dots, c, n+c, c+1, \dots, n$. By performing such re-indexing, the encoded state has the form

$$|\psi\rangle_L = \frac{1}{\sqrt{2^c}} \sum_{i_1,\dots,i_c\in\{0,1\}} |i_1 i_1\rangle \otimes \dots \otimes |i_c i_c\rangle \otimes |0\rangle^{\otimes s} \otimes |\psi\rangle = |\Phi^+\rangle^{\otimes c} \otimes |0\rangle^{\otimes s} \otimes |\psi\rangle . \tag{7.16}$$

Thus, the $[[n+c, n-c-s]]$ stabilizer code generated by $\mathcal{B}^e$, which in turn was constructed from $\mathcal{B}$ having c symplectic pairs and s isotropic generators, requires c EPR pairs and s ancillary qubits to encode the $n-k-c$ logical qubits into $n+c$ physical qubits.

The assumption of the EA coding framework is that the EPR pairs needed to construct the stabilizer code of $\mathcal{B}^e$ is shared between Alice and Bob. In other words, Alice is in possession of the n qubits corresponding to the generators of $\mathcal{B}$, while Bob has the c qubits needed in order to expand $\mathcal{B}$ to the abelian subgroup $\mathcal{B}^e$. That is, Alice possesses c of the ebits in addition to her $n-k-c$ qubit state to be encoded and s ancillary qubits for a total of n qubits, while Bob only possess the other c ebits. Since Alice only posses n qubits, it is quite misleading to denote the code generated by $\mathcal{B}^e$ as a $[[n+c, n-c-s]]$ code. Instead, it is denoted as a $[[n, n-c-s; c]]$ EA code, or more often a $[[n, k; d]]$ by letting $k=n-c-s$. The semicolon is used to distinguish it from an $[[n, k, d]]$ stabilizer code.

Although it should be clear from the above discussion, the parameters of a $[[n, k; c]]$ EA code are explicitly emphasised. The total number of qubits needed for the code is $n+c$, which is is given by the group $\mathcal{B}^e$ being a subgroup of $\mathcal{G}_{n+c}$. Of these both Alice and Bob possess c ebits since $\mathcal{B}_s$ contains c symplectic pairs. Since the i-th isotropic generator of $\mathcal{B}$ requires the i-th qubit of the encoded state to be a zero, Alice needs s ancillary qubits in order to define the stabilizer code, where s is the number of isotropic generators of $\mathcal{B}_I$. This implies that Alice can encode $k=n-c-s$ qubits of information in the code.

If the distance of the code is known, it is denoted an $[[n, k, d; c]]$ code. It should be noted that an $[[n, k]]$ stabilizer code is commonly denoted as an $[[n, k; 0]]$ EA code to be consistent with notation, which also clearly illustrate that stabilizer codes are a particular case of EA codes where no EPR pairs are needed. The rate of an $[[n, k; c]]$ code is however more ambiguous than for general quantum codes, since there are the following two interpretations:

i) If Alice can generate the c EPR pairs as well as distribute c ebits to Bob freely, then c has no impact on the rate of the code. Hence, the rate is simply k/n. This would be the case if Alice can build up a lot of entanglement that can be stored until quantum states must be communicated. In practice, this is however not possible due to decoherence.

ii) More realistically, the entanglement must be build up during the communication given that Alice and Bob share c Bell pairs beforehand. In each channel use, Alice can with an $[[n, k; c]]$ EA code encode k logical qubits into n physical qubits by utilising c ebits. Since the ebits are measured during the decoding process as discussed in Section 7.1.3, c ebits must be distributed in every channel use in order for the code to be usable multiple times. This implies that the decoder only can encode $k - c$ logical qubits and c ebits into the n physical qubits in every turn. Thus, the entanglement is essentially build at the sacrifice of sending qubits of information, which yield the so-called net rate $(k - c)/n$. Naturally, the above process only works if $k < c$ such that the net rate is positive, however, a non-positive net rate does not imply that information cannot be send between the encoder and decoder, but merely that the number of ebits required for such communication is larger than the amount of logical qubits that can be communicated.

The most appropriate measure of rate depends on the context. The code constructed by $\mathcal{B}^e$ defined in Equation (7.11) is examined in the EA coding framework.

> **Example 7.8: Entanglement-Assisted Code from Expanded Isomorphic Subgroup**
> Since $\mathcal{B} \subset \mathcal{G}_4$ can be decomposed into a pair of symplectic and two isotropic generators, then the parameters for the code generated by $\mathcal{B}^e$ are; $n = 4, c = 1, s = 2$, hence $k = 4 - 1 - 2 = 1$. Thus, $\mathcal{B}^e$ constitute an $[[4, 1; 1]]$ EA code on the form given in (7.14). The rate of this code is thus 1/4, while the net rate is zero.

Having defined the EA code generated by $\mathcal{B}^e$, the next problem is to find a code from $\mathcal{S}$, which actually was the initial purpose of this chapter.

7.1.1 Encoding

Defining a code from $\mathcal{S}$ defined in Example 7.1 is done similarly as how a code was constructed from $\mathcal{B}$ in Example 7.7. More precisely, $\mathcal{S}$ is expanded to $\mathcal{S}^e$ similarly to how $\mathcal{B}$ was expanded to $\mathcal{B}^e$. Then, the stabilizer code defined by $\mathcal{S}^e$ is constructed and put on the form of an EA code. This is firstly considered for $\mathcal{S}$ with the decomposition given in Example 7.3 to illustrate the idea.

> **Example 7.9: Entanglement-Assisted Code From Original Subgroup**
> The generators of $\mathcal{S}$ is by (7.2) given as $\{\hat{Z}_1, \hat{Z}_2, \hat{Z}_3, \hat{X}_1\} = \{Z_1X_2Z_3, Y_1X_2X_3Z_4, X_1Z_2Z_3Y_4, Z_1Z_2Z_4\}$. The anti-commutativity of $\hat{Z}_1$ with $\hat{X}_1$ can be resolved by appending an additional qubit to obtain $\mathcal{S}^e$. The generators of $\mathcal{S}^e$ are then found similarly to how the generators of $\mathcal{B}^e$ were found in Example 7.7, which is to let each symplectic generator be expanded with either Z or X on the additional qubit, while the isotropic generators are expanded with the identity. The generators of $\mathcal{S}^e$ therefore has the form
>
> $$\hat{Z}_1^e = Z_1X_2Z_3Z_5, \quad \hat{X}_1^e = Z_1Z_2Z_4X_5, \quad \hat{Z}_2^e = Y_1X_2X_3Z_4, \quad \hat{Z}_3^e = X_1Z_2Z_3Y_4.$$
>
> Thus, $\mathcal{S}^e$ is an abelian subgroup of $\mathcal{G}_5$. From these generators, one could in principle follow the same procedure as given in Example 7.7 to find the corresponding stabilizer code from $\mathcal{S}^e$. However, since $\mathcal{S}$ and $\mathcal{B}$ are unitarily equivalent by Theorem 7.4, where the specific unitary mapping, U, relating $\mathcal{S}$ and $\mathcal{B}$ is given in (7.10), one can utilise U to find the simultaneous +1-eigenspace of $\{\hat{Z}_1^e, \hat{Z}_2^e, \hat{Z}_3^e, \hat{X}_1^e\}$ from the +1-eigenspace of $\{Z_1^e, Z_2^e, Z_3^e, X_1^e\}$. More precisely, the logical ground state of $\mathcal{S}^e$, $|\hat{0}\rangle_L$, can be obtained by applying $(U^\dagger \otimes I_B)$ to $|0\rangle_L$, where $|0\rangle_L$ is the logical ground state of $\mathcal{B}^e$ given in (7.12). Although it should be clear from the dimensions, it is emphasised that $U^\dagger$ only acts on Alice's four qubits, while the

identity act on Bob's ebit. The fact that $|\hat{0}\rangle_L$ is in the simultaneous +1-eigenspace of the generators of $\mathcal{S}^e$ follows directly from their unitary equivalence with the the generators of $\mathcal{B}^e$ on the first four qubits. Stated explicitly, that is

$$\hat{Z}_i^e\,|\hat{0}\rangle_L = \hat{Z}_i^e\left(U^\dagger \otimes I_B\right)|0\rangle_L \overset{(a)}{=} (U^\dagger \otimes I_B) Z_i^e\,|0\rangle_L \overset{(b)}{=} (U^\dagger \otimes I_B)\,|0\rangle_L \overset{(c)}{=} |\hat{0}\rangle_L\,, \quad i = 1, 2, 3,$$

where (a) follows from the unitary equivalence of $\hat{Z}_i^e$ and Z_i^e on the qubits belonging to Alice as well as $\hat{Z}_i^e$ being identical to Z_i^e on Bob's ebit due to how $\mathcal{B}$ and $\mathcal{S}$ are expanded similarly, (b) from $|0\rangle_L$ being in the +1-eigenspace of Z_i^e by construction $|0\rangle_L$, and (c) from definition of $|\hat{0}\rangle_L$. The result follows analogously for $\hat{X}_1^e$.

By applying $(U^\dagger \otimes I)$ to $|0\rangle_L$, the logical ground state for $\mathcal{S}^e$ is seen to be

$$\begin{aligned} |\hat{0}\rangle_L = \frac{1}{4\sqrt{2}}\big[\,&|00000\rangle + |00001\rangle + |00010\rangle - |00011\rangle + |00100\rangle + |00101\rangle + |00110\rangle - |00111\rangle \\ &+ |01000\rangle - |01001\rangle + |01010\rangle + |01011\rangle - |01100\rangle + |01101\rangle - |01110\rangle - |01111\rangle \\ &\mathrm{i}\big(\,|10000\rangle - |10001\rangle - |10010\rangle - |10011\rangle - |10100\rangle + |10101\rangle + |10110\rangle + |10111\rangle \\ &- |11000\rangle - |11001\rangle + |11010\rangle - |11011\rangle - |11100\rangle - |11101\rangle + |11110\rangle - |11111\rangle\,\big)\big]. \end{aligned}$$

Analogously, $|\hat{1}\rangle_L = (U^\dagger \otimes I)\,|1\rangle_L$, where $|1\rangle_L$ also is given in (7.12). Thus, Alice can encode a qubit, $|\psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle$ into $|\hat{\psi}\rangle_L = \alpha\,|0\rangle_L + \beta\,|1\rangle_L$ by simply applying $U^\dagger$ to her four qubits, that is, her ebit, two ancillary qubits, and her qubit of information.

In the general case, the same idea holds. The non-abelian subgroup $\mathcal{S}$ of $\mathcal{G}_n$ generated by $\{\hat{Z}_1, \ldots, \hat{Z}_{c+s}, \hat{X}_1, \ldots, \hat{X}_c\}$ is expanded to an abelian subgroup $\mathcal{S}^e$ of $\mathcal{G}_{n+c}$ by appending Z_{n+i} (or X_{n+i}) to generator $\hat{Z}_i$ (or $\hat{X}_i$) if it corresponds to a symplectic generators. Then, $\mathcal{S}^e$ is generated by

$$\{\hat{Z}_1 Z_{n+1}, \ldots, \hat{Z}_c Z_{n+c}, \hat{Z}_{c+1}, \ldots, \hat{Z}_{c+s}, \hat{X}_1 X_{n+1}, \ldots, \hat{X}_c X_{n+c}\}.$$

Since $\mathcal{S}^e$ and $\mathcal{B}^e$ are isomorphic on the first n qubits, and identical on the last c, Theorem 7.4 implies that $\mathcal{S}^e$ and $\mathcal{B}^e$ are unitarily equivalent with U on the first n qubits. This particularly implies that the +1-eigenspace of the generators of $\mathcal{S}^e$ can be obtained by applying $(U^\dagger \otimes I)$ to the +1-eigenspace of generators of $\mathcal{B}^e$, where $U^\dagger$ acts on Alice's n qubits and the identity on Bob's c ebits. In turn, this implies that Alice can encode her $n - c - s$ qubits of information into n qubits by applying $U^\dagger$ on her n qubits. Thus, the corresponding EA code is by using (7.16) given as

$$|\hat{\psi}\rangle_L = (U^\dagger \otimes I)(|\Phi^+\rangle^{\otimes c} \otimes |0\rangle^{\otimes s} \otimes |\psi\rangle),$$

where the identity acts on Bob's ebits. It should be noted that the encoding can be efficiently implemented by using the algorithm described in [WKB10, ch. 6].

Having defined the EA code constructed by starting with a non-abelian subgroup $\mathcal{S}$ of $\mathcal{G}_n$, it is time to examine its capability to correct errors.

7.1.2 Existence of Correction Channel

Before discussing the error-correcting capabilities of the code generated by $\mathcal{S}^e$, the simpler code generated by $\mathcal{B}^e$ is analysed. To do so, an initial assumption on the entanglement between Alice and Bob is introduced. More precisely, the c ebits possessed by Bob are assumed to be noiseless, as they are not in general required to be transmitted over the same channel that is used for communication. Essentially, the ebits are assumed to be transmitted over a noiseless channel or encoded appropriately before using a noisy channel as discussed in Section 7.3. This assumption implies that the error corrupting the state $|\psi_L\rangle$ given in (7.16) has the form

$$E = E_A \otimes I_B,$$

where $E_A \in \mathcal{G}_n$ and $I_B = I \in \mathcal{G}_c$. The error E can be corrected if it satisfies the stabilizer Knill-Laflamme conditions given in Theorem 6.8. That is, E is correctable if it either is an element of $\mathcal{B}^e$ or anti-commutes with at least one of the generators of $\mathcal{B}^e$. These two cases are now considered separately.

i) Since symplectic generators of $\mathcal{B}$ are expanded with either an Z or X operator on one of the qubits corresponding to Bob's ebits when $\mathcal{B}$ is expanded to $\mathcal{B}^e$, E cannot be an element of $\mathcal{B}^e$ obtained from a symplectic generator of $\mathcal{B}$ due to the assumption of Bob's ebits being noiseless. Oppositely, isotropic generators of $\mathcal{B}$ are expanded with identity operators on Bob's ebits, thus E is an element of $\mathcal{B}^e$ if E_A is an an element of the isotropic group of $\mathcal{B}$, $\mathcal{B}_I$.

ii) The assumption of Bob's ebits being noiseless implies that the commutativity relations between E and the generators of $\mathcal{B}^e$ are equivalent to that of E_A and $\mathcal{B}$. Thus, E anti-commutes with at least one generator of $\mathcal{B}^e$ if E_A does with at least one generator of $\mathcal{B}$.

Thus, the assumption of Bob's ebits being noiseless implies that the error-correcting capability of the code generated by $\mathcal{B}^e$ is directly related to $\mathcal{B}$. More precisely, i) implies that E is correctable if $E \in \mathcal{B}_I$, while ii) implies that E is correctable if $E \notin N(\mathcal{B})$. By combining these results, a set of errors $\mathcal{E} \subset G_n$ acting on Alice's qubits described by Kraus operators $\{E_i\}_i$, the entanglement-assisted Knill-Laflamme conditions, which are informally described in the points above but not rigorously proven here, are given as

$$E_i^\dagger E_j \notin N(\mathcal{B}) \setminus \mathcal{B}_I, \quad \forall\, i, j.$$

The above arguments translate directly to the code generated by $\mathcal{S}^e$. Thus, the entanglement-assisted Knill-Laflamme conditions for the code generated by $\mathcal{S}^e$ are given by

$$E_i^\dagger E_j \notin N(\mathcal{S}) \setminus \mathcal{S}_I, \quad \forall\, i, j. \tag{7.17}$$

In order to construct an EA code with some desirable error-correcting capabilities, one simply needs to find a subgroup, $\mathcal{S}$, that satisfy the entanglement-assisted Knill-Laflamme conditions for the error set in question. To complete the example of the EA code constructed by $\mathcal{S}$ in Example 7.9, the error-correcting capabilities of the code is examined.

> **Example 7.10: Correctable Set of Errors for Entanglement-Assisted Code**
> From Example 7.3, then $\mathcal{S} = \langle Z_1X_2Z_3, Z_1Z_2Z_4, Y_1X_2X_3Z_4, X_1Z_2Z_3Y_4\rangle$, whereof the isotropic group of $\mathcal{S}$ has the form $\mathcal{S}_I = \langle Y_1X_2X_3Z_4, X_1Z_2Z_3Y_4\rangle$. In order to determine a correctable error set for the EA code constructed from $\mathcal{S}$, the normaliser of $\mathcal{S}$ is needed. By Theorem 6.7, the normaliser of $\mathcal{S}$ is equivalent to the centraliser of $\mathcal{S}$ since $-I \notin \mathcal{S}$ by assumption (the multiplicative factor is fixed such that all generators are Hermitian). Thus, the aim is to find low weight elements in $\mathcal{G}_4$ that commute with all generators of $\mathcal{S}$, i.e., low weight elements in $C(\mathcal{S})$.
>
> Looking at the generators of $\mathcal{S}$, it is easily seen that no element in $\mathcal{G}_4$ with weight one can be in $C(\mathcal{S})$ since no qubit has the same Pauli operator in every generator. For example, looking at the first qubit, the first generator implies that the element must be Z_1 or I to commute, but the third generator implies Y_1 or I, which only yields I as a suitable choice. However, I trivially commutes with all elements, has zero weight, and is an element of $\mathcal{S}_I$, hence is not of interest. Similarly, no element in $\mathcal{G}_4$ of weight two can be in $C(\mathcal{S})$ since no pair of qubits has the same commutation relations among all generators. For example, looking at the first two qubits, the first generator has anti-commuting Pauli operators, while the second has commuting, hence there is no way to commute with both of these generators with only Pauli operators on these two qubits.
>
> However, an element of weight three can commute with all. An example is $Z_1Z_3Z_4$, which compared to the first two generators has commuting Pauli operators on all four qubits, while it for the last two generators has two qubits where they commute and two where they anti-commute. Furthermore, $Z_1Z_3Z_4 \notin \mathcal{S}_I$. In conclusion $Z_1Z_2Z_3 \in N(\mathcal{S}) \setminus \mathcal{S}_I$, which imply that it would not be a correctable error. By the above discussion, it is in fact one of the lowest weight errors that is not correctable. Hence, the distance of the code generated by $\mathcal{S}$ is three. In other words, the code generated by $\mathcal{S}$ is a $[[4, 1, 3; 1]]$ EA code.

The result of Example 7.10 deserves a remark. The EA code generated from $\mathcal{S}$ enables Alice to encode a single qubit of information into four physical qubits such that one arbitrary error can be corrected. Comparing this with the quantum Singleton bound presented in Section 5.2, such a code cannot exist since it requires at least five qubits to achieve such performance. The reason why the EA code can exist anyhow is that it furthermore requires the EPR pair shared between Alice and Bob, which naturally does not come for free. The result nonetheless demonstrates the power of EA codes as Alice can use fewer physical qubits on her own to encode the information than she possibly could with stabilizer codes or any other type of quantum code for that matter.

Having determined which errors that are correctable for EA codes, it is discussed how such errors are detected and corrected.

7.1.3 Decoding

After Alice has encoded her k qubits of information into n physical qubits and transmitted them through a noisy quantum channel, Bob posses the n, potentially, noisy qubits from Alice as well as his c ebits. In order to detect error a correctable error, Bob measures the $2c + s$ generators of $\mathcal{S}^e$ as described in Section 6.2.4. The outcome of measuring these generators is a string of ± 1s of length $2c + s$ that indicate the error syndrome. Since a correctable error will rotate the encode state into a ± 1-eigenspace of the generators of $\mathcal{S}^e$, it can be corrected by simply applying the inverse rotation, i.e., of E happens, $E^\dagger$ can be applied by Bob. After having corrected the error, Bob naturally also has to apply the inverse encoding operation as well as tracing out the ancillary qubits in order to obtain $|\psi\rangle$.

Now that the general EA coding framework have been presented, it is briefly discussed how one can construct such codes.

7.2 Construction of Entanglement-Assisted Codes

The EA code described in the examples in the previous section served as an example of how to generate such codes. In principle, any subgroup of $\mathcal{G}_n$ can be used to define such a code, however, not all will yield the same performance. Perhaps the simplest manner to generate an EA code is to consider a stabilizer code such that the subgroup $\mathcal{S}$ already is abelian. Without further consideration, this will however not take shared EPR pairs into account. It is nonetheless easy to find an EA code that is equivalent to a stabilizer code. This is illustrated by an example.

> **Example 7.11: Entanglement-Assisted Codes from the Steane Code**
> Consider the Steane code described in Section 6.3, which is a $[[7, 1, 3]]$ stabilizer code with generators
>
> $$X_1X_4X_5X_7, \quad X_2X_4X_6X_7, \quad X_3X_5X_6X_7, \quad Z_1Z_4Z_5Z_7, \quad Z_2Z_4Z_6Z_7, \quad Z_3Z_5Z_6Z_7.$$
>
> One can then define a subgroup $\mathcal{S}$ of $\mathcal{G}_6$ by the generators above when removing the first qubit. In that case is it easily verified that only the first and fourth generator will be affected such that they will become an anti-commuting pair, while all other pairs of generators will remain commuting. In other words, there will be one symplectic pair of generators and four isotropic generators of $\mathcal{S}$. But $\mathcal{S}$ will then not generate a stabilizer code. It will, however, generate an EA code where a single ebit is needed due to there being one symplectic pair. In fact, the generators of $\mathcal{S}^e$ will be those of the Steane code (up to a re-labelling of the qubits), which imply that the EA code will have distance three as the Steane code. Thus, $\mathcal{S}$ generate an $[[6, 1, 3; 1]]$ code.
>
> The same arguments hold if one remove the first two or even the first three qubits, which will generate $[[5, 1, 3; 2]]$ and $[[4, 1, 3; 3]]$ EA codes, respectively.

Example 7.11 illustrates that one easily can define EA codes from stabilizer codes by 'transferring' some of the ancillary qubits from Alice to ebits belonging to Bob. The general form is that an $[[n, k, d; 0]]$ EA code can generate an $[[n - c, k, d; c]]$ EA code for some $c \leq \lfloor (n - k)/2 \rfloor$. The upper bound on c follows from there being at most $(n - k)/2$ symplectic pairs in the $n - k$ generators of the stabilizer code.

Although designing EA codes from stabilizer codes is simple, it has the drawback of not taking Bob's ebits being noiseless into account. More precisely, since the error-correcting capabilities of the EA code is similar to that of the stabilizer code, it can correct arbitrary errors on up to $\lfloor (d - 1)/2 \rfloor$ arbitrary qubits including Bob's noiseless ebits. Since the error-correcting capability on Bob's noiseless ebits is redundant, it should be better to construct EA codes with another approach such that it only can correct, hopefully more, errors on Alice's qubits. This can be done by choosing the subgroup $\mathcal{S} \subset \mathcal{G}_n$ such that the minimum weight of an element in $N(\mathcal{S}) \setminus \mathcal{S}_I$ is as high as possible under suitable conditions on $\mathcal{G}_n$ meaning that

n naturally cannot be larger than the qubit count of Alice's quantum computer. This is, however, not further discussed in this thesis as the aim is not to construct EA codes with good performance.

Instead of having to perform the symplectic Gram-Schmidt process to decompose a group, $\mathcal{S}$, into symplectic and isotropic generators to determine how many EPR pairs Alice and Bob need to share in order to use an EA code constructed from $\mathcal{S}$, it would be beneficial to have a simple formula for determining such. This is in fact possible as shown below.

Theorem 7.12: EPR Pairs Needed for an Entanglement-Assisted Code
Let $\mathcal{S} = \langle g_1, \dots, g_{n-k} \rangle$ be a subgroup of $\mathcal{G}_n$, and let the corresponding check matrix be

$$G = \Big[G_X \;\Big|\; G_Z \Big].$$

Then $\mathcal{S}$ generate, after the steps described in Section 7.1, an $[[n, k + c; c]]$ EA code, where $c = \frac{1}{2}\text{rank}\left(G_X G_Z^\top + G_Z G_X^\top\right)$. [WB08, p. 1]

Before giving the proof, it should be noted that it utilises that the symplectic Gram-Schmidt process yields the least possible amount of symplectic pairs for $\mathcal{S}$ [WB08, p. 2]. Intuitively, this is true since the algorithm recursively determines such pairs until all the remaining generators are isotropic. This will, however, not be more rigorously argued.

Proof
The matrix $G_X G_Z^\top + G_Z G_X^\top$ is exactly the symplectic inner product of G with itself, i.e., $G \Lambda G^\top$. Using the symplectic Gram-Schmidt process, only two operations are needed to decompose $\mathcal{S}$ into symplectic and isotropic groups. The first operation is re-labelling the generators, which corresponds to swapping rows in G. The second is to multiply a generator with another, which corresponds to adding a row of G to another. Thus, the symplectic Gram-Schmidt process can be performed by left-multiplying elementary matrices on G. Let A denote the matrix containing row operations needed to decompose $\mathcal{S}$ into the symplectic group $\mathcal{S}_S$ and the isotropic group $\mathcal{S}_I$. Furthermore, let $G' = AG$. Then

$$G' \Lambda (G')^\top = A G \Lambda G^\top A^\top = A \left(G_X G_Z^\top + G_Z G_X^\top\right) A^\top.$$

By (6.3), the (i, j)-th element of $G' \Lambda (G')^\top$ is zero if g_i and g_j commute, and one if they anti-commute. It therefore follows that, by re-labelling generators if needed, $G' \Lambda (G')^\top$ can be written as a diagonal block matrix with the Pauli matrix X in the first c submatrices and zeros elsewhere, where c is the number of symplectic pairs in $\mathcal{S}_S$. In other words, it can be written on the form

$$G' \Lambda (G')^\top = \bigoplus_{i=1}^{c} X \oplus \bigoplus_{j=1}^{n-k-2c} 0,$$

where $\oplus$ denotes the direct sum and $0 \in \text{End}(\mathbb{C}^1)$ is the zero matrix. It then follows that

$$\text{rank}\left(G' \Lambda (G')^\top\right) = \text{rank}\left(\bigoplus_{i=1}^{c} X \oplus \bigoplus_{j=1}^{n-k-2c} 0\right) \overset{(a)}{=} \sum_{i=1}^{c} \text{rank}(X) + \sum_{j=1}^{n-k-2c} 0 = 2c,$$

where (a) follows from the rank of a direct sum being the the sum of the ranks of each matrix. Therefore,

$$c = \frac{1}{2}\text{rank}\left(A\left(G_X G_Z^\top + G_Z G_X^\top\right) A^\top\right) = \frac{1}{2}\text{rank}\left(G_X G_Z^\top + G_Z G_X^\top\right),$$

where the last equality follows from A being a product of elementary operators implying that A has full rank. Since the encoder has n qubits whereof c are ebits and $n - k - 2c$ are ancillary qubits, it follows that $\mathcal{S}$ generate an $[[n, k + c; c]]$ EA code, which completes the proof. ■

Now that it briefly has been discussed how one can construct EA codes, it is worth emphasising that the framework presented hitherto assumes that Bob's ebits are noiseless. This assumption is now discussed based on [LB12].

7.3 Noisy Ebits on Bob's Half

The general EA coding framework assumes that Bob's ebits are noiseless. This is however difficult to guarantee in practice, where the following three sources of noise may corrupt his ebits:

i) The generation of EPR pairs may be noisy itself such that Bob's ebits are different than expected. For example, instead of generating $|\Phi^+\rangle = (|00\rangle + |11\rangle)/\sqrt{2}$ as expected, the process may generate the slightly different state $\cos(\pi/4.1)\,|00\rangle + \sin(\pi/4.1)\,|11\rangle$ or even a more different state such as $|00\rangle/\sqrt{2} + (|01\rangle + |11\rangle)/2$.

ii) Even if the EPR pairs can be generated perfectly, the distribution of Bob's ebits to him can corrupt the state, e.g., by a phase flip. In that case, $|\Phi^+\rangle$ will be transformed into $|\Phi^-\rangle$.

iii) If the EPR pairs can be generated and distributed perfectly, then Bob's ebits still needs to be stored until needed. During that time, decoherence may corrupt the state, e.g., it may spontaneously emit energy and fall to its ground state.

Since these sources of errors are inevitable in practice, EA codes can generally not be used without correcting the errors on Bob's ebits first or in some other way take the errors into account. A few different approaches to do this has been proposed in [LB12].

The most straightforward is to simply define an EA code that also can correct some errors on Bob's qubits. This can simply be done by constructing a stabilizer code and then 'transferring' some of the qubits to Bob as described in Section 7.2. The performance of such codes is, however, not tailored to the errors that may occur in this setup if the error-correcting capability is the same for all qubits. This is due to the fact that if Bob's ebits only are corrupted by storage noise as described in iii), then it would be reasonable to assume that the error rate on Bob's ebits is lower than that of the qubits being transmitted by Alice. With this assumption, generating codes as described above will, generally, not take this lower error rate on Bob's ebits into account.

It is henceforth assumed that Bob's ebits only are only corrupted by storage noise, and that its error rate is lower than that of transmission. In that case, the above approach works, however, it is probably not optimal. Since Bob's ebits only are corrupted by storage noise, he can after receiving his ebits from Alice, encode them himself to prevent such storage noisy to corrupt the ebits. In other words, Bob can use some $[[m, c, d_1; 0]]$ code to encode the information of his c ebits into m physical qubits such that up to $\lfloor (d_1 - 1)/2 \rfloor$ errors can be corrected. At the same time, Alice can use an $[[n, k, d; c]]$ code to encode k qubits of information into n physical ones such that errors on up to $\lfloor (d_2 - 1)/2 \rfloor$ can be corrected given that Alice and Bob share c EPR pairs. The idea is then that when Bob receives the encoded state from Alice, he can first decode his own ebits to obtain c perfect ebits (and thereby c perfect EPR pairs), which he then can use to decode the state received from Alice using the EA coding formalism. An example of this ides is presented in [LB12, p. 6], where Bob is using an $[[10, 4, 3; 0]]$ code and Alice an $[[5, 1, 5; 4]]$ code. This requires 15 physical qubits in total, whereof 10 are possessed in storage by Bob, and five must transmitted through the noisy channel. This coding scheme enables correction of up to two transmission errors as well as up to one storage error, however, it may also correct some error patterns not on the above form. Letting p_T, p_S denote the error rates of transmission and storage, respectively, the probability of Bob decoding correctly with this scheme is at least

$$\left[(1 - p_S)^{10} + 10p_S(1 - p_S)^9\right]\left[(1 - p_T)^5 + 5p_T(1 - p_T)^4 + 10p_T^2(1 - p_T)^3\right], \tag{7.18}$$

where the first sum corresponds to the probability of Bob decoding the $[[10, 4, 3; 0]]$ code correctly, and the second to decoding the $[[5, 1, 5; 4]]$ code correctly. Rather than using this coding scheme, they could use the best stabilizer code capable of encoding one qubit into 15, which is the $[[15, 1, 5; 0]]$ code. It can correct two arbitrary errors, however, requires 15 physical qubits to be transmitted. The probability of correctly decoding is then simply

$$(1 - p_T)^{15} + 15p_T(1 - p_T)^{14} + 105p_T^2(1 - p_T)^{13}. \tag{7.19}$$

The question is then which coding scheme that has the highest probability of decoding correctly. The first coding scheme only requires five qubits to be physically transmitted as opposed to 15, however, at the cost of Bob storing 10 qubits. Since the storage error rate is assumed to be lower than that of transmission, the first coding scheme seems favourable. Furthermore, the first coding scheme can correct some three-qubit error patterns, namely those corrupting two transmission qubits and one storage qubit, while the second scheme only can correct any two-qubit error pattern, which again is favourable to the first scheme. However, these advantages come at the cost of the first coding scheme is unable to correct all two-qubit error patterns as two storage errors cannot be corrected in general. To analyse which scheme that yields the best performance, a contour plot of the when the first coding scheme performs better than the other is utilised for simplicity. That is, (7.19) has been subtracted from (7.18) to yield a function of the error rates, p_T, p_S, for which a contour plot has been plotted in Figure 7.1. The function used in the contour plot indicates when the first coding scheme performs better than the second, hence positive values in the contour plot indicate that the first coding scheme is superior. From the figure, it can therefore be seen that the first coding scheme performs better only when $p_S \leq p_T$. The first coding scheme performs significantly better than the second when p_S is reasonably smaller than p_T.

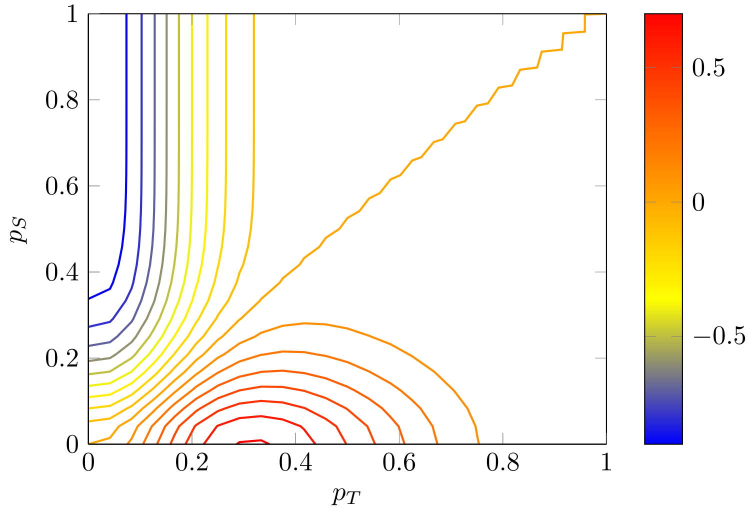


Figure 7.1: Contour plot of (7.18)−(7.19) as a function of the error rates for transmission, p_T, and storage, p_S.

The last approach presented here is entanglement distillation as described in [WKB10, pp. 5-6]. The aim of entanglement distillation is, as the name suggest, to distil some perfect EPR pairs from many noisy ones. Suppose that Alice has generated n noiseless EPR pairs, say $|\Phi^+\rangle^{\otimes n}$. Half of the ebits are then transmitted to Bob through a noiseless channel. Denote, $|\Phi_n^+\rangle = |\Phi_A^+\rangle^{\otimes n} \otimes |\Phi_B^+\rangle^{\otimes n}$ their shared EPR pairs written such that all of Alice's ebits are in the left state and all of Bob's in the right. After some time, Bob's ebits have been corrupted by storage noise. Let this noise be denoted E such that $E \in \mathcal{E} \subset \mathcal{G}_n$. The state of their composite system is then $(I \otimes E) |\Phi_n^+\rangle$, where I and E applies to Alice's and Bob's ebits, respectively. Now, suppose that in order to distil k noiseless EPR pairs from the n noisy ones, Alice and Bob has agreed to use some $[[n, k; 0]]$ stabilizer code for which $\mathcal{E}$ is a set of correctable errors. Let this stabilizer code have generates $\{g_1, \dots, g_{n-k}\}$. Each of these generators can cf. Section 6.2.1 be used as an observable for a projective measurement with two possible outcomes. Let $\{P_i\}_i$ denote the 2^{n-k} projectors corresponding to the $n - k$ generators such that $\{P_i\}_i$ map onto orthogonal subspaces. Now, Alice performs projective measurements on her ebits using the generators of the stabilizer code. Before discussing its effect on the state, it should be noted that the stabilizers of $(I \otimes E) |\Phi_n^+\rangle$ comes in pairs relating Alice's i-th ebit to Bob's i-th ebit. For example, if E is the identity, then $(I \otimes E) |\Phi_n^+\rangle$ is stabilized by $\{X_1X_{n+1}, \dots, X_nX_{2n}, Z_1Z_{n+1}, \dots, Z_nZ_{2n}\}$. If E is not the identity, some generators may flip sign or Pauli operator, but the structure of relating Alice's i-th ebit to Bob's i-th ebit remains. This implies that any generator in $\{g_i\}_i$ will anti-commute with at least one of the stabilizers of $(I \otimes E) |\Phi_n^+\rangle$ since the generators only act on Alice's ebits. By Section 6.2.1, it then follows that measuring generator g_i will randomly project $(I \otimes E) |\Phi_n^+\rangle$ onto either the +1 or −1-eigenspace of g_i. Say it projects according to P_i. The state after measuring g_i is then

$$(P_i \otimes I)(I \otimes E) |\Phi_n^+\rangle . \tag{7.20}$$

This expression can be rewritten by utilising the following matrix property for EPR pairs, which holds for any matrix A:

$$(A \otimes I) |\Phi_n^+\rangle = \left(I \otimes A^\top\right) |\Phi_n^+\rangle . \tag{7.21}$$

This property holds since

$$
\begin{aligned}
(A \otimes I) \left|\Phi_n^+\right\rangle &= \frac{1}{\sqrt{2^n}} \sum_{i \in \{0,1\}^n} (A \left|i\right\rangle) \otimes \left|i\right\rangle \overset{(a)}{=} \frac{1}{\sqrt{2^n}} \sum_{i=0}^{2^n-1} \left(\sum_{j=0}^{2^n-1} a_{j+1,i+1} \left|j\right\rangle \right) \otimes \left|i\right\rangle \\
&= \frac{1}{\sqrt{2^n}} \sum_{j=0}^{2^n-1} \left|j\right\rangle \otimes \left(\sum_{i=0}^{2^n-1} a_{j+1,i+1} \left|i\right\rangle \right) \overset{(b)}{=} \frac{1}{\sqrt{2^n}} \sum_{i \in \{0,1\}^n} \left|i\right\rangle \otimes (A^\top \left|i\right\rangle) = \left(I \otimes A^\top\right) \left|\Phi_n^+\right\rangle,
\end{aligned}
$$

where (a) follows from changing to the decimal representation of the binary numbers, and (b) from going back the the binary representation again. By continuing on (7.20), it then follows that

$$
\begin{aligned}
(P_i \otimes I)(I \otimes E) \left|\Phi_n^+\right\rangle = (P_i \otimes E) \left|\Phi_n^+\right\rangle = (I \otimes E)(P_i \otimes I) \left|\Phi_n^+\right\rangle &\overset{(a)}{=} (I \otimes E)(P_i^2 \otimes I) \left|\Phi_n^+\right\rangle \\
= (I \otimes E)(P_i \otimes I)(P_i \otimes I) \left|\Phi_n^+\right\rangle &\overset{(b)}{=} (I \otimes E)(P_i \otimes I)(I \otimes P_i^\top) \left|\Phi_n^+\right\rangle \\
= (I \otimes E)(P_i \otimes P_i^\top) \left|\Phi_n^+\right\rangle,
\end{aligned}
$$

where (a) follows from P_i being idempotent due to being a projection, and (b) from (7.21). Thus, Alice performing a measurement which projects her ebits onto the subspace spanned by P_i also maps Bob's ebits onto a related subspace, namely the one spanned by $P_i^\top$. The above discussion holds for all generators of the stabilizer code. After Alice has measured all of the $n-k$ generators, she applies appropriate operators from $\mathcal{G}_n$ such that her ebits are stabilized by all of the $n-k$ generators as described in Section 6.2.2. Alice then communicates her measurement outcomes, i.e., an $(n-k)$-dimensional string of ±1, to Bob. He then performs the same projective measurements as Alice, that is, he uses the $n-k$ generators as observables, on his ebits. He can then from Alice's measurement outcomes combined with his own determine an error syndrome. Based on this syndrome, he applies an appropriate recovery operation to his ebits. Thereafter, he also applies appropriate operators to his ebits such that they also are stabilized by all $n-k$ generators of the stabilizer code. This procedure result in Alice and Bob each posses k logical ebits, which they each can decode with an unitary operator on their own ebits to yield k noiseless EPR pairs.

All of the above approaches requires Alice and Bob to allocate some of their qubits to correcting Bob's ebits, which naturally will decrease number of qubits that can be used for other purposes, e.g., quantum computing. It would therefore be beneficial if such attention to Bob's ebits being noiseless could be omitted. However, the fundamental assumption of Bob's ebits being noiseless is the key to the performance of EA codes, hence in that sense it is an absolute necessity to use one of the above approaches. To see whether this exactly is of such importance, the assumption of Bob's ebits being noiseless is removed in order to see how it changes the performance of EA codes.

8 | Decoding Performances for Entanglement-Assisted Codes

In order to examine the importance of Bob's ebits being noiseless in the EA coding framework, it is analysed how noisy ebits affect the decoding performance of such codes. Thus, instead of trying to prevent Bob's ebits being corrupted by noise with one of the approaches presented in Section 7.3, EA codes are simply decoded with Bob's potentially noisy ebits to see to what extent quantum information can be extracted anyhow. To the best of the author's knowledge, no such analysis exists in the literature.

8.1 Setup

Consider Figure 1.2 once again. For this analysis, it is assumed that Alice can generate EPR pairs as well as distribute ebits to Bob perfectly. However, before Bob receives the encoded state from Alice, his ebits may be corrupted from storage noise. It is however known to Bob that such storage errors may occur, hence he can take this into account when decoding. Based on this, different error patterns on Bob's ebits as well as the encoded state are then considered.

For simplicity, both the channel Alice uses for transmission of the encoded state and the one Bob uses for storing his ebits are assumed to be the depolarising channel such that only Pauli errors may occur. The channels, however, are assumed to have different crossover probabilities, namely p_T and p_S for transmission and storage, respectively. It is furthermore assumed that the crossover probabilities are less than 0.5 such that errors are less likely to occur compared to occurring and that $p_S < p_T$ such that Bob's ebits probabilistically are less noisy that the transmitted qubits.

In order to compare different amount of ebits being corrupted without also changing the error-correcting capabilities of the codes, the EA codes obtained from the Steane code as described in Example 7.11 are utilised.

Rather than measuring the performance of the coding scheme for a particular error pattern using the channel fidelity, it is measured by how likely it is to decode correctly.

Before analysing the decoding performance for the $[[6, 1, 3; 1]]$ EA code obtained from the Steane code, the code is explicitly derived in the following section such there is no ambiguity about how the code is defined.

8.2 Derivation of Entanglement-Assisted Steane Code

Following Example 7.11, it is possible to obtain a $[[6, 1, 3; 1]]$ EA code from the Steane code by letting $\mathcal{S}$ be the generators of the Steane code without the first qubit. That is,

$$\mathcal{S} = \langle X_3X_4X_6, X_1X_3X_5X_6, X_2X_4X_5X_6, Z_3Z_4Z_6, Z_1Z_3Z_5Z_6, Z_2Z_4Z_5Z_6\rangle$$

after re-labelling the qubits such that they range from one to six rather than two to seven. Since only the first and forth generators anti-commute, $\mathcal{S}$ contains a single symplectic pair and four isotropic generators, which henceforth are denoted

$$\{\hat{Z}_1, \hat{X}_1, \hat{Z}_2, \hat{Z}_3, \hat{Z}_4, \hat{Z}_5\} = \{Z_3Z_4Z_6, X_3X_4X_6, Z_1Z_3Z_5Z_6, X_1X_3X_5X_6, Z_2Z_4Z_5Z_6, X_2X_4X_5X_6\}.$$

Since $\mathcal{S}$ contains one symplectic pair and four isotropic generators, it follows from (7.16) that the encoded state for the simpler extended subgroup $\mathcal{B}^e$ has the form

$$|\psi\rangle_L = |\Phi^+\rangle \otimes |0\rangle^{\otimes 4} \otimes |\psi\rangle = \frac{1}{\sqrt{2}} \left[(\alpha\,|000000\rangle + \beta\,|000001\rangle)\,|0\rangle_B + (\alpha\,|100000\rangle + \beta\,|100001\rangle)\,|1\rangle_B \right],$$

where the subscript is used to denote the qubit possessed by Bob. In order to determine the unitary operator $U \in \mathrm{End}((\mathbb{C}^2)^{\otimes 6})$ used for transforming the above encoded state for $\mathcal{B}^e$ into an encoded state for

$\mathcal{S}^e$, the generators $\{\hat{Z}_1, \ldots, \hat{Z}_5, \hat{X}_1\}$ must be expanded to a set generating $\mathcal{G}_6$. This is done by expanding the set of generators for $\mathcal{S}$ to six symplectic pairs. It is easily verified that the following set of generators yield six symplectic pairs

$$\begin{aligned}\{\hat{Z}_1, \ldots, \hat{Z}_6\} &= \{Z_3Z_4Z_6, Z_1Z_3Z_5Z_6, X_1X_3X_5X_6, Z_2Z_4Z_5Z_6, X_2X_4X_5X_6, X_3X_4X_5\},\\ \{\hat{X}_1, \ldots, \hat{X}_6\} &= \{X_3X_4X_6, X_4X_6, Z_1, X_3X_6, Z_2, Z_1Z_2Z_5\}.\end{aligned} \tag{8.1}$$

The simultaneous +1-eigenspace of generators in $\{\hat{Z}_i\}_i$ then defines the $|\widehat{000000}\rangle$ state. This can easily be found since $\hat{Z}_1$ implies that there must be either zero or two ones at qubit three, four, and six, which for $x_1, x_2, x_5 \in \{0, 1\}$ only is satisfied for the 32 basis states on the form:

$$|x_1x_200x_50\rangle, |x_1x_201x_51\rangle, |x_1x_210x_51\rangle, |x_1x_211x_50\rangle.$$

Similarly, it follows from $\hat{Z}_2$, that there must be zero, two, or four ones at qubit one, three, five, and six. Since this must be satisfied simultaneously with being one of the 32 states given above, only the 16 following basis states are in the simultaneous +1-eigenspace of $\hat{Z}_1$ and $\hat{Z}_2$:

$$|0x_20000\rangle, |1x_20010\rangle, |0x_20111\rangle, |1x_20101\rangle, |0x_21001\rangle, |1x_21011\rangle, |0x_21110\rangle, |1x_21100\rangle.$$

Analogously, $\hat{Z}_4$ imply that there must be zero, two, or four ones at qubit two, four, five, and six, which implies that only the following 8 basis states is in the simultaneous +1-eigenspace of $\hat{Z}_1, \hat{Z}_2, \hat{Z}_4$:

$$|000000\rangle, |110010\rangle, |010111\rangle, |100101\rangle, |011001\rangle, |101011\rangle, |001110\rangle, |111100\rangle.$$

The remaining generators in $\{\hat{Z}_i\}_i$ only has X operators on them, which essentially implies that the amplitudes of the above basis states must be identical. Thus, the state $|\widehat{000000}\rangle$ is defined as

$$|\widehat{000000}\rangle = \frac{1}{2\sqrt{2}} [|000000\rangle + |001110\rangle + |010111\rangle + |011001\rangle + |100101\rangle + |101011\rangle + |110010\rangle + |111100\rangle].$$

The other basis states can then be found by applying the appropriate generators from $\{\hat{X}_i\}_i$, e.g., $|\widehat{100000}\rangle = \hat{X}_1 |\widehat{000000}\rangle = X_3X_4X_6 |\widehat{000000}\rangle$. Writing it explicitly in terms of the canonical basis states yields

$$|\widehat{100000}\rangle = \frac{1}{2\sqrt{2}} [|001101\rangle + |000011\rangle + |011010\rangle + |010100\rangle + |101000\rangle + |100110\rangle + |111111\rangle + |110001\rangle].$$

From all of such states, U can be determined as $U = \sum_{i\in\{0,1\}^6} |i\rangle \langle\hat{i}|$. Now that U has been defined, $|\psi\rangle_L$ can be encoded into an encoded state of $\mathcal{S}^e$ by applying $(U^\dagger \otimes I_B)$ to $|\psi\rangle_L$. This yields

$$\begin{aligned}|\hat{\psi}\rangle_L &= (U^\dagger \otimes I_B) |\psi\rangle_L\\ &= \frac{1}{\sqrt{2}}\big[(\alpha U^\dagger |000000\rangle + \beta U^\dagger |000001\rangle) |0\rangle_B + (\alpha U^\dagger |100000\rangle + \beta U^\dagger |100001\rangle) |1\rangle_B \big]\\ &= \frac{1}{\sqrt{2}} \sum_{i\in\{0,1\}^6} \big[(\alpha |\hat{i}\rangle \langle i|000000\rangle + \beta |\hat{i}\rangle \langle i|000001\rangle) |0\rangle_B + (\alpha |\hat{i}\rangle \langle i|100000\rangle + \beta |\hat{i}\rangle \langle i|100001\rangle) |1\rangle_B \big]\\ &\overset{(a)}{=} \frac{1}{\sqrt{2}}\big[(\alpha |\widehat{000000}\rangle + \beta |\widehat{000001}\rangle) |0\rangle_B + (\alpha |\widehat{100000}\rangle + \beta |\widehat{100001}\rangle) |1\rangle_B \big]\end{aligned}$$

where (a) follows from $\{|i\rangle\}_i$ being the canonical basis which is orthonormal. By writing out the above states in the canonical basis, $|\hat{\psi}\rangle_L$ has the form

$$|\hat{\psi}\rangle_L = \frac{1}{\sqrt{2}}\big[(\alpha |\widehat{000000}\rangle + \beta\hat{X}_6 |\widehat{000000}\rangle) |0\rangle_B + (\alpha\hat{X}_1 |\widehat{000000}\rangle + \beta\hat{X}_1\hat{X}_6 |\widehat{000000}\rangle) |1\rangle_B\big] \tag{8.2}$$

$$\begin{aligned}\overset{(a)}{=} \frac{1}{4}\big[&\alpha(|000000\rangle + |001110\rangle + |010111\rangle + |011001\rangle + |100101\rangle + |101011\rangle + |110010\rangle + |111100\rangle) |0\rangle_B\\ &+\beta(|000000\rangle - |001110\rangle + |010111\rangle - |011001\rangle - |100101\rangle + |101011\rangle - |110010\rangle + |111100\rangle) |0\rangle_B\\ &+\alpha(|001101\rangle + |000011\rangle + |011010\rangle + |010100\rangle + |101000\rangle + |100110\rangle + |111111\rangle + |110001\rangle) |1\rangle_B\\ &+\beta(|001101\rangle - |000011\rangle + |011010\rangle - |010100\rangle - |101000\rangle + |100110\rangle - |111111\rangle + |110001\rangle) |1\rangle_B \big],\end{aligned}$$

where (a) follows from $\hat{X}_1 = X_3X_4X_6$ and $\hat{X}_6 = Z_1Z_2Z_5$. By construction, this state is then stabilized by the generators of $\mathcal{S}^e$, which is given as

$$\begin{aligned}\mathcal{S}^e &= \langle \hat{Z}_1^e, \hat{X}_1^e, \hat{Z}_2^e, \hat{Z}_3^e, \hat{Z}_4^e, \hat{Z}_5^e \rangle \\ &= \langle Z_3Z_4Z_6Z_7, X_3X_4X_6X_7, Z_1Z_3Z_5Z_6, X_1X_3X_5X_6, Z_2Z_4Z_5Z_6, X_2X_4X_5X_6 \rangle ,\end{aligned} \tag{8.3}$$

where the seventh qubit is the one possessed by Bob. The decoding performance for the code is now analysed.

8.3 Decoding Error Patterns with the $[[6, 1, 3; 1]]$ Code

Assume that no error happens during the physical transmission from Alice to Bob, but Bob's ebit is bit flipped. The error pattern is thus $E = I_A \otimes X = X_7$. Applying this to the logical state, which now is completely possessed by Bob, yields the corrupted logical state

$$\begin{aligned}E\,|\hat{\psi}\rangle_L &= (I_A \otimes X)(U^\dagger \otimes I_B)\,|\psi\rangle_L \\ &= \frac{1}{\sqrt{2}}\big[(\alpha\,|\widehat{000000}\rangle + \beta\,|\widehat{000001}\rangle)\,|1\rangle_B + (\alpha\,|\widehat{100000}\rangle + \beta\,|\widehat{100001}\rangle)\,|0\rangle_B\,\big].\end{aligned} \tag{8.4}$$

Before discussing how Bob should decode this corrupted state, consider the case where he simply decodes without error detection and correction. This is achieved by applying $(U \otimes I_B)$ to the state, yielding

$$\begin{aligned}(U \otimes I_B)E\,|\hat{\psi}\rangle_L &= (U \otimes I_B)(I_A \otimes X)(U^\dagger \otimes I_B)\,|\psi\rangle_L = (I_A \otimes X)\,|\psi\rangle_L \\ &= \frac{1}{\sqrt{2}}\big[(\alpha\,|000000\rangle + \beta\,|000001\rangle)\,|1\rangle_B + (\alpha\,|100000\rangle + \beta\,|100001\rangle)\,|0\rangle_B\,\big] \\ &= \frac{1}{\sqrt{2}}\big[\,|00000\psi\rangle\,|1\rangle_B + |10000\psi\rangle\,|0\rangle_B\,\big] \overset{(a)}{=} |\Psi^+\rangle \otimes |0\rangle^{\otimes 4} \otimes |\psi\rangle\,,\end{aligned}$$

where (a) follows by re-ordering the qubits such that Bob's ebit is the second qubit. From this 'corrected' logical state, Bob can then apply the appropriate CNOT and Hadamard operator, i.e., in the reverse order than described in the end of Example 7.7, such that the state becomes

$$\begin{aligned}(H \otimes I^{\otimes 6})(\text{CNOT})\frac{1}{\sqrt{2}}\big[\,|00000\psi\rangle\,|1\rangle_B + |10000\psi\rangle\,|0\rangle_B\,\big] &= (H \otimes I^{\otimes 6})\frac{1}{\sqrt{2}}\big[\,|00000\psi\rangle\,|1\rangle_B + |10000\psi\rangle\,|1\rangle_B\,\big] \\ &= (H \otimes I^{\otimes 6})\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)\,|0000\psi\rangle\,|1\rangle_B \\ &= |00000\psi\rangle\,|1\rangle_B\,.\end{aligned}$$

By tracing out all the ancillary qubits, that is all but the sixth qubit, Bob obtain $|\psi\rangle$. The calculations follows analogously for other Pauli errors on Bob's ebit, where the EPR pair changed depending on the given error. Hence, despite Bob's ebit being corrupted by a Pauli error, he still obtains the correct state by decoding without performing any error correction. This is quite remarkable, and resembles systematic encoding in classical coding theory, where errors only occurring in the redundancy bits do not corrupt the information. The issue is, however, that errors on the information bits can be confused with errors on the redundancy bits, which then imply that the decoding yield the wrong information. It is now examined whether something similar holds true for the $[[6, 1, 3; 1]]$ EA code.

8.3.1 Error Detection

The first step for Bob in order to decode $E\,|\hat{\psi}\rangle_L$ is to detect the error, which is done by syndrome measurements corresponding to measuring the generators of $\mathcal{S}^e$ given in (8.3). Since $E = X_7$ commutes with all generators of $\mathcal{S}^e$ except the first, the error syndrome obtained by the syndrome measurements is $(-1, +1, +1, +1, +1, +1)$. Given this error syndrome, the task is to determine the most likely error that yields this syndrome. Before determining the most likely error, all errors of low weight yielding this error syndrome are found, which is equivalent to finding elements in $\mathcal{G}_7$ that commute with all but the first generator of $\mathcal{S}^e$. That is to determine E' such that $G\Lambda r(E')^\top = e_1$, where G is the check matrix corresponding to $\mathcal{S}^e$. Explicitly,

$$G\Lambda = \left[\begin{array}{ccccccc|ccccccc} 0 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 0 \end{array}\right].$$

If all qubits had been subject to the same channel, finding the most likely error with the given error syndrome would simply correspond to determining the lowest weight error, E', that yields the syndrome. In this case, it is easily seen to be $E' = X_7$ since only the seventh column of $G\Lambda$ is e_1. However, since Bob's ebits is stored rather than transmitted, the seventh qubit is prone to error with probability p_S whereas the other six are with probability p_T. As it is assumed that $p_S < p_T$, it may be more likely that two errors corrupt the transmitted qubits rather than the stored qubit being corrupted. Since the first and third columns of $G\Lambda$ sum to e_1, $E' = X_1X_3$ is an error with weight two that yields the error syndrome. Analogously, $E' = X_2X_4$ and $E' = X_5X_6$ are plausible two-weight errors with the given error syndrome. In principle, one could also determine errors with weight three, four, and so on, however these will be unlikely to occur compared to those with weight one or two, hence errors with such high weight will not be needed to decode $E\,|\hat{\psi}\rangle_L$. In conclusion, the possible low-weight errors yielding the error syndrome is $E' \in \{X_7, X_1X_3, X_2X_4, X_5X_6\}$. Naturally, Bob cannot distinguish between the last three errors since they yield the same syndrome and are equally likely, hence to uniquely determine an error, the probability of X_7 occurring must be larger than that of X_1X_3, X_2X_4, and X_5X_6. In order to determine when this is the case, the respective probabilities for an error on Bob's ebit and two on the transmitted qubits are determined:

$$\begin{aligned} p_1 &= \Pr(\text{One error on Bob's ebit}) = (1-p_T)^6 p_S \\ p_2 &= \Pr(\text{Two errors on transmitted qubits}) = p_T^2(1-p_T)^4(1-p_S). \end{aligned}$$

The analysis is then concluded by solving $p_1 > p_2$:

$$\begin{aligned} p_1 > p_2 &\implies (1-p_T)^6 p_S > p_T^2(1-p_T)^4(1-p_S) \implies (1-p_T)^2 p_S > p_T^2(1-p_S) \\ &\implies p_S((1-p_T)^2 + p_T^2) > p_T^2 \implies p_S > \frac{p_T^2}{2p_T^2 - 2p_T + 1}. \end{aligned} \tag{8.5}$$

The curve for $p_T^2/(2p_T^2 - 2p_T + 1)$ is plotted in Figure 8.1, wherefrom it follows that the error is more likely to be X_7 when p_S is above the curve, but more likely to be X_1X_3, X_2X_4, or X_5X_6, respectively, when p_S is below the curve. In the case where p_S is on the curve, all four errors are equally likely.

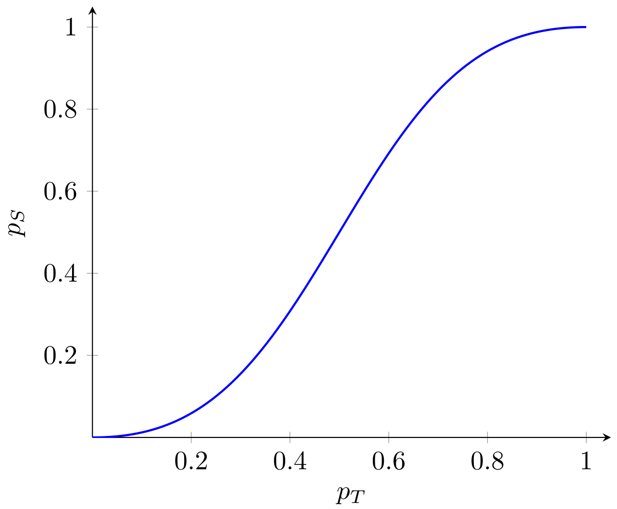


Figure 8.1: Curve of $p_T^2/(2p_T^2 - 2p_T + 1)$, which indicate when the error is most likely to be on Bob's ebit or two of the transmitted qubits.

Thus, if Bob knows the crossover probabilities, p_T, p_S, he can determine whether $E' = X_7$ or $E' \in \{X_1X_3, X_2X_4, X_5X_6\}$ is most likely.

8.3.2 Error Correction and Decoding

If $p_1 > p_2$ such that $E' = X_7$ is most likely, he corrects the error of $E\,|\hat{\psi}\rangle_L$ by applying $E' = (I_A \otimes X)$, whereafter he can decode by applying $(U \otimes I_B)$. By doing so, the state becomes

$$(U \otimes I_B)E'E\,|\hat{\psi}\rangle_L = (U \otimes I_B)(I_A \otimes X)(I_A \otimes X)(U^\dagger \otimes I)\,|\psi\rangle_L = |\psi\rangle_L\,.$$

Hence, Bob obtains $|\psi\rangle_L$ wherefrom he can obtain $|\psi\rangle$ by performing a full decoding. As expected, given that Bob detects the correct error, he can correct it and recover $|\psi\rangle$ even though the error is on his ebit. The fact that he can detect an error on his ebit follows from the $[[6, 1, 3; 1]]$ EA code being equivalent to the Steane code that have distance three such that any error with weight one can be corrected.

If $p_1 < p_2$, Bob cannot uniquely determine the most likely error as $E' \in \{X_1X_3, X_2X_4, X_5X_6\}$ at random. The effect of not being able to uniquely determine the error can be seen by considering the Hermitian operator, $H \in \mathrm{End}(\mathbb{C}^3)$, obtained from the Knill-Laflamme conditions with error set $\mathcal{E} = \{X_1X_3, X_2X_4, X_5X_6\}$, henceforth denoted $\{E_1, E_2, E_3\}$. Since an arbitrary state of the code, $|\hat{\psi}\rangle_L$, is described by an orthonormal basis, it follows from the discussion in Section 5.1 that $h_{ij} = \langle\hat{\psi}|_L\, E_i^\dagger E_j\, |\hat{\psi}\rangle_L$. Hence, the diagonal entries are trivially ones. Since H is Hermitian, it is sufficient to calculate h_{12}, h_{13}, and h_{23} in order to determine H. This can be done by simply inserting the expressions for the errors into the above formula, which yields

$$\begin{aligned}
h_{12} &= \langle\hat{\psi}|_L\, E_1^\dagger E_2\, |\hat{\psi}\rangle_L = \langle\hat{\psi}|_L\, (X_1X_3)(X_2X_4)\, |\hat{\psi}\rangle_L \overset{(a)}{=} \langle\hat{\psi}|_L\, \hat{Z}_3\hat{Z}_5\, |\hat{\psi}\rangle_L \overset{(b)}{=} \langle\hat{\psi}|_L\, |\hat{\psi}\rangle_L = 1,\\
h_{13} &= \langle\hat{\psi}|_L\, E_1^\dagger E_3\, |\hat{\psi}\rangle_L = \langle\hat{\psi}|_L\, (X_1X_3)(X_5X_6)\, |\hat{\psi}\rangle_L \overset{(a)}{=} \langle\hat{\psi}|_L\, \hat{Z}_3\, |\hat{\psi}\rangle_L \overset{(b)}{=} \langle\hat{\psi}|_L\, |\hat{\psi}\rangle_L = 1,\\
h_{23} &= \langle\hat{\psi}|_L\, E_2^\dagger E_3\, |\hat{\psi}\rangle_L = \langle\hat{\psi}|_L\, (X_2X_4)(X_5X_6)\, |\hat{\psi}\rangle_L \overset{(a)}{=} \langle\hat{\psi}|_L\, \hat{Z}_3\hat{Z}_5\, |\hat{\psi}\rangle_L \overset{(b)}{=} \langle\hat{\psi}|_L\, |\hat{\psi}\rangle_L = 1,
\end{aligned}$$

where (a) follows from $\hat{Z}_3 = X_1X_3X_5X_6$ and $\hat{Z}_5 = X_2X_4X_5X_6$ cf. (8.1), while (b) follows from $|\hat{\psi}\rangle_L$ being stabilized by $\hat{Z}_3, \hat{Z}_5$ by construction. Thus, H contains ones in all its entries. Simple calculations then show that the eigenvalues of H are $\lambda_1 = \lambda_2 = 0$ and $\lambda_3 = 3$, which implies that H can be diagonalised such that the diagonal entries are three, zero, and zero, respectively. This implies that the errors in $\mathcal{E}$ all map $|\hat{\psi}\rangle_L$ to the same state, i.e., the code is degenerate with respect to the errors $\mathcal{E}$, hence it is irrelevant which of the errors in $\mathcal{E}$ that Bob uses for correction. Hence, it is henceforth assumed without loss of generality that Bob uses $E' = X_1X_3$ to correct the error. In that case, the state after error correction is

$$E'E\,|\hat{\psi}\rangle_L = X_1X_3X_7\,|\hat{\psi}_L\rangle\,.$$

For simplicity, the analysis is firstly conducted by written out the 'corrected' state in the canonical basis, i.e., by flipping the first, third, and seventh qubit of the last form given in (8.2). Doing so yields the state

$$\begin{aligned}
&\frac{1}{4}\big[\alpha(|101000\rangle + |100110\rangle + |111111\rangle + |110001\rangle + |001101\rangle + |000011\rangle + |011010\rangle + |010100\rangle)\,|1\rangle_B\\
&\quad+\beta(|101000\rangle - |100110\rangle + |111111\rangle - |110001\rangle - |001101\rangle + |000011\rangle - |011010\rangle + |010100\rangle)\,|1\rangle_B\\
&\quad+\alpha(|100101\rangle + |101011\rangle + |110010\rangle + |111100\rangle + |000000\rangle + |001110\rangle + |010111\rangle + |011001\rangle)\,|0\rangle_B\\
&\quad+\beta(|100101\rangle - |101011\rangle + |110010\rangle - |111100\rangle - |000000\rangle + |001110\rangle - |010111\rangle + |011001\rangle)\,|0\rangle_B\big].
\end{aligned}$$

By comparing this state with $|\hat{\psi}\rangle_L$ given in (8.2), it can be seen that the only difference, up to a rearrangement of the sums, is that the sign of the amplitudes for every term associated with β has been flipped. By writing this in terms of the basis states $\{|\hat{i}\rangle\}_i$, that is equivalent to stating that

$$E'E\,|\hat{\psi}\rangle_L = \frac{1}{\sqrt{2}}\big[(\alpha\,|\widehat{000000}\rangle - \beta\,|\widehat{000001}\rangle)\,|0\rangle_B + (\alpha\,|\widehat{100000}\rangle + \beta\,|\widehat{100001}\rangle)\,|1\rangle_B\big]. \tag{8.6}$$

By applying the decoding operator $(U \otimes I_B)$, Bob then obtains the state

$$\begin{aligned}
(U \otimes I_B)E'E\,|\hat{\psi}\rangle_L &= \frac{1}{\sqrt{2}}\big[(\alpha U\,|\widehat{000000}\rangle - \beta U\,|\widehat{000001}\rangle)\,|0\rangle_B + (\alpha U\,|\widehat{100000}\rangle - \beta U\,|\widehat{100001}\rangle)\,|1\rangle_B\,\big] \\
&= \frac{1}{\sqrt{2}}\sum_{i\in\{0,1\}^6}\big[(\alpha\,|i\rangle\,\langle\hat{i}|\widehat{000000}\rangle - \beta\,|i\rangle\,\langle\hat{i}|\widehat{000001}\rangle)\,|0\rangle_B + (\alpha\,|i\rangle\,\langle\hat{i}|\widehat{100000}\rangle - \beta\,|i\rangle\,\langle\hat{i}|\widehat{100001}\rangle)\,|1\rangle_B\,\big] \\
&\overset{(a)}{=} \frac{1}{\sqrt{2}}\big[(\alpha\,|000000\rangle - \beta\,|000001\rangle)\,|0\rangle_B + (\alpha\,|100000\rangle - \beta\,|100001\rangle)\,|1\rangle_B\,\big] \\
&= \frac{1}{\sqrt{2}}\big[\,|00000\rangle \otimes (Z\,|\psi\rangle) \otimes |0\rangle_B + |10000\rangle \otimes (Z\,|\psi\rangle) \otimes |1\rangle_B\,\big] \overset{(b)}{=} |\Phi^+\rangle \otimes |0\rangle^{\otimes 4} \otimes (Z\,|\psi\rangle),
\end{aligned}$$

where (a) follows from $\{|\hat{i}\rangle\}_i$ constituting an orthonormal basis and (b) from permuting all but the first qubit once to the right. From this state, Bob can then perform a complete decoding and obtain $Z\,|\psi\rangle$. Thus, correcting the error with X_1X_3 although the actual error is X_7 implicates that the obtained state has been phase flipped. This analysis have provided great insight into the decoding performance for this particular error pattern, however, it is quite tedious since the form for $|\hat{\psi}\rangle_L$ used is long since it is expressed in the canonical basis. Thus, the same analysis is now performed in the $\{|\hat{i}\rangle\}_i$ basis such that $|\hat{\psi}\rangle_L$ has the first form given in (8.2), i.e.,

$$|\hat{\psi}\rangle_L = \frac{1}{\sqrt{2}}\big[(\alpha\,|\widehat{000000}\rangle + \beta\hat{X}_6\,|\widehat{000000}\rangle)\,|0\rangle_B + (\alpha\hat{X}_1\,|\widehat{000000}\rangle + \beta\hat{X}_1\hat{X}_6\,|\widehat{000000}\rangle)\,|1\rangle_B\big]. \tag{8.7}$$

After Bob's ebit has been corrupted by the bit flip, the state becomes $E\,|\hat{\psi}\rangle_L$, which as already stated in (8.4) is given as

$$E\,|\hat{\psi}\rangle_L = \frac{1}{\sqrt{2}}\big[(\alpha\,|\widehat{000000}\rangle + \beta\hat{X}_6\,|\widehat{000000}\rangle)\,|1\rangle_B + (\alpha\hat{X}_1\,|\widehat{000000}\rangle + \beta\hat{X}_1\hat{X}_6\,|\widehat{000000}\rangle)\,|0\rangle_B\big].$$

Now, Bob 'corrects' this state by applying $E' = X_1X_3$, which yields

$$\begin{aligned}
E'E\,|\hat{\psi}\rangle_L = \frac{1}{\sqrt{2}}\big[&(\alpha E'\,|\widehat{000000}\rangle + \beta E'\hat{X}_6\,|\widehat{000000}\rangle)\,|1\rangle_B \\
&+(\alpha E'\hat{X}_1\,|\widehat{000000}\rangle + \beta E'\hat{X}_1\hat{X}_6\,|\widehat{000000}\rangle)\,|0\rangle_B\big].
\end{aligned} \tag{8.8}$$

By comparing this state with $|\hat{\psi}\rangle_L$ given in (8.7), it follows that for these two states to be identical, it is necessary that $E'\hat{X}_1\,|\widehat{000000}\rangle = |\widehat{000000}\rangle$, however, it may not be sufficient. Thus, this condition is now examined:

$$E'\hat{X}_1\,|\widehat{000000}\rangle \overset{(a)}{=} (X_1X_3)(X_3X_4X_6)\,|\widehat{000000}\rangle = X_1X_4X_6\,|\widehat{000000}\rangle\,,$$

where (a) follows from $\hat{X}_1 = X_3X_4X_6$ as stated in (8.1). The constraint is clearly not satisfied as stated on the form above. By utilising that $|\widehat{000000}\rangle$ is stabilized by $\{\hat{Z}_i\}_i$ by construction, one can right-multiply $E'\hat{X}_1$ with elements of $\{\hat{Z}_i\}_i$ without changing the state. In other words, the constraint is satisfied if there exists $i_1, \ldots, i_6 \in \{0,1\}$ such that

$$E'\hat{X}_1\hat{Z}_1^{i_1}\cdots\hat{Z}_6^{i_6}\,|\widehat{000000}\rangle = |\widehat{000000}\rangle \implies E'\hat{X}_1\hat{Z}_1^{i_1}\cdots\hat{Z}_6^{i_6} = I.$$

By writing out the operator $E'\hat{X}_1\hat{Z}_1^{i_1}\cdots\hat{Z}_6^{i_6}$, it can be determined when the latter condition is satisfied, if it even is possible. That is,

$$\begin{aligned}
I &= E'\hat{X}_1\hat{Z}_1^{i_1}\cdots\hat{Z}_6^{i_6} \\
&= (X_1X_3)(X_3X_4X_6)(Z_3Z_4Z_6)^{i_1}(Z_1Z_3Z_5Z_6)^{i_2}(X_1X_3X_5X_6)^{i_3}(Z_2Z_4Z_5Z_6)^{i_4}(X_2X_4X_5X_6)^{i_5}(X_3X_4X_5)^{i_6} \\
&\overset{(a)}{=} (XZ^{i_2}X^{i_3}) \otimes (Z^{i_4}X^{i_5}) \otimes (XXZ^{i_1}Z^{i_2}X^{i_3}X^{i_6}) \otimes (XZ^{i_1}Z^{i_4}X^{i_5}X^{i_6}) \\
&\quad\ \otimes (Z^{i_2}X^{i_3}Z^{i_4}X^{i_5}X^{i_6}) \otimes (XZ^{i_1}Z^{i_2}X^{i_3}Z^{i_4}X^{i_5}),
\end{aligned}$$

where (a) follows from writing the operators for each qubit together. The constraint is only satisfied when all qubits are affected by the identity, which is when there is an even amount of both X and Z operators on each qubit. Thus, the first qubit implies that $i_2 = 0, i_3 = 1$, while the second qubit implies that $i_4 = i_5 = 0$. Using these, qubit three then implies that $i_1 = 0, i_6 = 1$. By using these in the remaining three qubits, it is seen that $i_1 = i_2 = i_4 = i_5 = 0, i_3 = i_6 = 1$ in fact is the only solution to the constraint. Thus, $E'\hat{X}_1\hat{Z}_3\hat{Z}_6 = I$, which in turn implies that $E'\hat{Z}_3\hat{Z}_6 = \hat{X}_1$ since $\hat{X}_1$ commutes with both $\hat{Z}_3$ and $\hat{Z}_6$ by construction. By utilising this in all terms of (8.8), it can be rewritten as

$$\begin{aligned}
E'E\left|\hat{\psi}\right\rangle_L &= \frac{1}{\sqrt{2}}\big[(\alpha E'\hat{Z}_3\hat{Z}_6\left|\widehat{000000}\right\rangle + \beta E'\hat{X}_6\hat{Z}_3\hat{Z}_6\left|\widehat{000000}\right\rangle)\left|1\right\rangle_B \\
&\qquad +(\alpha E'\hat{X}_1\hat{Z}_3\hat{Z}_6\left|\widehat{000000}\right\rangle + \beta E'\hat{X}_1\hat{X}_6\hat{Z}_3\hat{Z}_6\left|\widehat{000000}\right\rangle)\left|0\right\rangle_B\big] \\
&\overset{(a)}{=} \frac{1}{\sqrt{2}}\big[(\alpha\hat{X}_1\left|\widehat{000000}\right\rangle - \beta\hat{X}_1\hat{X}_6\left|\widehat{000000}\right\rangle)\left|1\right\rangle_B + (\alpha\left|\widehat{000000}\right\rangle - \beta\hat{X}_6\left|\widehat{000000}\right\rangle)\left|0\right\rangle_B\big] \\
&= \frac{1}{\sqrt{2}}\big[(\alpha\left|\widehat{100000}\right\rangle - \beta\left|\widehat{100001}\right\rangle)\left|1\right\rangle_B + (\alpha\left|\widehat{000000}\right\rangle - \beta\left|\widehat{000001}\right\rangle)\left|0\right\rangle_B\big] \\
&\overset{(b)}{=} \left|\Phi^+\right\rangle \otimes \left|0\right\rangle^{\otimes 4} \otimes (Z\left|\psi\right\rangle),
\end{aligned}$$

where (a) follows from $\hat{X}_6$ anti-commuting with $\hat{Z}_6$ since they form a symplectic pair and (b) from comparing the penultimate state with (8.6). The result of this analysis naturally corresponds to the prior analysis since it merely as a reformulation of the problem. However, this second approach is less tedious since it essentially only requires one to determine which stabilizers in $\{\hat{Z}_i\}_i$ that can be used to reformulate $E'E\left|\hat{\psi}\right\rangle_L$ such that it resembles $\left|\hat{\psi}\right\rangle_L$. This approach is therefore used henceforth, where some of the details are omitted, since the idea of the method should be clear from the above analysis.

The performance of Bob's decoding has so far been analysed with the assumption that Bob knows the crossover probabilities for transmission and storage. In that case, he knows whether to correct using X_7 such that he obtains $\left|\psi\right\rangle$ after decoding, or to 'correct' with X_1X_3, in which case he obtains $Z\left|\psi\right\rangle_L$. Bob does, however, probably not know the crossover probabilities perfectly in practice, hence he cannot determine whether to correct with X_7 or X_1X_3. In that case, Bob can only determine $\left|\psi\right\rangle$ up to a phase flip, which is better than having no information about $\left|\psi\right\rangle$, however, it is not optimal. One approach to overcome this is to perform another syndrome measurement. Since he knows that the error either is X_7 or X_1X_3, he can, in theory, simply perform a syndrome measurement of the seventh qubit by using Z_7 as an observable. If he measures -1 (or $+1$), he knows with certainty that the occurred error is X_7 (or X_1X_3), and can then correct the error properly. In practice, Bob wants to perform the error-correction as quickly as possible such that decoherence is less likely to occur, hence performing the additional syndrome measurement may lead to addition errors that is unaccounted for.

Having discussed the performance of the $[[6,1,3;1]]$ EA code for a bit flip on Bob's ebit, the performance with arbitrary error patterns are considered to see whether the results holds in general.

8.3.3 General Error Patterns

As previously, consider the case where the error E happens such that it is corrected with E'. First, consider the case where only the transmitted qubits are corrupted by E, hence $E'E \in \mathcal{G}_6$, where $\mathcal{G}_6$ can be generated by $\{\hat{Z}_1, \ldots, \hat{Z}_6, \hat{X}_1, \ldots, \hat{X}_6\}$ given in (8.1). In that case, the state after error correction is simply

$$\begin{aligned}
E'E\left|\hat{\psi}\right\rangle_L = \frac{1}{\sqrt{2}}\big[&(\alpha E'E\left|\widehat{000000}\right\rangle + \beta E'E\hat{X}_6\left|\widehat{000000}\right\rangle)\left|0\right\rangle_B \\
&+(\alpha E'E\hat{X}_1\left|\widehat{000000}\right\rangle + \beta E'E\hat{X}_1\hat{X}_6\left|\widehat{000000}\right\rangle)\left|1\right\rangle_B\big].
\end{aligned}$$

The error-correcting capabilities of this code is then analysed by considering different possibilities of $E'E$. It turns out that there are seven possibilities, which are obtained by splitting $\mathcal{G}_6$ into subgroups based on the generators. Thus, let

$$\mathcal{S}_1 = I, \mathcal{S}_2 = \langle\hat{Z}_2, \ldots, \hat{Z}_5\rangle, \mathcal{S}_3 = \langle\hat{X}_2, \ldots, \hat{X}_5\rangle, \mathcal{S}_4 = \langle\hat{Z}_6\rangle, \mathcal{S}_5 = \langle\hat{X}_6\rangle, \mathcal{S}_6 = \langle\hat{Z}_1\rangle, \mathcal{S}_7 = \langle\hat{X}_1\rangle.$$

Each of these possibilities are now considered separately:

i) If $E'E \in \mathcal{S}_1$, then it clearly holds that $E'E\left|\hat{\psi}\right\rangle_L = \left|\hat{\psi}\right\rangle_L$ as desired.

ii) If $E'E \in \mathcal{S}_2 \setminus \mathcal{S}_1$, then $E'E$ commutes with both $\hat{X}_1$ and $\hat{X}_6$, which implies that $E'E$ can be annihilated by $\left|\widehat{000000}\right\rangle$ in every term of $E'E\left|\hat{\psi}\right\rangle_L$ due to $\left|\widehat{000000}\right\rangle$ being stabilized by $\{\hat{Z}_1, \ldots, \hat{Z}_5\}$ by construction. Thus, $E'E\left|\hat{\psi}\right\rangle_L = \left|\hat{\psi}\right\rangle_L$ as desired.

iii) If $E'E \in \mathcal{S}_3 \setminus \mathcal{S}_1$, then the only change in $E'E\,|\hat{\psi}\rangle_L$ is that some additional ones comes into the logical basis states, i.e., $\{\hat{i}\}_i$. Since the same change happens in every term, this has no effect on the decoding, hence $E'E\,|\hat{\psi}\rangle_L = |\hat{\psi}\rangle_L$ as desired. For example, if $E'E = \hat{X}_2$, then

$$\begin{aligned} E'E\,|\hat{\psi}\rangle_L &= \frac{1}{\sqrt{2}}\big[(\alpha\hat{X}_2\,|\widehat{000000}\rangle + \beta\hat{X}_2\hat{X}_6\,|\widehat{000000}\rangle)\,|0\rangle_B \\ &\qquad +(\alpha\hat{X}_1\hat{X}_2\,|\widehat{000000}\rangle + \beta\hat{X}_1\hat{X}_2\hat{X}_6\,|\widehat{000000}\rangle)\,|1\rangle_B\,\big] \\ &= \frac{1}{\sqrt{2}}\big[(\alpha\,|\widehat{010000}\rangle + \beta\,|\widehat{010001}\rangle)\,|0\rangle_B + (\alpha\,|\widehat{110000}\rangle + \beta\,|\widehat{110001}\rangle)\,|1\rangle_B\,\big]. \end{aligned}$$

By applying $(U \otimes I)$ to decode, the states are simply changed to the corresponding basis state in $\{|i\rangle\}_i$. Since the second qubit merely is an ancillary qubit, it has no impact on the decoding, hence Bob will obtain $|\psi\rangle$ by performing a full decoding.

iv) If $E'E \in \mathcal{S}_4 \setminus \mathcal{S}_1$, then $E'E$ will anti-commute with $\hat{X}_6$, hence it will flip the sign of the terms corresponding to β in $E'E\,|\hat{\psi}\rangle_L$ in order to be annihilated by $|\widehat{000000}\rangle$. Thus, Bob obtains $|\hat{\psi}\rangle_L$ with a phase flip, i.e., $\bar{Z}\,|\hat{\psi}\rangle_L$ by using the notation from Chapter 6.

v) If $E'E \in \mathcal{S}_5 \setminus \mathcal{S}_1$, the $\hat{X}_6$ factors are moved from the terms corresponding to β to those corresponding to α. Since the sixth qubit essentially contains $|\psi\rangle$ after decoding, this would correspond to a bit flip of $|\hat{\psi}\rangle_L$, i.e., one obtain $\bar{X}|\hat{\psi}\rangle_L$.

vi) If $E'E \in \mathcal{S}_6 \setminus \mathcal{S}_1$, then $E'E$ anti-commutes with $\hat{X}_1$, hence the sign of the terms corresponding to $|1\rangle_B$ is flipped in order for $E'E$ to be annihilated by $|\widehat{000000}\rangle$. This essentially corresponds to applying a phase flip to the ebit possessed by Bob. However, by the discussion at the start of Section 8.3, such phase flip on Bob's ebit has no impact on the decoding. Hence, $E'E\,|\hat{\psi}\rangle_L = |\hat{\psi}\rangle_L$.

vii) If $E'E \in \mathcal{S}_7 \setminus \mathcal{S}_1$, the $\hat{X}_1$ factors are moved from the terms corresponding to $|1\rangle_B$ to those corresponding to $|0\rangle_B$. By similar reasoning as above, this is equivalent to a bit flip on Bob's ebit, which again has no impact on the decoding, hence $E'E\,|\hat{\psi}\rangle_L = |\hat{\psi}\rangle_L$.

Using the decomposition of the generators of $\mathcal{G}_6$ made in the points, taking combinations of them will simply apply both of there affects on $|\hat{\psi}\rangle_L$. For example, if $E'E = \hat{Z}_2\hat{X}_4$, then it will have no impact on $|\hat{\psi}\rangle_L$, however if $E'E = \hat{Z}_6\hat{X}_6$, then $|\hat{\psi}\rangle_L$ will be corrupted by both a phase flip and a bit flip, i.e., $\bar{Y}$. This decomposition has two important implications. Firstly, since $\hat{X}_1$ and $\hat{Z}_1$ correspond to applying a phase flip and bit flip to the seventh qubit, respectively, it is in some sense sufficient to only consider $\mathcal{G}_6$, hence the above decomposition, in order to describe errors that may corrupt $|\hat{\psi}\rangle_L$. For example, a bit flip on the seventh qubit can simply be considered by letting $E'E = \hat{Z}_1$, while phase flip on the first, fourth, fifth and seventh qubit can be considered as $E'E = \hat{Z}_2$ since $E'E = \hat{Z}_1\hat{Z}_2Z_7$ such that $\hat{Z}_1$ cancels with Z_7. Secondly, the decomposition allows a fairly simple analysis of the decoding of different error patterns since it only requires one to locate which subgroup, $\mathcal{S}_1, \dots, \mathcal{S}_6$, or combinations hereof, that $E'E$ belong to. That corresponds to solving for $i_1, \dots, i_6, j_1, \dots, j_6 \in \{0, 1\}$ in

$$E'E = \hat{Z}_1^{i_1} \cdots \hat{Z}_6^{i_6}\hat{X}_1^{j_1} \cdots \hat{X}_6^{j_6}. \tag{8.9}$$

Since all errors except those containing either $\hat{Z}_6$ and/or $\hat{X}_6$ have no impact on $|\hat{\psi}\rangle_L$, it follows that

$$E'E\,|\hat{\psi}\rangle_L = \begin{cases} |\hat{\psi}\rangle_L\,, & \text{if } (i_6, j_6) = (0,0), \\ \bar{X}\,|\hat{\psi}\rangle_L\,, & \text{if } (i_6, j_6) = (0,1), \\ \bar{Z}\,|\hat{\psi}\rangle_L\,, & \text{if } (i_6, j_6) = (1,0), \\ \bar{Y}\,|\hat{\psi}\rangle_L\,, & \text{if } (i_6, j_6) = (1,1). \end{cases} \tag{8.10}$$

There are clearly 2^{10} error patterns in each of these mappings, which expands to 2^{12} when considering $\mathcal{G}_7$ by using the discussion about errors on Bob's ebit being equivalent to $\hat{Z}_1, \hat{X}_1$, or $\hat{Z}_1\hat{X}_1$ depending on the error corrupting the ebit. More precisely, since every error in $\mathcal{G}_6$ can be written on the form given in (8.9), the errors on Bob's ebit can be included by including the factor $\hat{Z}_1^{i_7}\hat{X}_1^{j_7}$ such that all 2^{14} possible error pattern is accounted for. But the factors in (8.9) corresponding to $\hat{Z}_1$ and $\hat{X}_1$ can then be written as $\hat{Z}_1^{i_1+i_7}$ and $\hat{Z}_1^{j_1+j_7}$, respectively. For example, since $X_1X_3 = \hat{Z}_3\hat{Z}_6\hat{X}_1$, then $X_1X_3X_7 = \hat{Z}_3\hat{Z}_6\hat{X}_1\hat{X}_1$, which is equivalent to $\hat{Z}_3\hat{Z}_6 = X_1X_4X_6$. Thus, by only considering $E'E$ on the form given in (7.7), all 2^{14} error patterns in $\mathcal{G}_7$ are accounted for.

The question of interest now is how likely each of the mappings in (8.10) are. This is analysed under the assumption that the crossover probabilities, p_T and p_S are sufficiently small such that it is unlikely that more than two errors occur. That is, approximate probabilities for each of these mappings are calculated by considering possible error patterns of up to weight two.

Before considering specific error patterns, it should be explicitly noted that the $[[6,1,3;1]]$ EA code is designed such that for any given non-trivial error syndrome, it will almost always assume that an error of weight one has occurred if only bit flips or phase flips occur. If both types happens on distinct errors, then an error of weight two will be most likely. In contrast, if the Steane code was considered, this would always hold due to being constructed from the $[7,4]$ Hamming code. However, for the $[[6,1,3;1]]$, if the error syndrome detects that an error may have occurred on the seventh qubit, then it is most likely that the error actually has been on the seventh qubit if $p_1 > p_2$ as depicted in Figure 8.1, while some error of weight two is more likely if the inequality sign is flipped. Based on this, it is fairly straight-forward to analyse how the $[[6,1,3;1]]$ maps $|\hat{\psi}\rangle_L$ for small-weight errors for which the results are summarised in Table 8.1.

Error of Weight Zero

If an error of weight zero occurs, i.e., $E = I$, then $E' = I$ naturally hold. This scenario occurs with probability $(1-p_T)^6(1-p_S)$.

Errors of Weight One

If an error of weight one occurs, i.e., $E \in \{X_1,\ldots,X_7,Z_1,\ldots,Z_7,Y_1,\ldots,Y_7\}$, then $E' = E$ at all times unless the seventh qubit is corrupted. If the seventh qubit is corrupted, then $E' = E$ only if $p_1 > p_2$. If $p_1 > p_2$, then $E'E = I$ in all cases such that $E'E$ acts as the identity on $|\hat{\psi}\rangle_L$. This scenario happens with probability

$$6p_T(1-p_T)^5(1-p_S) + (1-p_T)^6 p_S.$$

If $p_1 < p_2$ and $E \in \{X_7, Z_7, Y_7\}$, then Bob 'corrects' with $E' \neq E$. More precisely, if $E = X_7$, then $E' \in \{X_1X_3, X_2X_4, X_5X_6\}$ as previously discussed. It holds analogously for $E = Z_7$ and $E = Y_7$. Looking at the generators of $\mathcal{G}_6$, the above error patterns can be rewritten as

$$X_1X_3 = \hat{Z}_3\hat{Z}_6\hat{X}_1, \quad X_2X_4 = \hat{Z}_5\hat{Z}_6\hat{X}_1, \quad X_5X_6 = \hat{Z}_1\hat{Z}_6\hat{X}_1,$$
$$Z_1Z_3 = \hat{Z}_1\hat{Z}_4\hat{X}_6, \quad Z_2Z_4 = \hat{Z}_1\hat{Z}_2\hat{X}_6, \quad Z_5Z_6 = \hat{Z}_1\hat{Z}_2\hat{Z}_4\hat{X}_6.$$

Hence, in all cases where $E = X_7$, then $E'E|\hat{\psi}\rangle_L = \bar{Z}|\hat{\psi}\rangle_L$, while it in cases where $E = Z_7$ follows that $E'E|\hat{\psi}\rangle_L = \bar{X}|\hat{\psi}\rangle_L$. By combining these, it also holds that $E = Y_7$ implying that $E'E|\hat{\psi}\rangle_L = \bar{Y}|\hat{\psi}\rangle_L$. Combining all of the above results, it holds that when $p_1 < p_2$, then $E'E$ maps $|\hat{\psi}\rangle_L$ to $|\hat{\psi}\rangle_L$ with probability $6p_T(1-p_T)^5(1-p_S)$, while for $p_1 < p_2$, it maps to $\bar{X}|\hat{\psi}\rangle_L$, $\bar{Z}|\hat{\psi}\rangle_L$, and $\bar{Y}|\hat{\psi}\rangle_L$ with probability $(1-p_T)^6 p_S/3$ each.

Errors of Weight Two

If an error of weight two occurs, i.e.,

$$E \in \{A_iA_j, A_iB_k \mid A,B \in \{X,Z,Y\}, i,j,k = 1,\ldots,7 \text{ satisfying } i<j, i \neq k, A \neq B\},$$

the errors can be grouped into six groups; a) two bit flips, b) two phase flips, c) two phase-flips, d) a bit flip and a phase flip, e) a bit flip and a bit-phase flip, and f) a phase flip and a bit-phase flip, where the errors in d)-f) naturally occur on different qubits such that the error has weight two. Since a bit-phase flip corresponds to a bit flip and a phase flip on the same qubit, the mapping for errors in c)-f) can be obtained by only considered the errors in a) and b). This is further elaborated upon after considered how the errors in a) and b) map $|\hat{\psi}\rangle_L$. Firstly, the errors in a) are considered. Since there are $\binom{7}{2} = 21$ error patterns of weight two containing only bit flips, only seven error patterns lead to distinct decompositions of generators by symmetry. That is, if $E = X_1X_2$, then Bob would 'detect' the error to be $E' = X_5$, but by symmetry it then also holds that if the error was $E = X_1X_5$, then Bob would detect $E' = X_2$ to be the occurred error. In both cases $E'E = X_1X_2X_5$, hence the mappings of both errors are identical. Thus, it is sufficient to only consider the following seven error patterns, where their detected error, E', also has been included:

$$E = X_1X_2 \implies E' = X_5, \quad E = X_1X_3 \overset{(a)}{\implies} E' \in \{X_7, X_1X_3\}, \quad E = X_1X_4 \implies E' = X_6,$$
$$E = X_2X_3 \implies E' = X_6, \quad E = X_2X_4 \overset{(a)}{\implies} E' \in \{X_7, X_1X_3\}, \quad E = X_3X_4 \implies E' = X_5,$$
$$E = X_5X_6 \overset{(a)}{\implies} E' \in \{X_7, X_5X_6\},$$

where E' in (a) depends on whether $p_1 > p_2$ or not. To analyse the error patterns that depend on p_1 and p_2, i.e., $E \in \{X_1X_3, X_1X_7, X_3X_7, X_2X_4, X_2X_7, X_4X_7, X_5X_6, X_5X_7, X_6X_7\}$, it is sufficient to only analyse the effect of the first three, whereafter the remaining follow analogously. If $E = X_1X_3$, then Bob detects $E' = X_1X_3$ if $p_1 < p_2$, in which case $E'E = I$, hence maps $|\hat{\psi}\rangle_L$ to itself. On the other hand, if $p_1 > p_2$, he wrongly detects $E' = X_7$ such that $E'E = X_1X_3X_7$. This is similar to the remaining error patterns, where an error of weight two is wrongly detected to be of weight one, hence is analysed later. If $E = X_1X_7$, Bob will always wrongly detect that $E' = X_3$ since one error on the transmitted qubits is more likely than an error on the transmitted combined with an error on Bob's stored ebit. Similarly, if $E = X_3X_7$, then Bob wrongly detects $E' = X_1$. These are also similar to the other one error patterns, hence analysed with the remaining error patterns. Thus, it has so far only been deduced that if $p_1 < p_2$ and $E \in \{X_1X_3, X_2X_4, X_5X_6\}$, then $E'E$ maps $|\hat{\psi}\rangle_L$ to itself. This set of errors occur with probability $3\left(\frac{p_T}{3}\right)^2(1-p_T)^4(1-p_S) = (p_T^2(1-p_T)^4(1-p_S))/3$, which then can be multiplied by three to include the corresponding error patterns containing phase flips and bit-phase flips. In all other cases, any two-weight error of bit flips will be detected as a different single qubit error. The effect of doing so is now analysed. By the previous discussion, it is sufficient to consider

$$E'E \in \{X_1X_2X_5, X_1X_3X_7, X_1X_4X_6, X_2X_3X_6, X_2X_4X_7, X_3X_4X_5, X_5X_6X_7\}.$$

All of these elements are now written in terms of the generators:

$$X_1X_2X_5 = \hat{Z}_3\hat{Z}_5\hat{Z}_6, \quad X_1X_3X_7 \overset{(a)}{=} \hat{Z}_3\hat{Z}_6\hat{X}_1X_7, \quad X_1X_4X_6 = \hat{Z}_3\hat{Z}_6,$$
$$X_2X_3X_6 = \hat{Z}_5\hat{Z}_6, \quad X_2X_4X_7 \overset{(a)}{=} \hat{Z}_5\hat{Z}_6\hat{X}_1X_7, \quad X_3X_4X_5 = \hat{Z}_6, \quad X_5X_6X_7 \overset{(a)}{=} \hat{Z}_6\hat{X}_1X_7,$$

where (a) follows by explicitly writing X_7 along with the generators for the corresponding errors on the first six qubits for clarification. However, since X_7 is equivalent to $\hat{X}_1$, both of these factors could be omitted without loss of generality. Since all of the above possibilities for $E'E$ contain $\hat{Z}_6$, but not $\hat{X}_6$, they all map $|\hat{\psi}\rangle_L$ to $\bar{Z}\,|\hat{\psi}\rangle_L$. Since there are $\binom{6}{2} = 15$ of such error patterns not containing X_7 and six that contain it, the probability that an error with two bit flips occurring is

$$15\left(\frac{p_T}{3}\right)^2(1-p_T)^4(1-p_S) + 6\frac{p_T}{3}(1-p_T)^5\frac{p_S}{3} = \frac{1}{3}(5p_T^2(1-p_T)^4(1-p_S) + 2p_T(1-p_T)^5p_S).$$

In the case where $p_1 > p_2$, it is always assumed that one bit flip has occurred, hence $E'E$ will map $|\hat{\psi}\rangle_L$ to $\bar{Z}\,|\hat{\psi}\rangle_L$ with the probability given above. However, if $p_1 < p_2$, then $E'E$ maps $|\hat{\psi}\rangle_L$ to itself if $E \in \{X_1X_3, X_2X_4, X_5X_6\}$, and to $\bar{Z}\,|\hat{\psi}\rangle_L$ if any error two-weight bit flip error happens. These mappings occur with probability $(p_T^2(1-p_T)^4(1-p_S)/3$ and $(4p_T^2(1-p_T)^4(1-p_S)+2p_T(1-p_T)^5p_S)/3$, respectively.

By similar reasoning, it can be deduced that it for b) is sufficient to only consider the same error patterns, but naturally changing every X to an Z. The error patterns can then be written as

$$Z_1Z_2Z_5 = \hat{X}_6, \quad Z_1Z_3Z_7 \overset{(a)}{=} \hat{Z}_1\hat{Z}_4\hat{X}_6Z_7, \quad Z_1Z_4Z_6 = \hat{Z}_4\hat{X}_6, \quad Z_2Z_3Z_6 = \hat{Z}_2\hat{X}_6,$$
$$Z_2Z_4Z_7 \overset{(a)}{=} \hat{Z}_1\hat{Z}_2\hat{X}_6Z_7, \quad Z_3Z_4Z_5 = \hat{Z}_2\hat{Z}_4\hat{X}_6, \quad Z_5Z_6Z_7 \overset{(a)}{=} \hat{Z}_1\hat{Z}_2\hat{Z}_4\hat{X}_6Z_7,$$

where (a) once again follow by writing both $\hat{Z}_1$ and Z_7 although they essentially cancel each other. Since all of these factors contain $\hat{X}_6$ but not $\hat{Z}_6$, the conclusion is the similar as the one for two bit flips by changing the mapping of $E'E$ to $\bar{X}\,|\hat{\psi}\rangle_L$.

By combining the two results, it follows that errors in c), which contain two bit-phase flips, the conclusion for the mapping is again similar by changing it to $\bar{Y}\,|\hat{\psi}\rangle_L$.

Now that errors of weight two with only one type of Pauli error has been discussed, it is considered how $|\hat{\psi}\rangle_L$ is mapped when two different types occur, namely by considering d), where a bit flip and a phase flip occur. Say $E = X_1Z_2$ happens. Then, Bob can detect that the state has been corrupted by both a bit flip and a phase flip (or bit-phase flips). In other words, no error of weight less than two can yield the obtained syndrome. However, there are three possible weight two errors yielding the same syndrome, that is $E' \in \{X_1Z_2, Y_1Z_5, Y_2X_5\}$ since the fifth sum of the check matrix of $\mathcal{S}^e$ is the sum of the first and the second. This implies that $E'E \in \{I, Z_1Z_2Z_5, X_1X_2X_5\}$, which by the previous discussion implies that the mapping of $E'E$ on $|\hat{\psi}\rangle_L$ is either $I, \bar{X}$, or $\bar{Z}$, respectively. As all of these errors has the same weight and only corrupt transmitted qubits, they are equally likely to occur, hence Bob cannot uniquely determine E', which in turn imply that it cannot be uniquely determined how $E'E$ acts upon $|\hat{\psi}\rangle_L$. If Bob randomly chooses to correct any of the three errors, $E'E$ will for this particular error pattern act as $I, \bar{Z}, \bar{X}$ on $|\hat{\psi}\rangle_L$ each with probability $\frac{1}{3}\left(\frac{p_T}{3}\right)^2(1-p_T)^4(1-p_S) = (p_T^2(1-p_T)^4(1-p_S))/27$ on average. If instead the error is $E = X_1Z_3$, then $E' \in \{X_1Z_3, Y_3Z_7, X_1Y_7\}$, in which case Bob detects $E' = X_1Z_3$ since $p_S < p_T$ by assumption. Thus, $E'E$ acts as the identity on $|\hat{\psi}\rangle_L$ for this error pattern, which occur with probability $(p_T^2(1-p_T)^4(1-p_S))/9$. Thus, there are two cases for error patterns on the transmitted qubits that contain both a bit flip and a phase flip. If $E \in \{X_1Z_3, X_2Z_4, X_5Z_6, Z_1X_3, Z_2X_4, Z_5X_6\}$, then $E'E$ acts as the identity on $|\hat{\psi}\rangle_L$, which occur with probability $(2p_T^2(1-p_T)^4(1-p_S))/3$. In all of the 24 other cases, E' cannot be uniquely determined, but by randomly choosing one of the possible errors, $E'E$ will on average map as $I, \bar{Z}, \bar{X}$ each with probability $(8p_T^2(1-p_T)^4(1-p_S))/9$.

By symmetry, if one of the errors occur on Bob's ebit, say $E = X_1Z_7$, then $E' \in \{X_1Z_7, Y_1Z_3, X_3Y_7\}$ such that $E' = Y_1Z_3$ with certainty. But then, $E'E = Z_1Z_3Z_7$, which maps $|\hat{\psi}\rangle_L$ to $\bar{X}\,|\hat{\psi}\rangle_L$ by the above arguments. This holds for all cases where one of the transmitted qubits is bit flipped and Bob's ebit is phase flipped, which occur with probability $6\left(\frac{p_T}{3}\right)^2(1-p_T)^4(1-p_S) = (2p_T^2(1-p_T)^4(1-p_S))/3$. Analogously, if a transmitted qubit is phase flipped and Bob's ebit is bit flipped, then $E'E\,|\hat{\psi}\rangle_L = \bar{Z}\,|\hat{\psi}\rangle_L$ with the same probability.

If instead a bit-phase flip occurs on one of the transmitted qubits, say $E = X_1Y_2$, then $E' \in \{X_1Y_2, Z_2X_5, Z_1Y_5\}$ such that he cannot uniquely determined the error once again. Similarly, as before $E'E \in \{I, X_1X_2X_5, Y_1Y_2Y_5\}$, hence $E'E$ maps as $I, \bar{Z}, \bar{Y}$, respectively. Thus, the conclusion is completely equivalent to before by changing the possible mapping $\bar{X}$ to $\bar{Y}$ due to changing the error from a phase flip to a bit-phase flip. Hence, the case where one of the errors is on Bob's ebit or when a phase flip and a bit-phase flip occurs also follows from the above analysis by changing the mappings appropriately.

Approximate Probabilities for the Mappings

The different probabilities obtained in the above analysis are now summarised in Table 8.1, where the first column contains possible error sets, the second how $E'E$ maps $|\hat{\psi}\rangle_L$ for a given error set, the third the probability of this mapping, and the last under which conditions the mapping holds.

Based on the probabilities summarised in Table 8.1, the probabilities for the different mappings can be written explicitly as

$$\begin{aligned}
\Pr(E'E = I) &= (1-p_T)^4(1-p_S)\left[(1-p_T)^2 + 6p_T(1-p_T) + \frac{14}{3}p_T^2\right] \\
&\quad + \chi_{[p_1>p_2]}\left[(1-p_T)^6 p_S\right] + \chi_{[p_1<p_2]}\left[p_T^2(1-p_T)^4(1-p_S)\right], \\
\Pr\bigl(E'E = \bar{Z}\bigr) &= \Pr\bigl(E'E = \bar{X}\bigr) = \Pr\bigl(E'E = \bar{Y}\bigr) \\
&= \frac{2}{3}p_T(1-p_T)^4\left[\frac{14}{3}p_T(1-p_S) + 3(1-p_T)p_S\right] \\
&\quad + \chi_{[p_1>p_2]}\left[\frac{1}{3}p_T^2(1-p_T)^4(1-p_S)\right] + \chi_{[p_1<p_2]}\left[\frac{1}{3}(1-p_T)^6 p_S\right],
\end{aligned}$$

where $\chi_{[A]}$ is the indicator function yielding one when the condition A is satisfied and zero otherwise. The equalities above should be read as the approximate probabilities for the four mappings obtained by neglecting error patterns of weight three or more.

Before further analysing the above probabilities, a few remarks about the error sets in Table 8.1 are in order. Firstly, although the $[[6, 1, 3; 1]]$ EA code cannot always correct errors on Bob's ebit, it still has distance three as the distance only accounts for errors on the transmitted qubits in the EA coding

Error set	Mapping of $\vert\hat{\psi}\rangle_L$	Probability of error set	Condition for mapping
I	I	$(1-p_T)^6(1-p_S)$	None
A_i	I	$6p_T(1-p_T)^5(1-p_S)$	None
A_7	I	$(1-p_T)^6 p_S$	$p_1 > p_2$
A_7	$\bar{Z}, \bar{X}, \bar{Y}$	$\frac{1}{3}(1-p_T)^6 p_S$	$p_1 < p_2$
A_1A_3, A_2A_4, A_5A_6	$\bar{Z}, \bar{X}, \bar{Y}$	$\frac{1}{3}p_T^2(1-p_T)^4(1-p_S)$	$p_1 > p_2$
A_1A_3, A_2A_4, A_5A_6	I	$p_T^2(1-p_T)^4(1-p_S)$	$p_1 < p_2$
A_iA_j (not above)	$\bar{Z}, \bar{X}, \bar{Y}$	$\frac{4}{3}p_T^2(1-p_T)^4(1-p_S)$	None
A_iA_7	$\bar{Z}, \bar{X}, \bar{Y}$	$\frac{2}{3}p_T(1-p_T)^5 p_S$	None
A_1B_3, A_2B_4, A_5B_6	I	$2p_T^2(1-p_T)^4(1-p_S)$	None
A_iB_k (not above)	I	$\frac{8}{3}p_T^2(1-p_T)^4(1-p_S)$	Average
A_iB_k (not above)	$\bar{Z}, \bar{X}, \bar{Y}$	$\frac{16}{9}p_T^2(1-p_T)^4(1-p_S)$	Average
A_iB_7	$\bar{Z}, \bar{X}, \bar{Y}$	$\frac{4}{3}p_T(1-p_T)^5 p_S$	None

Table 8.1: Results of the analysis of the decoding performance of the $[[6,1,3;1]]$ EA code. The first column contains all possible errors of up to weight two separated into groups depending on how the map $|\hat{\psi}\rangle_L$. The column should be read with the convention that $A, B \in \{X, Z, Y\}$ for $A \neq B$, and $i, j, k = 1, \ldots, 6$ with $i < j$ and $i \neq k$. The second column contains how that particular error maps $|\hat{\psi}\rangle_L$ when it is being 'corrected', which for rows with more than one mapping should be read as different mappings depending on which of the multiple error set in the first column that is used. The third column contains how likely that error set is to occur with the given mapping, while the last column describes under which conditions the error set has that particular mapping, where 'None' should be read as there being no restrictions, hence such cases always hold.

framework. However, if $p_1 > p_2$, then all errors of weight one, including those on Bob's ebit, can be corrected such that it has distance three in the classical sense too. Secondly, some errors of weight two are correctable. Namely, if $E \in \{A_1B_3, A_2B_4, A_5B_6\}$ for $A, B \in \{X, Y, Z\}, A \neq B$, Bob can correct the error, while it for any other error of weight two that only corrupts the transmitted qubits with different error types is corrected sometimes due to the error detection being ambiguous. Hence, out of the 189 possible errors of weight two, 18 out of them are always corrected, while another 72 are corrected a third of the time on average. Thus, almost half of the errors of weight two can be corrected in theory. Lastly, an error on Bob's ebit is destructive in the sense that if it is corrupted, then Bob cannot correctly decode the state if one of the transmitted qubit also have been corrupted. In fact, even if the only error is on his ebit, he can only decode it correctly if $p_1 > p_2$.

Having highlighted some important remarks about the error sets, it it time to analyse how likely $E'E$ is to be each of the four mappings more illustratively. Hence, these probabilities are illustrated in Figure 8.2 as a function of $p_S \in [0, p_T)$ for a fixed $p_T = 0.25$. In that case, $p_1 > p_2$ when $p_S > 0.1$. Since the probabilities for the different mappings only are approximate as error patterns of weight three or higher has been neglected, the cumulative (approximate) probability of all the mappings is included to illustrate how likely errors of weight two or lower are for the given crossover probabilities. In order to visually illustrate the effect of increasing p_S such that the inequality in $p_1 < p_2$ flips to $p_1 > p_2$, the probability for the mappings in the case where $p_1 < p_2$ is continued with dashed curves even after the flip of the inequality.

From the figure, it can be seen that when $p_T = 0.25$ and $p_S = 0$, Bob decodes correctly around 62% of the time, while each of the logical Pauli operators is applied to the encoded state around 7% of the time. Hence, an error of weight two or less occurs a little over 83% of the time, which coincide with its theoretical value. It can furthermore be seen that as p_S increases, errors of higher weight becomes slightly more likely, the probability of decoding correctly decreases, and the probability of wrongly decoding increases, which is as expected. Even as p_S approaches p_T, the probability of correctly decoding is still above 50%. The last notable thing as that the blue dashed curve is below the filled blue curve, which implies that as p_S increases from zero to the point where the inequality flips in $p_1 > p_2$, i.e., $p_S = 0.1$, the the probability of decoding correctly decreases faster than when p_S increases from 0.1 to 0.25. Hence, a slight increase in p_S is more critical for the decoding performance when $p_S < 0.1$ than when $p_S > 0.1$.

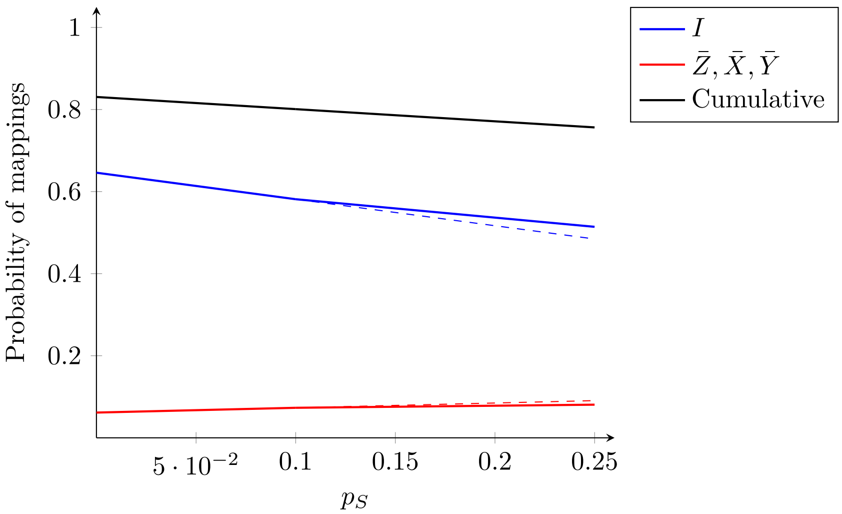


Figure 8.2: Curves of the probability for each of the mappings as a function of p_S together with the cumulative probability for these mappings for $p_T = 0.25$.

Having analysed the case where $p_T = 0.25$, another value of p_T is now used to see the effect of this change. Thus, Figure 8.3 illustrates the same curves in the case where $p_T = 0.1$, in which case $p_1 > p_2$ when $p_S > 0.0122$.

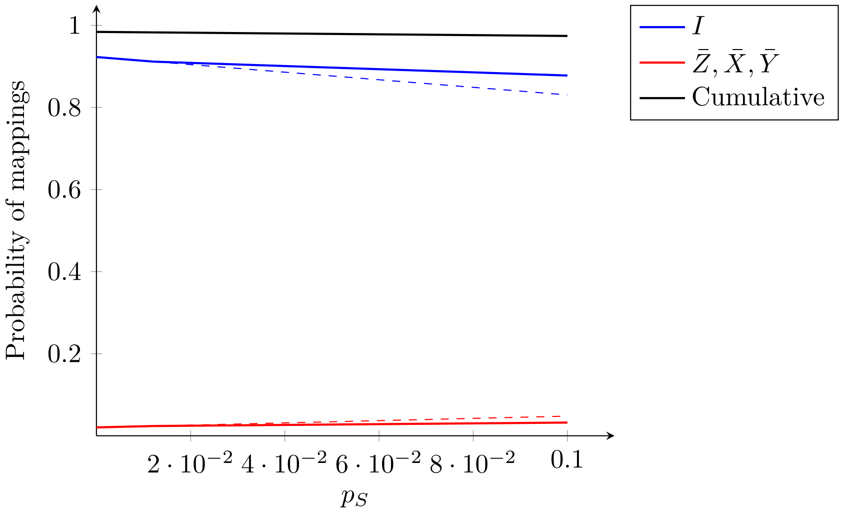


Figure 8.3: Curves of the probability for each of the mappings as a function of p_S together with the cumulative probability for these mappings for $p_T = 0.10$.

All the same conclusions can be drawn from this case, where the only change is that the performance has increased due to using lower crossover probabilities, which is as expected.

The conclusion of this analysis of the decoding performance for the $[[6, 1, 3; 1]]$ EA code is that it generally performs quite well as even many errors of weight two can be corrected. However, noise on Bob's ebit is critical as it prohibits correctly decoding unless only his ebit is corrupted and $p_1 > p_2$ as previously discussed. To see whether this generalises to cases with more ebits, a similar analysis is conducted for the $[[5, 1, 3; 2]]$ EA code.

8.4 Decoding Error Patterns with the $[[5, 1, 3; 2]]$ Code

Before performing the analysis, the $[[5, 1, 3; 2]]$ EA code is explicitly defined, however, without some details as the process was thoroughly described for the $[[6, 1, 3; 1]]$ EA code.

The initial non-abelian group for the code is $\mathcal{S} = \langle X_2X_3X_5, X_2X_4X_5, X_1X_3X_4X_5, Z_2Z_3Z_5, Z_2Z_4Z_5, Z_1Z_3Z_4Z_5 \rangle$. The two symplectic pairs and the two isotropic generators are obtained by permuting the ordering of the generators and re-labelling:

$$\mathcal{S} = \langle \hat{Z}_1, \hat{X}_1, \hat{Z}_2, \hat{X}_2, \hat{Z}_3, \hat{Z}_4 \rangle = \langle Z_2Z_3Z_5, X_2X_3X_5, Z_2Z_4Z_5, X_2X_4X_5, Z_1Z_3Z_4Z_5, X_1X_3X_4X_5 \rangle \,.$$

This set of generators is then expanded to one generating $\mathcal{G}_5$. It is easily verified that the following sets do so:

$$\begin{aligned} \{\hat{Z}_1, \ldots, \hat{Z}_5\} &= \langle Z_2Z_3Z_5, Z_2Z_4Z_5, Z_1Z_3Z_4Z_5, X_1X_3X_4X_5, X_1X_2X_5 \rangle \,, \\ \{\hat{X}_1, \ldots, \hat{X}_5\} &= \langle X_2X_3X_5, X_2x_4X_5, X_1, Z_2Z_5, Z_2Z_3Z_4 \rangle \,. \end{aligned}$$

Since $\mathcal{S}$ has two symplectic pairs and two isotropic generators, the logical state obtained from the simpler group $\mathcal{B}^e$ that is isomorphic to $\mathcal{S}$ on the first five qubits, which by (7.16) is given by

$$\begin{aligned} |\psi\rangle_L = |\Phi^+\rangle^{\otimes 2} \otimes |0\rangle^{\otimes 2} \otimes |\psi\rangle = \frac{1}{2}\big[& (\alpha\,|00000\rangle + \beta\,|00001\rangle)\,|00\rangle_B + (\alpha\,|01000\rangle + \beta\,|01001\rangle)\,|01\rangle_B \\ & (\alpha\,|10000\rangle + \beta\,|10001\rangle)\,|10\rangle_B + (\alpha\,|11000\rangle + |11001\rangle)\,|11\rangle_B \big], \end{aligned}$$

where the qubits has been permuted such that Bob's ebits are on the right. To obtain $|\hat{\psi}\rangle_L$, the simultaneous +1-eigenspace of $\{\hat{Z}_i\}_i$ is needed. By similar reasoning as for the other code, it can be deduced that the logical ground state has the form

$$|\widehat{00000}\rangle = \frac{1}{2}[|00000\rangle + |01110\rangle + |10111\rangle + |11001\rangle].$$

However, it will not be explicitly needed henceforth as it has been argued that it is sufficient to analyse the performance of the code in the $\{|\hat{i}\rangle\}_i$ basis, for which $|00000\rangle$ nonetheless is the building block. By encoding $|\psi\rangle_L$ with $(U^\dagger \otimes I)$, the logical state is seen to be

$$\begin{aligned} |\hat{\psi}\rangle_L = \frac{1}{2}\big[& (\alpha\,|\widehat{00000}\rangle + \beta\,|\widehat{00001}\rangle)\,|00\rangle_B + (\alpha\,|\widehat{01000}\rangle + \beta\,|\widehat{01001}\rangle)\,|01\rangle_B \\ & (\alpha\,|\widehat{10000}\rangle + \beta\,|\widehat{10001}\rangle)\,|10\rangle_B + (\alpha\,|\widehat{11000}\rangle + \beta\,|\widehat{11001}\rangle)\,|11\rangle_B \big] \\ = \frac{1}{2}\big[& (\alpha\,|\widehat{00000}\rangle + \beta\hat{X}_5\,|\widehat{00000}\rangle)\,|00\rangle_B + (\alpha\hat{X}_2\,|\widehat{00000}\rangle + \beta\hat{X}_2\hat{X}_5\,|\widehat{00000}\rangle)\,|01\rangle_B \\ & (\alpha\hat{X}_1\,|\widehat{00000}\rangle + \beta\hat{X}_1\hat{X}_5\,|\widehat{00000}\rangle)\,|10\rangle_B + (\alpha\hat{X}_1\hat{X}_2\,|\widehat{00000}\rangle + \beta\hat{X}_1\hat{X}_2\hat{X}_5\,|\widehat{00000}\rangle)\,|11\rangle_B \big]. \end{aligned}$$

The stabilizers for the code is $\mathcal{S}^e = \langle Z_2Z_3Z_5Z_6, X_2X_3X_5X_6, Z_2Z_4Z_5Z_7, X_2X_4X_5X_7, Z_1Z_3Z_4Z_5, X_1X3X_4X_5 \rangle$. By the same reasoning as in the previous analysis, errors on Bob's ebits are in principle not corrupting the information of the state, hence it is sufficient to only consider errors on the transmitted qubits, i.e., in $\mathcal{G}_5$, which is generated by $\{\hat{Z}_1, \ldots, \hat{Z}_5, \hat{X}_1, \ldots, \hat{X}_5\}$. Thus, every error pattern is decomposed into these factors, which then indicate how it affects $|\hat{\psi}\rangle_L$. More precisely, if the occurred error, E, is 'corrected' by Bob with E', then it is sufficient to consider $E'E\,|\hat{\psi}\rangle_L$ by decomposing $E'E$ into factors of generators of $\mathcal{G}_5$. By looking at $|\hat{\psi}\rangle_L$, it can be concluded that the same mappings as in (8.10) hold for this EA code by instead considering (i_5, j_5) since $\hat{Z}_6, \hat{X}_6$ does not exists for this analysis as $\mathcal{G}_5$ is considered rather than $\mathcal{G}_6$. For example, if $E'E = \hat{X}_5$, then all terms with an α amplitude gain an $\hat{X}_5$, while the $\hat{X}_5$ factors are annihilated for the terms with a β factor. In other words, $\hat{X}_5$ acts as an logical bit flip, $\bar{X}$, on the code. Similarly, $\hat{Z}_5$ acts as $\bar{Z}$, and $\hat{X}_5\hat{Y}_5$ as $\bar{Y}$. Errors containing none of these factors are correctable. All error patterns of weight two or less are now analysed, and the results are summarised in Table 8.2.

Error of Weight Zero

Again, if the state has not been corrupted, then it is not corrected, hence $E'E = I$. This occurs with probability $(1 - p_T)^5(1 - p_S)^2$.

Errors of Weight One

The analysis of errors of weight one follow analogously as for the $[[6, 1, 3; 1]]$ EA code by modifying the probabilities appropriately. Explicitly, if any error occurs on one of the transmitted qubits, Bob detects and corrects it. Since there now only are 15 such error patterns, this occur with probability $5p_T(1-p_T)^4(1-p_S)^2$.

If one of Bob's ebits is corrupted, then he detects and corrects it if $p_1 > p_2$, but if $p_1 < p_2$, he wrongly 'corrects' it with an error of weight two in which case it $E'E \in \{\bar{Z}, \bar{X}, \bar{Y}\}$ depending on which type of error that occurs. As there now are two ebits that can be corrupted, the conclusion of this scenario is that if $p_1 > p_2$, then $E'E = I$ which occur with probability $2(1-p_T)^5 p_S(1-p_S)$, but if $p_1 < p_2$, then $E'E \in \{\bar{Z}, \bar{X}, \bar{Y}\}$ each with probability $(2(1-p_T)^5 p_S(1-p_S))/3$.

Errors of Weight Two

Since it in the previous was shown that the analysis holds analogously for all types of Pauli errors, the case of errors of weight two are separated into six groups depending on whether both errors are of the same type and where they occur; a) one type of Pauli error occur on transmitted qubits, b) one type occur on one transmitted and one stored qubit, c) one type corrupt both stored ebits, d) two types of errors corrupt transmitted qubits, e) two types occur on one transmitted and one stored qubit, and f) two types corrupt both the stored ebits. Since going from the $[[6,1,3;1]]$ EA code to the $[[5,1,3;2]]$ EA code essentially only corresponds to permuting the columns of the check matrix, much of the analysis is equivalent by re-numbering.

For errors in a), consider the case where two bit flips occur. They are detected by Bob as:

$$E = X_1X_2 \implies E' = X_5, \quad E = X_1X_3 \implies E' \in \{X_6, X_1X_3, X_2X_7, X_4X_5\},$$
$$E = X_1X_4 \implies E' \in \{X_7, X_1X_4, X_2X_6, X_3X_5\}, \quad E = X_2X_3 \implies E' = X_4,$$

where some errors have been omitted due to symmetry, e.g, $E = X_1X_5$ implies $E' = X_2$. Thus, if $E \in \{X_1X_2, X_1X_5, X_2X_5, X_2X_3, X_2X_4, X_3X_4\}$, Bob uniquely detects an error, however the wrong one. Due to the symmetry, it holds that in all cases, then $E'E \in \{X_1X_2X_5, X_2X_3X_4\} = \{\hat{Z}_5, \hat{Z}_4\hat{Z}_5\}$, hence $E'E$ acts as $\bar{Z}$ on $|\hat{\psi}\rangle_L$. The same reasoning holds if the errors were phase flips or bit phase flips. Thus, since there are six of such errors, $E'E \in \{\bar{Z}, \bar{X}, \bar{Y}\}$ each occurring with probability $(2p_T^2(1-p_T)^3(1-p_S))/3$.

On the other hand, of $E \in \{X_1X_3, X_1X_4, X_3X_5, X_4X_5\}$, then Bob detects E' depending on whether $p_1 > p_2$ or not. If $p_1 > p_2$, he wrongly detects a bit flip on one of his ebits, which once again implies that $E'E$ acts as $\bar{Z}$. If $p_1 < p_2$, he detects the correct error (at least one that is degenerate to the occurred error, hence it can be assumed that he detects the correct one without loss of generality), in which case $E'E = I$ naturally holds. Since the same reasoning holds for bit flips and bit-phase flips, the conclusion is that if $E \in \{A_1A_3, A_1A_4, A_3A_5, A_4A_5\}$ for $A \in \{X, Z, Y\}$, then $E'E = I$ given that $p_1 < p_2$, and $E'E \in \{\bar{Z}, \bar{X}, \bar{Z}\}$ if $p_1 > p_2$. These mappings occur with probability $(4p_T^2(1-p_T)^3(1-p_S))/3$ and $(4p_T^2(1-p_T)^3(1-p_S)^2)/9$, respectively. Notice that the last probability is for each of the mapping $\bar{Z}, \bar{X}, \bar{Y}$.

If one of the errors occur on one of Bob's ebits, then Bob will in almost all cases detect that one error has occurred. In fact, this will be the case for all errors unless the error on the transmitted qubit is on the second since this column in the check matrix is equal to the sum of the two columns corresponding to Bob's ebits. This implies that if the error is on the second transmitted qubit and one of Bob's ebits, then the lowest weight error yielding the same syndrome, is an error on Bob's other ebit. For example, if $E = X_2X_6$, then $E' \in \{X_7, X_2X_6, X_1X_4, X_3X_5\}$. But X_2X_6 is less likely to occur than the other errors due to $p_S < p_T$ by assumption, and X_1X_4 is degenerate with X_3X_5 as discussed previously. Hence, $E' \in \{X_7, X_3X_5\}$. If $p_1 > p_2$, $E'E = X_2X_6X_7$, which already has been argued to imply that $E'E = \bar{Z}$. If $p_1 < p_2$, then $E'E = X_2X_3X_5X_6 = \hat{X}_1X_6$, hence $E'E = I$. If the corrupted transmitted qubit is not the second, then he will always wrongly detect an error of weight one, in which case $E'E \in \{X_1X_3X_6, X_1X_4X_7, X_3X_5X_7, X_4X_5X_6\}$, which already has been argued to implicate that $E'E = \bar{Z}$. Since this holds similarly for the error types, it can be concluded that if $E \in \{A_2A_6, A_2A_7\}$ for $A \in \{X, Z, Y\}$, then $E'E = I$ given that $p_1 < p_2$, and $E'E \in \{\bar{Z}, \bar{X}, \bar{Y}\}$ if $p_1 > p_2$. These mappings occur with probability $(2p_T(1-p_T)^4 p_S(1-p_S))/3$ and $(2p_T(1-p_T)^4 p_S(1-p_S))/9$, respectively. If the errors falls in b), but not in the above case, then $E'E \in \{\bar{Z}, \bar{X}, \bar{Y}\}$, which each occur with probability $(8p_T(1-p_T)^4 p_S(1-p_S))/9$.

If both of Bob's ebits are corrupted by the same error type, the above discussion imply that $E'E$ acts as a logical Pauli operator. For example, if $E = X_6X_7$, then $E' = X_2$, which by above implicates that $E'E = \bar{Z}$. Hence, in this scenario, $E'E \in \{\bar{Z}, \bar{X}, \bar{Y}\}$ each occurring with probability $(1-p_T)^5 p_S^2/9$.

If instead two transmitted qubits are corrupted by different error types, then Bob cannot always uniquely determine the most likely error. To see this, it is by symmetry sufficient to consider two errors, namely $E \in \{X_1Z_2, X_1Z_3\}$. In the first case, $E' \in \{X_1Z_2, Y_2X_5, Y_1Z_5\}$, hence $E'E \in \{I, X_1X_2X_5, Z_1Z_2Z_5\} = \{I, \bar{X}, \bar{Z}\}$ such that the effect of the correction cannot be uniquely determined. In the other case, $E' \in \{X_1Z_3, Y_1Z_6, Y_3X_6\}$, which implies that $E' = X_1Z_3$ by the assumption that $p_S < p_T$. Thus, $E'E = I$ in that case. By generalising these results by symmetry, this second case occurs when $E \in \{A_1B_3, A_1B_4, A_3B_5, A_4B_5\}$ for $A, B \in \{X, Y, Z\}$ and $A \neq B$ such that the error types are different. Hence, $E'E = I$ with probability $(8p_T^2(1-p_T)^3(1-p_S))/3$. The first case occurs for all other error patterns on the form A_iB_k with $i, k = 1, \ldots, 5$ such that $i \neq k$. Thus, on average $E'E = I$ with probability $(4p_T^2(1-p_T)^3(1-p_S)^2)/3$, while $E'E \in \{\bar{Z}, \bar{X}, \bar{Y}\}$ each occur with probability $(8p_T^2(1-p_T)^3(1-p_S)^2)/9$.

If one of the errors then is on one of Bob's ebits, there are by symmetry again two cases to consider, which again is the case where the transmitted qubit that is corrupted is the second, and the rest. That is $E \in \{X_1Z_7, X_2Z_7\}$, whereafter the rest follow analogously. In the first case, $E' \in \{X_1Z_7, Y_1Z_4, X_4Y_7\}$, which by the assumption that $p_S < p_T$ implies that $E' = Y_1Z_4$. Then, $E'E = Z_1Z_4Z_7$, which already has been shown to implicate that $E'E = \bar{X}$. Since this holds in all cases where the second transmitted qubit is not corrupted, $E'E \in \{\bar{Z}, \bar{X}, \bar{Y}\}$ each occur with probability $(8p_T(1-p_T)^4p_S(1-p_S))/9$. In the other case, the only errors of weight two yielding the same syndrome are $E' \in \{X_2Z_7, Y_2Z_6, X_6Y_7\}$, whereof the last can be ruled out by $p_S < p_T$. However, as the remaining errors contain an error on Bob's ebit, it is more likely that three errors occur on the transmitted qubits if $p_1 < p_2$. Before considering this, if $p_1 > p_2$, then $E'E \in \{I, Z_2Z_6Z_7\} = \{I, \bar{X}\}$. Since this generalises to the other error types, it follows that on average, $E'E = I$ occurs with probability $(2p_T(1-p_T)^4p_S(1-p_S))/3$, while $E'E \in \{\bar{Z}, \bar{X}, \bar{Y}\}$ each with probability $(2p_T(1-p_T)^4p_S(1-p_S))/9$. If $p_1 < p_2$, then errors of weight three on the transmitted qubits are most likely, hence
$E' \in \{Y_1Z_4X_5, Z_1Y_2Z_3, Y_3X_4Z_5, Z_1X_3Y_4, X_1Z_3Y_5, Z_1X_2Z_3\}$, where some equivalent errors have been omitted. In all cases, $E'E \in \{\bar{X}, \bar{Y}\}$, which cannot be uniquely determined. As this generalises to all errors $E \in \{A_2B_6, A_2B_7\}$, on average $E'E \in \{\bar{Z}, \bar{X}, \bar{Y}\}$ with probability $(4p_T(1-p_T)^4p_S(1-p_S))/9$.

The last case to consider is when both Bob's ebits are corrupted by different types of error. This follows by similar reasoning to the case above where the second transmitted qubit was corrupted. For example, if $E = X_6Z_7$, then $E' \in \{X_6Z_7, Z_2Y_6, X_2Y_7\}$, whereof X_6Z_7 is less likely to occur than the other. This implies that $E'E\{X_2X_6X_7, Z_2Z_6Z_7\} = \{\hat{Z}_1\hat{Z}_2\hat{X}_5, \hat{Z}_4\hat{Z}_5\hat{X}_1\hat{X}_2\} = \{\bar{X}, \bar{Z}\}$, hence the effect cannot be uniquely determined. By symmetry, this also by swapping the error types on the ebits. In fact, the reasoning also holds for all error patterns by changing the logical Pauli operators appropriately. Hence, on average $E'E \in \{\bar{Z}, \bar{X}, \bar{Y}\}$ each with probability $(2(1-p_T)^5p_S^2)/9$. However, this case is only satisfied when $p_1 > p_2$, since if the inequality is swapped, it is more likely that the occurred error has been three errors on the transmitted qubits. This case is now considered, where it is sufficient to only examine the following three possible errors of weight three on the transmitted qubits that yield the same syndrome, as all the other valid errors are equivalent to one of these: $E' \in \{Y_1X_3Z_4, X_2Y_1Y_4, Z_2Y_1Y_3\}$. Then, $E'E \in \{X_1X_3X_6Z_1Z_4Z_7, X_2X_3X_5X_6Z_1Z_4Z_7, X_1X_3X_6Z_1Z_2Z_3Z_7\}$. Since $X_1X_3X_6 = \hat{Z}_5\hat{X}_1X_6, Z_1Z_4Z_7 = \hat{Z}_2\hat{Z}_3\hat{Z}_5Z_7, X_2X_3X_5X_6 = \hat{X}_1X_6$, and $Z_1Z_2Z_3Z_7 = \hat{Z}_2\hat{Z}_3Z_7$, it follows that $E'E \in \{\bar{Y}, \bar{X}, \bar{Z}\}$. Once again, the error cannot be uniquely determined, which imply that on average, $E'E \in \{\bar{Z}, \bar{X}, \bar{Y}\}$ each with probability $(2(1-p_T)^5p_S^2)/9$. Hence, although the reasoning is different for the cases where $p_1 > p_2$ and $p_1 < p_2$, the conclusion is in either case that, on average, $E'E \in \{\bar{Z}, \bar{X}, \bar{Y}\}$ each with probability $(2(1-p_T)^5p_S^2)/9$.

Approximate Probabilities for the Mapping

The results from the above analysis are summarised in Table 8.2, where the same notation is used as in Table 8.1 where the numbers has been fixed such that $i, j, k = 1, \ldots, 5$.

Based on the table, the approximate probabilities that the different mappings occur can be written

Error set	Mapping of $\vert\hat{\psi}\rangle_L$	Probability of error set	Condition for mapping
I	I	$(1-p_T)^5(1-p_S)^2$	None
A_i	I	$5p_T(1-p_T)^4(1-p_S)^2$	None
A_6, A_7	I	$2(1-p_T)^5 p_S(1-p_S)$	$p_1 > p_2$
A_6, A_7	$\bar{Z}, \bar{X}, \bar{Y}$	$\frac{2}{3}(1-p_T)^5 p_S(1-p_S)$	$p_1 < p_2$
$A_1A_3, A_1A_4, A_3A_5, A_4A_5$	$\bar{Z}, \bar{X}, \bar{Y}$	$\frac{4}{9}p_T^2(1-p_T)^3(1-p_S)^2$	$p_1 > p_2$
$A_1A_3, A_1A_4, A_3A_5, A_4A_5$	I	$\frac{4}{3}p_T^2(1-p_T)^3(1-p_S)^2$	$p_1 < p_2$
A_iA_j (not above)	$\bar{Z}, \bar{X}, \bar{Y}$	$\frac{2}{3}p_T^2(1-p_T)^3(1-p_S)^2$	None
A_2A_6, A_2A_7	$\bar{Z}, \bar{X}, \bar{Y}$	$\frac{2}{9}p_T(1-p_T)^4 p_S(1-p_S)$	$p_1 > p_2$
A_2A_6, A_2A_7	I	$\frac{2}{3}p_T(1-p_T)^4 p_S(1-p_S)$	$p_1 < p_2$
A_iA_6, A_iA_7 (not above)	$\bar{Z}, \bar{X}, \bar{Y}$	$\frac{8}{9}p_T(1-p_T)^4 p_S(1-p_S)$	None
A_6A_7	$\bar{Z}, \bar{X}, \bar{Y}$	$\frac{1}{9}(1-p_T)^5 p_S^2$	None
$A_1B_3, A_1B_4, A_3B_5, A_4B_5$	I	$\frac{8}{3}p_T^2(1-p_T)^3(1-p_S)^2$	None
A_iB_k (not above)	I	$\frac{4}{3}p_T^2(1-p_T)^3(1-p_S)^2$	Average
A_iB_k (not above)	$\bar{Z}, \bar{X}, \bar{Y}$	$\frac{8}{9}p_T^2(1-p_T)^3(1-p_S)^2$	Average
A_2B_6, A_2B_7	I	$\frac{2}{3}p_T(1-p_T)^4 p_S(1-p_S)$	$p_1 > p_2$, Average
A_2B_6, A_2B_7	$\bar{Z}, \bar{X}, \bar{Y}$	$\frac{2}{9}p_T(1-p_T)^4 p_S(1-p_S)$	$p_1 > p_2$, Average
A_2B_6, A_2B_7	$\bar{Z}, \bar{X}, \bar{Y}$	$\frac{4}{9}p_T(1-p_T)^4 p_S(1-p_S)$	$p_1 < p_2$, Average
A_iB_6, A_iB_7 (not above)	$\bar{Z}, \bar{X}, \bar{Y}$	$\frac{8}{9}p_T(1-p_T)^4 p_S(1-p_S)$	None
A_6B_7	$\bar{Z}, \bar{X}, \bar{Y}$	$\frac{2}{9}(1-p_T)^5 p_S^2$	Average

Table 8.2: Probabilities of how different error sets are decoding using the [[5, 1, 3; 2]] EA code.

explicitly:

$$\begin{aligned}
\Pr(E'E = I) &= (1-p_T)^3(1-p_S)^2\left[(1-p_T)^2 + 5p_T(1-p_T) + 4p_T^2\right] \\
&\quad + \chi_{[p_1>p_2]}\left[2(1-p_T)^4 p_S(1-p_S)((1-p_T) + \frac{1}{3}p_T)\right] \\
&\quad + \chi_{[p_1<p_2]}\left[\frac{2}{3}p_T(1-p_T)^3(1-p_S)(2p_T(1-p_S) + (1-p_T)p_S)\right], \\
\Pr\big(E'E = \bar{Z}\big) &= \Pr\big(E'E = \bar{X}\big) = \Pr\big(E'E = \bar{Y}\big) \\
&= \frac{1}{9}(1-p_T)^3\left[14p_T^2(1-p_S)^2 + 16p_T(1-p_T)p_S(1-p_S) + 3(1-p_T)^2 p_S^2\right] \\
&\quad + \chi_{[p_1>p_2]}\left[\frac{2}{9}p_T(1-p_T)^3(1-p_S)(2p_T(1-p_S) + 2(1-p_T)p_S)\right] \\
&\quad + \chi_{[p_1<p_2]}\left[\frac{2}{9}(1-p_T)^4 p_S(1-p_S)(3(1-p_T) + 2p_T)\right].
\end{aligned}$$

The conclusion regarding the above probabilities and error sets is essentially equivalent to that for the [[6, 1, 3; 1]] EA code. Firstly, the [[5, 1, 3; 2]] EA code can also only correct all errors of weight one if $p_1 > p_2$, in which case it has distance three in the classical sense. Secondly, some errors of weight two are always correctable, while some are dependent on the whether $p_1 > p_2$ or not. Out of the 189 possible errors of weight two, 24 are always correctly decoded, while another 18 are if $p_1 < p_2$, another 36 are a third of the time on average, and 12 additional ones are half of the time given that $p_1 > p_2$. Hence, 90 error patterns of weight two are correctable in theory, which is equivalent to the other code. However, this code has more patterns that are always correctly decoded, hence seemingly performs better in that regard. Lastly, errors on Bob's ebits are critical as they once again prevent correct decoding in most cases.

The probabilities for the different mappings are now visually illustrated in Figure 8.4, where $p_T = 0.25$ once again is fixed.

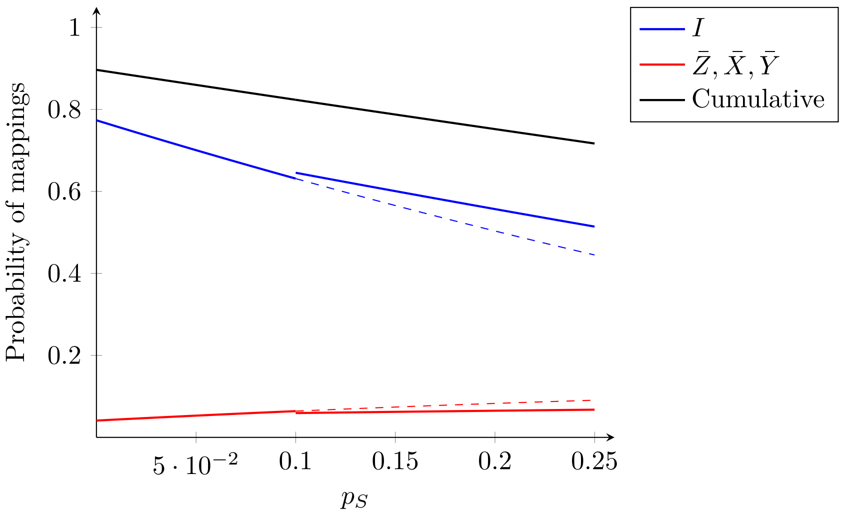


Figure 8.4: Curves of the probability for each of the mappings as a function of p_S together with the cumulative probability for these mappings for $p_T = 0.25$.

The same conclusions as for the $[[6,1,3;1]]$ EA code can be drawn. Particularly, for the $[[5,1,3;2]]$ EA code, the error can be corrected in almost 80% of the time when $p_T = 0.25$ and $p_S = 0$. However, as p_S increases, the performance quickly decreases, yet the error can be corrected over half the time for all $p_S \in [0, p_T)$. Again, the decrease is more extreme for low p_S. Another noteworthy remark is that the cumulative probability has increased to about 0.9 (as it theoretically should) instead of just above 0.8, which is a result of adding another ebit at the expense of a transmitted qubit such that fewer qubits are prone to the higher crossover probability p_T. The last remark to the figure is that the curves are discontinues at $p_S = 0.1$, which is where the inequality flips from $p_1 < p_2$ to $p_1 > p_2$. The reasoning for this is that if $p_1 > p_2$, then Bob correctly decodes an error of weight one on either of his ebits, however wrongly decodes an error of weight two of the same type on four qubit patterns, i.e., if $E \in \{A_1A_3, A_1A_4, A_3A_5, A_4A_5\}$. If $p_1 < p_2$, the opposite holds true. Thus, if $p_1 > p_2$, he can correctly decode six errors, which each occur with probability $((1-p_T)^5 p_S(1-p_S))/3$, while he wrongly decodes 12 errors that each occur with probability $(p_T^2(1-p_T)^3(1-p_S)^2)/9$. Hence, he is more likely to decode correctly if

$$2(1-p_T)^5 p_S(1-p_S) > \frac{4}{3}p_T^2(1-p_T)^3(1-p_S)^2 \implies (1-p_T)^2 p_S > \frac{2}{3}p_T^2(1-p_S)$$
$$\implies p_S > \frac{2p_T^2}{3(5p_T^2 - 2p_T + 1)}$$

Hence, this factor of 2/3 implies that by swapping decoding strategy at where the inequality swaps from $p_1 < p_2$ to $p_1 > p_2$ will yield a jump in the probability of decoding correctly. This in turn implies that Bob actually could obtain a better decoding performance by decoding as $p_1 > p_2$ even slightly before this holds true. More, precisely, by inserting $p_T = 0.25$ in the above formula, it follows that if he uses $p_1 > p_2$ as the rule for decoding when $p_S = 2/29$ or above, he obtains the optimal decoding performance in this regard. The idea that Bob can use this decoding approach is known to him as it is essentially is based upon how columns in the check matrix sum to those corresponding to his ebit. He does, however, not know the crossover probabilities for the channels in practice, which naturally implicates him not being able to precisely determine which rule to use.

Having analysed two different EA codes separately, they are now compared to analyse the effect of increasing the number of ebits needed in the EA coding framework.

8.5 Comparison of the Two Entanglement-Assisted Codes

As previously discussed, the $[[5,1,3;2]]$ EA code generally performs better than the $[[6,1,3;1]]$ EA code due to having one less qubit being transmitted but stored with lower error rate.

To see how much difference there are, the probabilities for the mappings with each code is plotted in Figure 8.5, where $p_T = 0.25$ once again has been used.

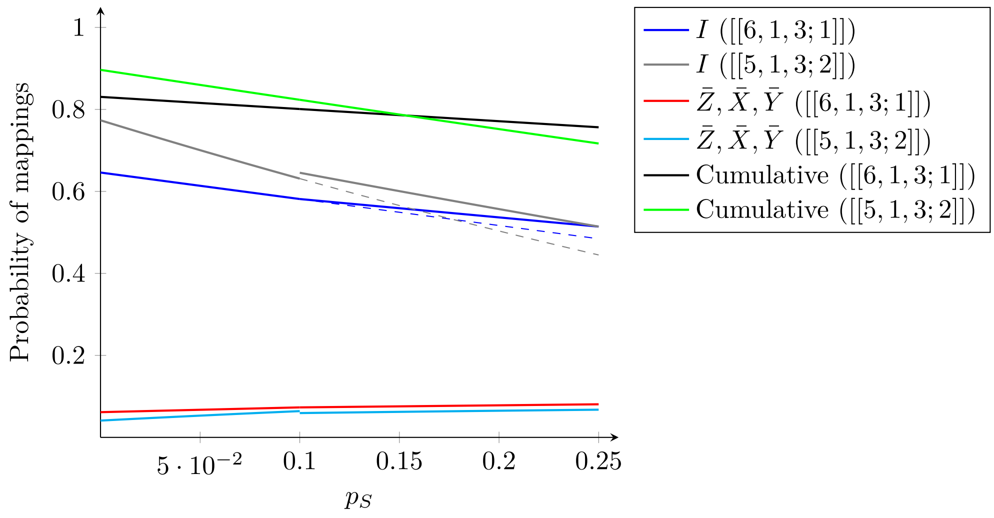


Figure 8.5: Curves of the probability for each of the mappings as a function of p_S together with the cumulative probability for these mappings for both the [[6, 1, 3; 1]] and the [[5, 1, 3; 2]] EA codes in the case of $p_T = 0.25$.

Four important conclusions can be drawn from the figure. Firstly, the [[5, 1, 3; 2]] EA code always has better performance in the sense that it is always at least as likely to decode correctly with the [[5, 1, 3; 2]] EA code compared to the [[6, 1, 3; 1]] EA code. Secondly, if $p_S = p_T$, then the probability of correctly decoding is identical for the codes, which is expected as this essentially corresponds to sending all of the seven qubits, i.e., including Bob's ebit(s), through the same channel. Hence, Bob does not achieve some additional error-correcting capability by having an ebit more when $p_S = p_T$. Thirdly, for large p_S, it can be seen that the [[6, 1, 3; 1]] EA code in some sense is more stable, meaning that even if Bob uses the wrong rule with respect to p_1 vs. p_2, then his performance does not degrade significantly (3%). If he uses the [[5, 1, 3; 2]] EA code, his performance is, nonetheless, degraded quite much (7%). Lastly, for all p_S, the probability of wrongly decoding is higher for the [[6, 1, 3; 1]] EA code, but it increases approximately at the same rate as for the [[5, 1, 3; 2]] EA code as p_S increases. Particularly, when $p_S = p_T$, then only errors of weight two may be non-correctable, hence the difference in the probabilities merely becomes a counting problem. For the [[6, 1, 3; 1]] EA code, there are 171 errors that are non-correctable, while there only are 153 for the [[5, 1, 3; 2]] EA code. Notice that both of these numbers include errors that only can be corrected on average.

To see whether these conclusion holds in general, the same plot, but with $p_T = 0.1$ is given in Figure 8.6. The exact same conclusions can be drawn from this plot, however, with other probabilities as the crossover probabilities have changed.

Now that the two EA codes have been compared, it is now considered what the impact of p_T is. That is, by fixing p_S, how does the performance of the EA codes compare?

Decoding Performance as a Function of Crossover Probability for the Transmission Channel

In order to resemble the previous plots as much as possible, it is chosen to fix $p_S = 0.1$ such that $p_1 > p_2$ when $p_T < 0.25$. The results are plotted in Figure 8.7.

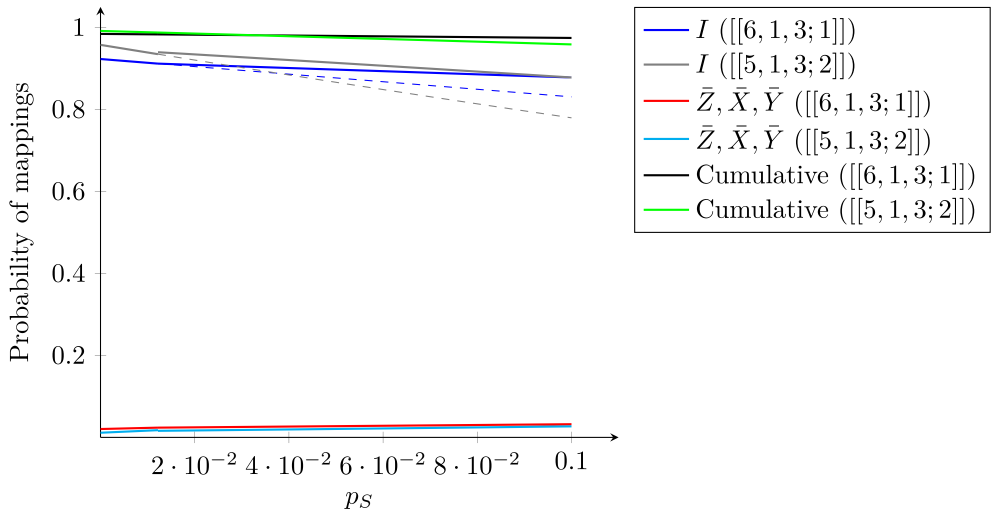


Figure 8.6: Curves of the probability for each of the mappings as a function of p_S together with the cumulative probability for these mappings for both the $[[6, 1, 3; 1]]$ and the $[[5, 1, 3; 2]]$ EA codes in the case of $p_T = 0.1$.

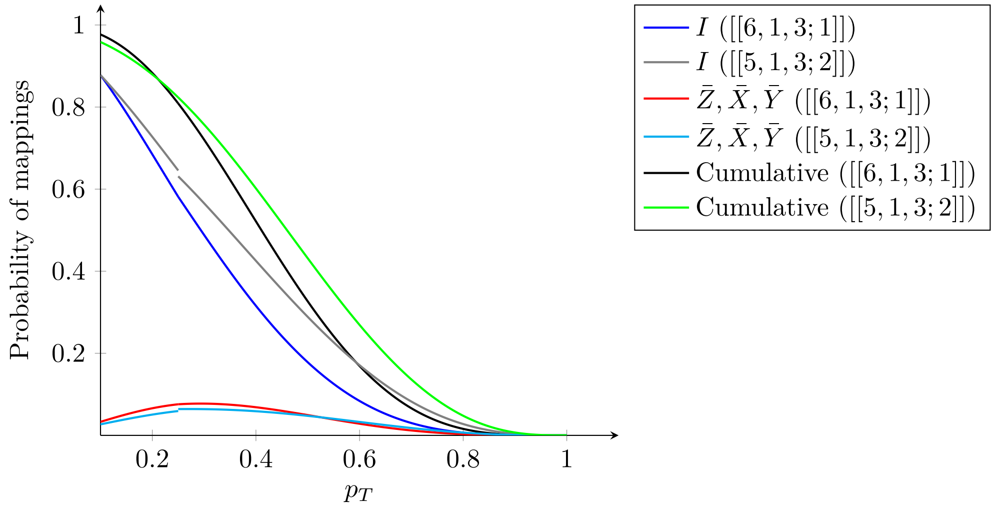


Figure 8.7: Curves of the probability for each of the mappings as a function of p_T together with the cumulative probability for these mappings for both the $[[6, 1, 3; 1]]$ and the $[[5, 1, 3; 2]]$ EA codes in the case of $p_S = 0.1$.

The plot is as expected; the performance of the $[[5, 1, 3; 2]]$ EA code is better then the $[[6, 1, 3; 1]]$ EA code, and both performances degrade as the crossover probability for the transmission channel increases.

The scenario where the EA codes are compared for a fixed p_S it suitable for the case where Alice and Bob knows the crossover probability on his storage channel and then wants to determine which EA code to use. The $[[6, 1, 3; 1]]$ EA code has lower performance, however, it requires one less EPR pair to be shared between Alice and Bob, hence there is a trade-off between these two. For simplicity, say that Alice and Bob are satisfied with being able to decode correctly with 80% of the time. If they use the $[[6, 1, 3; 1]]$ EA code, then $p_T < 0.1427$ must hold to obtain such performance, while it for the $[[5, 1, 3; 2]]$ EA code is $p_T < 0.1538$. To decide which of the EA codes to use, it must be considered whether it is more costly to require an additional ebit or to use a transmission channel with approximately one percentage point lower crossover probability. By lowering p_S, the gap between how small p_T must be in order to satisfy the given success threshold increases as illustrated in Figure 8.8, where $p_S = 0.02$ has been fixed.

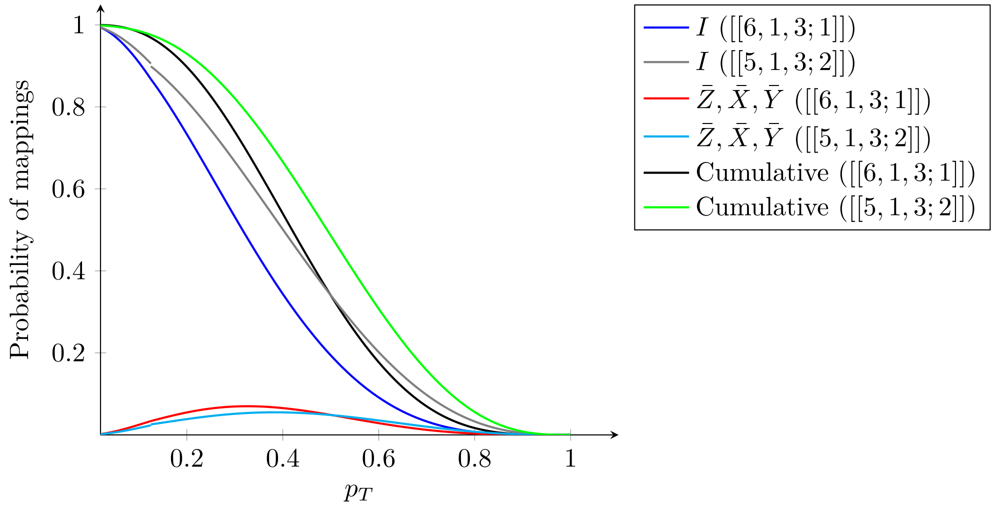


Figure 8.8: Curves of the probability for each of the mappings as a function of p_T together with the cumulative probability for these mappings for both the [[6, 1, 3; 1]] and the [[5, 1, 3; 2]] EA codes in the case of $p_S = 0.02$.

It is visually seen that the horizontal gap between the gray and the blue curves has increased for a probability of 0.8 compared to Figure 8.7. More precisely, for the [[6, 1, 3; 1]] EA code, $p_T < 0.1647$ must hold for the threshold to be satisfied, while it for the [[5, 1, 3; 2]] EA code is $p_T < 0.2105$.

The last analysis is to consider how noise on the ebit(s) affect the decoding performance of the EA codes. It has already been discussed that such errors are critical, however, it will now further examined.

Effect of Noisy Ebits on the Decoding Performance

From now on, only errors where an ebit has been corrupted is considered since any other error pattern has no impact on this analysis. Hence, for the [[6, 1, 3; 1]] EA code, the error patterns of interest are $\{A_7, A_iA_7, A_iB_7\}$. The probabilities for each of the mappings can in this case be simplified to

$$\Pr(E'E = I) = \chi_{[p_1>p_2]}\left[(1-p_T)^6 p_S\right],$$

$$\Pr\big(E'E = \bar{Z}\big) = \Pr\big(E'E = \bar{X}\big) = \Pr\big(E'E = \bar{Y}\big) = 2p_T(1-p_T)^5 p_S + \chi_{[p_1<p_2]}\left[\frac{1}{3}(1-p_T)^6 p_S\right].$$

Hence, it is only possible to correctly decode if $p_1 > p_2$, in which case the error on Bob's ebit can be corrected. In all other cases, Bob will wrongly 'correct' the error such that it is flipped by one of the Pauli errors. However, due to the syndrome measurements performed not leading to uncertainty in which error that has occurred, he will in fact know which of the flips it is. In other words, he knows the encoded state up to a Pauli flip. For example, if $E = X_7$, then, dependent on how p_1 relates to p_2, he either detects and corrects it, or he applies a logical bit flip operator to the encoded state. Strictly speaking, he does not know the state up to a bit flip as he naturally does not know that $E = X_7$ was in fact the occurred error and it could have been an error that he 'corrects' by applying another logical Pauli operator. However, if the crossover probabilities are sufficiently low such that such errors that must be of high-weight are unlikely, he knows the error up to a bit flip in some sense.

For the [[5, 1, 3; 2]] EA code, the error patterns of interest are $\{A_6, A_7, A_iA_6, A_iA_7, A_6A_7, A_iB_6, A_iB_7, A_6, A_6B_7\}$, where it should be explicitly noted that this include the special case where $i = 2$ in Table 8.2. Hence, by only considering these error patterns, the probabilities for the different mappings are simplified to

$$\Pr(E'E = I) = \chi_{[p_1>p_2]}\left[2(1-p_T)^4 p_S(1-p_S)((1-p_T) + \frac{1}{3}p_T)\right] + \chi_{[p_1<p_2]}\left[\frac{2}{3}p_T(1-p_T)^4 p_S(1-p_S)\right],$$

$$\Pr\big(E'E = \bar{Z}\big) = \Pr\big(E'E = \bar{X}\big) = \Pr\big(E'E = \bar{Y}\big) = \frac{1}{9}(1-p_T)^4 p_S\left[16p_T(1-p_S) + 3(1-p_T)p_S\right]$$
$$+ \chi_{[p_1>p_2]}\left[\frac{4}{9}p_T(1-p_T)^4 p_S(1-p_S)\right] + \chi_{[p_1<p_2]}\left[\frac{2}{9}(1-p_T)^4 p_S(1-p_S)(3(1-p_T) + 2p_T)\right].$$

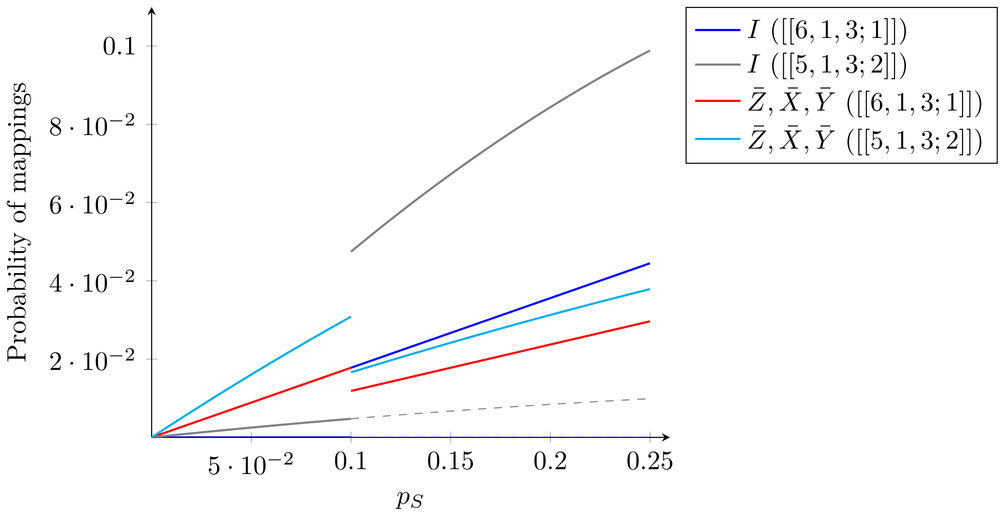


Figure 8.9: Curves of the probability for each of the mappings as a function of p_S for both the $[[6, 1, 3; 1]]$ and the $[[5, 1, 3; 2]]$ EA codes in the case of $p_T = 0.25$.

In this case, Bob will always detect some error patterns correctly, as one of the inequalities in the indicator functions always will be satisfied (the case where $p_1 = p_2$ is neglected, but even if it was the case, Bob would not uniquely determine an error, hence by randomly choosing a suitable error, he would decode correctly occasionally. Thus, the statement even holds in this case). However, there are cases where Bob cannot locate the error up to a single Pauli flip.. For example, if $E = X_6 Z_7$ and $p_1 < p_2$, then $E'E \in \{\bar{X}, \bar{Z}, \bar{Y}\}$ such that he cannot locate which logical Pauli operator that may have corrupted the encoded state.

All of the four probabilities above are now plotted in as a function of p_S for a fixed $p_T = 0.25$ in order to better understand how they compare. Notice that the cumulative probabilities have been omitted.

The figure clearly illustrates that it is significantly more likely to decode correctly with the $[[5, 1, 3; 2]]$ EA code when only considered errors with noisy ebits. However, as the $[[5, 1, 3; 2]]$ EA code has more ebits than the $[[6, 1, 3; 1]]$ EA code, there are more error patterns containing noisy ebits in the first case, which implies that the cumulative probability for the mappings simply is higher. Particularly, when $p_1 > p_2$, then all error patterns that only corrupt one ebit are correctable. Since the former code has twice as many of such error patterns, the probability of correctly decoding will simply be higher in terms of these errors. The caveat is that this increase in error patterns also imply that the probability of decoding wrongly is higher for this code.

The importance of the ebits being noiseless is now considered by decomposing the probability of correctly decoding into the cases where either zero, one, or two of the ebits have been corrupted for the two EA codes. That is, the error patterns are separated into whether zero, one, or two ebits have been corrupted. The results of doing so is plotted in Figure 8.10 and Figure 8.11 for the $[[6, 1, 3; 1]]$ and the $[[5, 1, 3; 2]]$ EA codes, respectively. It can from both plots be concluded that noisy ebits heavily decrease the performance of EA codes significantly. If p_S is small, then an error on an ebit is absolutely disastrous as the probability of decoding correctly then is zero (or almost zero for the $[[5, 1, 3; 2]]$ EA code). However, if p_S is such that $p_1 > p_2$, then it is possible to correct some errors, however fairly unlikely ones. More specifically, the noise on that particular ebit becomes correctable. Despite this increase, the performance of the case with no noisy ebits is still significantly better. It should be explicitly noted that if both ebits are corrupted in the case of the $[[5, 1, 3; 2]]$ EA code, then no error can be corrected, which also in indicated in Table 8.2.

Based on the performed analyses, it can be concluded that increasing the number of ebits in an EA code increase its error-correcting capability, at least under the assumptions used in this chapter, e.g., $p_S < p_T$. However, if an ebit is corrupted, the performance significantly reduces. Hence, the increase in decoding performance by adding more ebits comes at the risk of introducing errors on these ebits, which in turn decrease the performance. Since the potential decrease in performance due to noisy ebits is larger than the increase obtained by adding an extra ebit, the probability that the ebits are corrupted must be sufficiently low before such a risk is sensible.

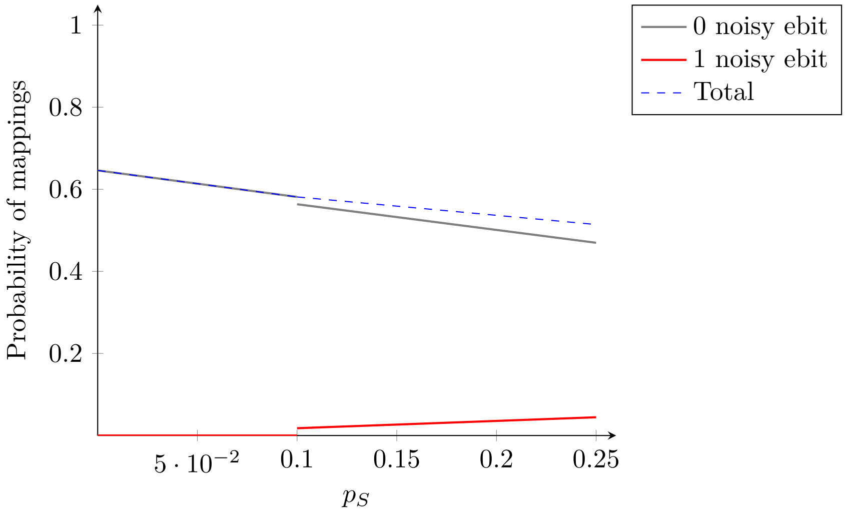


Figure 8.10: Curves of the probability of decoding correctly with the $[[6, 1, 3; 1]]$ EA code as a function of p_S in the case of $p_T = 0.25$. The probability have been decomposed into Bob's ebit is corrupted or not.

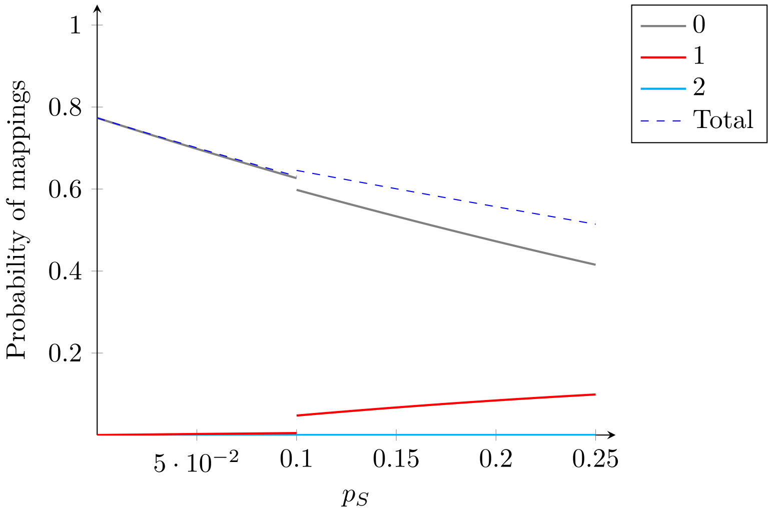


Figure 8.11: Curves of the probability of decoding correctly with the $[[5, 1, 3; 2]]$ EA code as a function of p_S in the case of $p_T = 0.25$. The probability have been decomposed into the cases where none, one, or two of Bob's ebits are corrupted

9 | Discussion

There are different aspects to the decoding performances analysed in Chapter 8 that deserves a remark.

Firstly, the analyses has been based upon EA codes obtained from the Steane code. As discussed in Section 7.2, such codes may not have optimal performance in the EA coding framework since they can correct errors on Bob's ebits under suitable conditions. This is clearly seen in the analyses, where Bob can correct errors on his ebit if $p_1 > p_2$. If different EA codes were used, this may then not be possible. However, in the EA framework where noisy ebits are allowed, it would be silly to use EA codes not capable of handling this. All in all, it could be interesting to consider other EA codes.

Secondly, it has throughout the analyses been assumed that Bob knows how to decode whenever the error syndrome describes a single most likely error. In practice, Bob does probably not know p_S nor p_T, hence he will not know when to swap between the decoding strategies. This could be accounted for by performing an analysis where he, e.g., knows the storage crossover probability to be Gaussian distributed with mean p_S and some variance τ.

Thirdly, it could be interesting to further analyse the idea of introducing an additional syndrome measurement in cases where he cannot determine whether the error is on one of his ebits or on some transmitted qubits. It would then be necessary to introduce some kind of penalty for performing this syndrome measurement, e.g., another channel on the ebits with some low crossover probability. In that case, the additional syndrome measurement comes at the cost of potentially introducing additional errors.

Fourthly, to make the scenario more realistic, it could be interesting to consider different error types, e.g., a decoherence channel on Bob's ebits.

Fifthly, another measure for performance could be used. The probability for correctly decoding is simple, however, it does not paint the full picture. Instead, it could be interesting to consider channel fidelities as a measure of performance.

Lastly, it could be interesting to take a more analytical approach by analysing how Knill-Laflamme conditions may be formulated in this formalism.

10 | Conclusion

The interest for quantum computers has exploded in the last decades as illustrated by the plethora of research activities and billion dollar investments in the field. By utilising quantum mechanical phenomena such as superposition and entanglement, quantum computers has the potential to efficiently solve problems that cannot be done efficiently with classical computers such as many-body problems and breaking the RSA encryption scheme. It is therefore believed that quantum computers will disrupt several industries such as pharmaceutical and finance. The physical realisation of quantum computers capable of causing such disruption has, however, turned out to be challenging. The most fatal problem is that quantum computers are extremely sensitive to noise, particularly decoherence which destroy they quantum mechanical properties that is harnessed in quantum computing. Hence, it is an absolute necessity to diminish the noise in quantum computers in order to enable them to reach their potential. One possible method to do so is quantum error-correction. In fact, it is believed that quantum error-correction is the ultimate enabler to achieve noiseless quantum computers.

The information of qubits used in quantum computing can, by utilising additional qubits in the quantum computer, be spread across many qubits by entangling the qubits. By doing so, quantum computers becomes less sensitive to noise since it is possible to correct some noise. Although a continuum of errors may occur, it is sufficiently to only correct bit flips, phase flips, and bit-phase flips. The caveat is, nonetheless, that a substantial amount of qubits is needed in order to be able to entangle them such that they can correct errors sufficiently well. Since obtaining large-scale quantum computers also is problematic, a quantum network is believed to be required to achieve the potential of quantum computers. Entanglement-assisted quantum error-correcting codes are a suitable candidate to perform quantum error-correcting over quantum networks since they require the existence of entanglement between the sender and the receiver similarly to many quantum communication protocols. The general framework for entanglement-assisted quantum error-correcting codes assumes that the receiver's part of the shared entanglement (the receiver's ebits) is noiseless, however, this is very difficult to realise in practice. The performance of entanglement-assisted quantum error-correcting codes when removing this assumption is therefore analysed.

A general approach to analyse how errors from the depolarising channel may corrupt the encoded information was firstly derived for the $[[6, 1, 3; 1]]$ entanglement-assisted quantum-error correcting code obtained from the Steane code. Based on this approach, it was found that it is still possible to correct some errors when the assumption of the receiver's ebit being noiseless is removed. In terms of how likely the correctable errors are, the code performs quite well when the probability of the receiver's ebit being corrupted is sufficiently low. However, when the receiver's ebit is corrupted, the decoding performance of the code decreases significantly. In fact, the probability of obtaining the correct state by decoding becomes nearly zero.

The same analysis was performed for the $[[5, 1, 3; 2]]$ entanglement-assisted quantum error-correcting code obtained from the Steane code. Although the $[[5, 1, 3; 2]]$ code generally has better performance than the $[[6, 1, 3; 1]]$ code, the decoding performance of the $[[5, 1, 3; 2]]$ code is nearly zero then at least one of the receiver's ebits is corrupted by noise. In fact, if both the receiver's ebits are corrupted, the probability of correctly decoding is zero.

Bibliography

[AAB+19] Frank Arute, Kunal Arya, Ryan Babbush, et al. Quantum supremacy using a programmable superconducting processor. Nature, 574:505–510, 2019.

[AAH+22] Adel Allahmadi, Ahmad AlKenani, Rola Hijazi, Najat Muthana, Ferruh Özbudak, and Patrick Solé. New constructions of entanglement-assisted quantum codes. Cryptography and Communications, 14(1):15–37, 2022.

[ABI+19] Andris Ambainis, Kaspars Balodis, Jānis Iraids, Martins Kokainis, Kriānis Krišjānis, and Jevgēnijs Vihrovs. Quantum speedups for exponential-time dynamic programming algorithms. In Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms, pages 1783–1793. SIAM, 2019.

[ACFW05] Sergio Albeverio, Laura Cattaneo, Shao-Ming Fei, and Xiao-Hong Wang. Equivalence of tripartite quantum states under local unitary transformations. International Journal of Quantum Information, Vol. 3, No. 4 (2005) 603-609, 2005.

[AGP08] Panos Aliferis, Daniel Gottesman, and John Preskill. Accuracy threshold for postselected quantum computation. Quantum Information & Computation, 8(3):181–244, 2008.

[AI21] Quantum AI. Quantum computer datasheet. https://quantumai.google/hardware/datasheet/weber.pdf, May 2021. Accessed: 20/05/2024.

[AI23] Team Google Quantum AI. Suppressing quantum errors by scaling a surface code logical qubit. Nature, 614(7949):676–681, 2023.

[App23] Apple. Apple introduces m2 ultra. https://www.apple.com/newsroom/2023/06/apple-introduces-m2-ultra/, June 2023. Accessed: 2023/10/05.

[Ato23] Quantum startup atom computing first to exceed 1,000 qubits. https://atom-computing.com/quantum-startup-atom-computing-first-to-exceed-1000-qubits/, October 2023. Accessed: 01/04/2024.

[BCG+24] Sergey Bravyi, Andrew. W. Cross, Jay M. Gambetta, Dmitri Maslov, Patrick Rall, and Theodore J. Yoder. High-threshold and low-overhead fault-tolerant quantum memory. Nature, 627(8005):778–782, 2024.

[BDH06] Todd Brun, Igor Devetak, and Min-Hsiu Hsieh. Correcting quantum errors with entanglement. Science, 314(5798):436–439, 2006.

[CAF+22] Marcello Caleffi, Michele Amoretti, Davide Ferrari, Danielle Cuomo, Jessica Illiano, Antonio Manzalini, and Angela Sara Cacciapuoti. Distributed quantum computing: a survey. arXiv preprint arXiv:2212.10609, 2022.

[CGK98] Isaac L. Chuang, Neil Gerschenfeld, and Mark Kubinec. Experimental implementation of fast quantum searching. Physical Review Letters, 80(15):3408, 1998.

[CRSS97] A. Robert Calderbank, Eric M. Rains, Peter W. Shor, and Neil J.A. Sloane. Quantum error correction and orthogonal geometry. Physical Review Letters, 78(3):405, 1997.

[CS96] A. Robert Calderbank and Peter W. Shor. Good quantum error-correcting codes exist. Physical Review A, 54(2):1098, 1996.

[Den19] Denmark makes decision to spend 1 billion dkk on quantum research and innovation strategy. https://investindk.com/insights/denmark-makes-decision-to-spend-1-billion-dkk-on-quantum-research-and-innovation-strategy, 2023/06/19. Accessed: 2024/03/18.

[Deu85] David Deutsch. Quantum theory, the church–turing principle and the universal quantum computer. Proceedings of the Royal Society of London. A. Mathematical and Physical Sciences, 400(1818):97–117, 1985.

[DF04] David S. Dummit and Richard M. Foote. Abstract Algebra. John Wiley & Sons, Inc., 3rd edition, 2004.

[DJ92] David Deutsch and Richard Jozsa. Rapid solution of problems by quantum computation. Proceedings of the Royal Society of London. Series A: Mathematical and Physical Sciences, 439(1907):553–558, 1992.

[DKLP02] Eric Dennis, Alexei Kitaev, Andrew Landahl, and John Preskill. Topological quantum memory. Journal of Mathematical Physics, 43(9):4452–4505, 2002.

[Fey65] Richard Feynman. The Character of Physical Law. MIT Press, 1965.

[Fey82] Richard Feynman. Simulating physics with computers. International Journal of Theoretical Physics, 21:467–488, 1982.

[FLS13] Richard Feynman, Robert Leighton, and Matthew Sands. The Feynman Lectures on Physics Volume III: New Millennium Edition. California Institute of Technology, 2013.

[FLZ$^+$22] Jihao Fan, Jun Li, Yongbin Zhou, Min-Hsie Hsieh, and H. Vincent Poor. Entanglement-assisted concatenated quantum codes. Proceedings of the National Academy of Sciences, 119(24):e2202235119, 2022.

[FXDC23] Ji-Hao Fan, Pei-Wen Xia, Di-Kang Dai, and Yi-Xiao Chen. Performance of entanglement-assisted quantum codes with noisy ebits over asymmetric and memory channels. Chinese Physics B, 32(12):120304, 2023.

[Gam23] Jay Gambetta. The hardware and software for the era of quantum utility is here. https://www.ibm.com/quantum/blog/quantum-roadmap-2033, December 2023. Accessed: 2024/04/01.

[GE21] Craig Gidney and Martin Ekerå. How to factor 2048 bit rsa integers in 8 hours using 20 million noisy qubits. Quantum, 5:433, 2021.

[GL23] James Grime and Alice Loxton. A history of the world in spy objects. james grime: Enigma machine. https://spyscape.com/podcast/james-grime-enigma-machine, December 2023. Accessed: 2024/03/18.

[Got96] Daniel Gottesman. Class of quantum error-correcting codes saturating the quantum hamming bound. Physical Review A, 54(3):1862, 1996.

[Got97] Daniel Gottesman. Stabilizer codes and quantum error correction. https://arxiv.org/abs/quant-ph/9705052, 1997.

[Gro96] Lov K. Grover. A fast quantum mechanical algorithm for database search. In Proceedings of the twenty-eighth annual ACM symposium on Theory of computing, pages 212–219, 1996.

[GZT$^+$20] Aurore Guillevic, Paul Zimmerman, Emmanuel Thomé, Pierrick Gaudry, Nadia Heninger, and Fabrice Boudot. New record set for cryptographic challenge. https://cse.ucsd.edu/about/news/new-record-set-cryptographic-challenge, March 2020. Accessed: 2024/04/01.

[Hit73] William C. Hittinger. Metal-oxide-semiconductor technology. Scientific American, 229(2):48–59, 1973.

[HK62] Michael Held and Richard M. Karp. A dynamic programming approach to sequencing problems. Journal of the Society for Industrial and Applied mathematics, 10(1):196–210, 1962.

[HLV10] John P. Holdren, Eric Lander, and Harold Varmus. Designing a digital future: Federally funded reseach and development in networking and information technology. https://www.nitrd.gov/pubs/PCAST-NITRD-report-2010.pdf, December 2010. Accessed: 2024/03/21.

[Ifr01] Georges Ifrah. The Universal History of Computing: From the Abacus to the Quantum Computer. John Wiley & Sons, Inc., 2001.

[Kas19] Michael Kastoryano. Lecture notes in quantum error correction. https://www.thp.uni-koeln.de/kastoryano/ExSheets/Notes_v5.pdf, 2019. Accessed: 2024/04/01.

[Kha16] Mukesh Khare. How to squeeze billions of transistors onto a computer chip. https://www.ibm.com/thought-leadership/innovation-explanations/mukesh-khare-on-smaller-transistors-analytics, 2016. Accessed: 2023/10/05.

[Kit03] Alexei Kitaev. Fault-tolerant quantum computation by anyons. Annals of physics, 303(1):2–30, 2003.

[KLZ98] Emanuel Knill, Raymond Laflamme, and Wojciech H Zurek. Resilient quantum computation. Science, 279(5349):342–345, 1998.

[KN98] Tadashi Kadowaki and Hidetoshi Nishimori. Quantum annealing in the transverse ising model. Physical Review E, 58(5):5355, 1998.

[KSB^{+}20] Morten Kjaergaard, Mollie E. Schwartz, Jochen Braumüller, Philip Krantz, Joel I-J Wang, Simon Gustavsson, and William D. Oliver. Superconducting qubits: Current state of play. Annual Review of Condensed Matter Physics, 11:369–395, 2020.

[L^{+}20] Charles E. Leiserson et al. There's plenty of room at the top: What will drive computer performance after moore's law? Science, 368, 6495, June 2020.

[LB12] Ching-Yi Lai and Todd A. Brun. Entanglement-assisted quantum error-correcting codes with imperfect ebits. Physical Review A, 86(3):032319, 2012.

[McD22] Mickey McDonald. Establishing world-record coherence times on nuclear spin qubits made from neutral atoms. https://atom-computing.com/establishing-world-record-coherence-times-on-nuclear-spin-qubits-made-from-neutral-atoms/, May 2022. Accessed: 20/05/2024.

[MLLL^{+}12] Enrique Martin-Lopez, Anthony Laing, Thomas Lawson, Roberto Alvarez, Ziao-Qi Zhou, and Jeremy L. O'brien. Experimental realization of shor's quantum factoring algorithm using qubit recycling. Nature photonics, 6(11):773–776, 2012.

[Moo65] Gordon E. Moore. Cramming more components onto integrated circuits. Electronics, 38(8), April 1965.

[MZ^{+}23] Mateusz Masiowski, Matija Zesko, et al. Quantum technology monitor. https://www.google.com/url?sa=t&source=web&rct=j&opi=89978449&url=https://www.mckinsey.com/~/media/mckinsey/business%2520functions/mckinsey%2520digital/our%2520insights/quantum%2520technology%2520sees%2520record%2520investments%2520progress%2520on%2520talent%2520gap/quantum-technology-monitor-april-2023.pdf&ved=2ahUKEwi_wvutzP2EAxVczgIHHUIXBmUQFnoECB4QAQ&usg=AOvVaw2Aku-N0xWaYSrSI6s5-o2I, April 2023.

[NC10] Michael A. Nielsen and Isaac L. Chuang. Quantum Computation and Quantum Information: 10th Anniversary Edition. Cambridge University Press, 2010.

[NH^{+}19] Soon Xin Ng, Lajos Hanzo, et al. Duality of quantum and classical error correction codes: Design principles and examples. IEEE Communications Surveys & Tutorials, 21(1):970–1010, 2019.

[OW^{+}22] Atila Orhon, Aseem Wadhws, et al. Deploying transformers on the apple neural engine. https://machinelearning.apple.com/research/neural-engine-transformers, June 2022. Accessed: 2024/03/21.

[Pla] IBM Quantum Platform. Systems: Ibm-kyiv. https://quantum.ibm.com/services/resources?system=ibm_kyiv. Accessed: 20/05/2024.

[Pre23] John Preskill. Quantum computing 40 years later. In Feynman Lectures on Computation, pages 193–244. CRC Press, 2023.

[PZ03] John Proos and Christof Zalka. Shor's discrete logarithm quantum algorithm for elliptic curves. Quantum Information & Computation, 3(4):317–344, 2003.

[Rig] Rigetti. Rigetti systems: Ankaa-2 quantum processor. https://qcs.rigetti.com/qpus. Accessed: 20/05/2024.

[RXY+17] Ji-Gang Ren, Ping Xu, Hai-Lin Yong, Liang Zhang, Sheng-Kai Liao, Juan Yin, Wei-Yue Liu, Wen-Qi Cai, Meng Yang, Li Li, et al. Ground-to-satellite quantum teleportation. Nature, 549(7670):70–73, 2017.

[Sci] Babbage's difference engine no 2, 2022. https://collection.sciencemuseumgroup.org.uk/people/cp818. Accessed: 2024/03/18.

[Sco] https://www.scopus.com/results/results.uri?sort=cp-f&src=s&st1=%22Quantum+technology%22&sid=d9379a710508320179bcdc0bdc01cc74&sot=b&sdt=b&sl=35&s=TITLE-ABS-KEY%28%22quantum+technology%22+OR+%22quantum+computing%22+or+%22quantum+computer%22%29&origin=searchbasic&editSaveSearch=&yearFrom=Before+1960&yearTo=Present&sessionSearchId=d9379a710508320179bcdc0bdc01cc74&limit=10. Accessed: 2024/06/02.

[Sha48] Claude Elwood Shannon. A mathematical theory of communication. The Bell system technical journal, 27(3):379–423, 1948.

[Sho94] Peter W. Shor. Algorithms for quantum computation: discrete logarithms and factoring. In Proceedings 35th annual symposium on foundations of computer science, pages 124–134. Ieee, 1994.

[Sho95] Peter W. Shor. Scheme for reducing decoherence in quantum computer memory. Physical review A, 52(4):R2493, 1995.

[Sho96] Peter W. Shor. Fault-tolerant quantum computation. In Proceedings of 37th conference on foundations of computer science, pages 56–65. IEEE, 1996.

[Sho99] Peter W. Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. SIAM review, 41(2):303–332, 1999.

[Sta24] IonQ Staff. Ionq aria: Practical performances. https://ionq.com/resources/ionq-aria-practical-performance, January 2024. Accessed: 20/05/2024.

[Ste96] Andrew Steane. Multiple-particle interference and quantum error correction. Proceedings of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences, 452(1954):2551–2577, 1996.

[Top23] Highlights - november 2023. https://www.top500.org/lists/top500/2023/11/highs/, November 2023. Accessed: 2024/03/21.

[Tur36] Alan M. Turing. On computable numbers, with an application to the Entscheidungsproblem. Proceedings of the London Mathematical Society, 2(42):230–265, 1936.

[WB08] Mark M. Wilde and Todd A. Brun. Optimal entanglement formulas for entanglement-assisted quantum coding. Physical Review A, 77(6):064302, 2008.

[WKB10] Mark M. Wilde, Hari Krovi, and Todd A. Brun. Convolutional entanglement distillation. In 2010 IEEE International Symposium on Information Theory, pages 2657–2661. IEEE, 2010.

[YBL+10] Zhen-Sheng Yuan, Xiao-Hui Bao, Chao-Yang Lu, Jun Zhang, Cheng-Zhi Peng, and Jian-Wei Pan. Entangled photons and quantum communication. Physics Reports, 497(1):1–40, 2010.

Appendices

A | Deutsch-Jozsa Algorithm

The Deutsch-Jozsa algorithm is presented in this appendix. It is a simple and deterministic quantum algorithm, which albeit its little practical use is important due to being one of the first quantum algorithms to be exponentially faster than its deterministic classical counterparts. It is based on [NC10, ch. 1.4.4].

Consider a function taking on an n-bit string and outputs a bit, i.e., $f : \{0, 1\}^n \to \{0, 1\}$. Assume that f is either constant (all inputs yield either 0 or 1) or balanced (half of the inputs yield 0, the other half yield 1). The problem is then to determine whether f is constant or balanced by using a black-box implementation of f, typically known as an oracle.

Before introducing the Deutsch-Jozsa algorithm, the complexity of a deterministic classical algorithm for this problem is considered. The case where the fewest evaluations of f is needed is when it is balanced and the first two evaluations are distinct, which clearly indicates that f is balanced. However, the first 2^{n-1} evaluations may be identical, in which case no information about f can be determined with certainty. The next evaluation will nonetheless either be the same or distinct from the previous evaluations, indicating that f is constant or balanced, respectively. That is, $2^{n-1} + 1$ evaluations are required in the the worst case, meaning that the complexity of the algorithm is exponential. As a side note, it should be noted that fewer evaluations can determine the property of f with high probability by using a probabilistic algorithm. However, this is not elaborated further upon as the Deutsch-Jozsa algorithm is deterministic, hence it is only compared to a deterministic algorithm.

The Deutsch-Jozsa algorithm can now be presented. Notice that the states are assumed to be perfect in the sense that decoherence is not present. Furthermore, the oracle is defined as the operator U_f satisfying $(|x\rangle \otimes |y\rangle) \mapsto (|x\rangle \otimes |y \oplus f(x)\rangle)$, where $\oplus$ is addition modulo 2. The first system of the tensor product therefore corresponds to inputs of f, while the second holds information about the output. Since there are 2^n possible inputs, n qubits is required to describe all of the inputs, while a single qubit can describe the output. In the Deutsch-Jozsa algorithm, the input is initialised in the $\{0\}^n$, while the output is initialised in 1, yielding the quantum state

$$|\psi_0\rangle = |0\rangle^{\otimes n} \otimes |1\rangle .$$

In order to consider all the possible inputs and outputs, the $(n + 1)$-fold Hadamard gate is applied, which gives the states

$$\begin{aligned}|\psi_1\rangle = H^{\otimes(n+1)} |\psi_0\rangle = |+\rangle^{\otimes n} \otimes |-\rangle &= \left(\frac{1}{\sqrt{2^n}} \sum_{x\in\{0,1\}^n} |x\rangle\right) \otimes \left(\frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)\right) \\ &= \frac{1}{\sqrt{2^{n+1}}} \sum_{x\in\{0,1\}^n} |x\rangle \otimes (|0\rangle - |1\rangle).\end{aligned}$$

The first system is now in an even superposition of all inputs of f, while the second is of the two outputs of f. The oracle can therefore be applied now to evaluate f at every input simultaneously, i.e., quantum parallelism. This yields the state

$$\begin{aligned}|\psi_2\rangle = U_f |\psi_1\rangle &= \frac{1}{\sqrt{2^{n+1}}} \sum_{x\in\{0,1\}^n} |x\rangle \otimes (|0 \oplus f(x)\rangle - |1 \oplus f(x)\rangle) \\ &= \frac{1}{\sqrt{2^{n+1}}} \sum_{x\in\{0,1\}^n} |x\rangle \otimes ((-1)^{f(x)}(|0\rangle - |1\rangle)) = \frac{1}{\sqrt{2^{n+1}}} \sum_{x\in\{0,1\}^n} (-1)^{f(x)} |x\rangle \otimes (|0\rangle - |1\rangle).\end{aligned}$$

At this point, the information regarding f is incorporated into the sign of the amplitudes of the inputs of f, i.e., the first system containing $|x\rangle$. Since the second system $(\frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$ is independent of x, it contains no information about f, and can therefore be disregarded. The state can thus be simplified to

$$|\psi_3\rangle = \frac{1}{\sqrt{2^n}} \sum_{x\in\{0,1\}^n} (-1)^{f(x)} |x\rangle .$$

Performing a measurement at this point will not provide useful information about f. It is therefore necessary to implement a step such that states interfere with each other such that the property of f can be deduced. This is done by applying the n-fold Hadamard gate, which yields the state

$$\begin{aligned}
|\psi_4\rangle = H^{\otimes n}|\psi_3\rangle &= \frac{1}{\sqrt{2^n}} \sum_{x\in\{0,1\}^n} (-1)^{f(x)} H^{\otimes n}|x\rangle \\
&= \frac{1}{\sqrt{2^n}} \sum_{x\in\{0,1\}^n} (-1)^{f(x)} \left(\frac{1}{\sqrt{2}}(|0\rangle + (-1)^{x_1}|1\rangle)\right) \otimes \cdots \otimes \left(\frac{1}{\sqrt{2}}(|0\rangle + (-1)^{x_n}|1\rangle)\right) \\
&\overset{(a)}{=} \frac{1}{\sqrt{2^n}} \sum_{x\in\{0,1\}^n} (-1)^{f(x)} \left(\frac{1}{\sqrt{2^n}} \sum_{z\in\{0,1\}^n} (-1)^{z^\top x}|z\rangle\right) \\
&= \sum_{z\in\{0,1\}^n} \left(\frac{1}{2^n} \sum_{x\in\{0,1\}^n} (-1)^{f(x)+z^\top x}\right)|z\rangle,
\end{aligned}$$

where (a) follows by letting $z^\top x$ be the bit-wise inner product with addition modulo 2. Thus, for any $z \in \{0,1\}^n$, the state $|z\rangle$ has amplitude

$$\alpha_z = \frac{1}{2^n} \sum_{x\in\{0,1\}^n} (-1)^{f(x)+z^\top x},$$

which implies that measuring the system yields state $|z\rangle$ with probability $|\alpha_z|^2$. Particularly, the state $|0\rangle^{\otimes n}$ has amplitude

$$\alpha_0 = \frac{1}{2^n} \sum_{x\in\{0,1\}^n} (-1)^{f(x)} = \begin{cases} 1, & \text{if } f \text{ is constant } 0, \\ 0, & \text{if } f \text{ is balanced,} \\ -1, & \text{if } f \text{ is constant } 1 \end{cases} \implies |\alpha_0|^2 = \begin{cases} 1, & \text{if } f \text{ is constant,} \\ 0, & \text{if } f \text{ is balanced.} \end{cases}$$

Thus, $|0\rangle^{\otimes n}$ is measured with certainty if f is constant, while it cannot be measured if f is balanced. In other words, if $|0\rangle^{\otimes n}$ is measured, it can be deduced that f is constant, while it can be deduced that f is balanced if any other state is measured. Therefore, a single evaluation of U_f is enough to determine whether f is constant or balanced, which is an exponential speed-up compared to the classical algorithm as promised. It should however be noted that the evaluations in the classical and quantum algorithms are different, thus a comparison of only these parameters does not provide the full picture. It does nonetheless indicate the power of quantum computing.

B | Miscellaneous Linear Algebra

In this appendix is some miscellaneous linear algebra used throughout the thesis introduced. The concepts follow the order in which they appear in the thesis.

B.1 Kronecker Product

This section is based upon [DF04, pp. 420-422].

For clarification and completeness, the Kronecker product is defined.

Definition B.1: Kronecker Product
Let $A \in \operatorname{Hom}(\mathbb{C}^m, \mathbb{C}^n)$ and $B \in \operatorname{Hom}(\mathbb{C}^p, \mathbb{C}^q)$ be given. The Kronecker product $(A \otimes B) \in \operatorname{Hom}(\mathbb{C}^{mp}, \mathbb{C}^{nq})$ is defined as

$$A \otimes B = \begin{bmatrix} a_{11}B & \cdots & a_{1n}B \\ \vdots & \ddots & \vdots \\ a_{m1}B & \cdots & a_{mn}B \end{bmatrix}.$$

[DF04, p. 421]

From the definition, it is clear that the general expression for the i, j-th term of $A \otimes B$ has the form

$$(A \otimes B)_{i,j} = a_{x,y} b_{v,w}$$

for appropriate x, y, v, w. However, since $A \otimes B$ is a block matrix, where each block is a scaled B, x, y, v, w can be expressed in terms of the dimension of B. First, x can be rewritten by realising that the first p elements in the j-th column of $A \otimes B$ all has the coefficient $a_{1,j}$, the next p elements has $a_{2,j}$, etc. This implies that $x = \lceil i/p \rceil$, where $\lceil \cdot \rceil$ is the ceiling function. Similar reasoning for the rows yields $y = \lceil j/q \rceil$. Now, rewriting v can be done by seeing that all rows on the form $kp + 1$ for $k = 0, 1, \ldots, m - 1$ has the same b coefficients. Since element numerations starts at 1, this is equivalent to saying that $v = (i - 1) \pmod p + 1$. Analogously, for columns it follows that $w = (j - 1) \pmod p + 1$. Thus, giving the dimensions of B, one can determine the i, j-th element as

$$(A \otimes B)_{i,j} = a_{\lceil i/p \rceil, \lceil j/q \rceil} b_{(i-1) \pmod p + 1, (j-1) \pmod p + 1}.$$

Even though this expression is more convoluted than the definition, it can be useful if a certain element in the Kronecker product is needed. It is often only needed to use the definition, e.g., when proving properties of the Kronecker product. Some of the simple yet desirable properties are stated below.

Theorem B.2: Basic Properties of Kronecker Product
Let $A \in \operatorname{Hom}(\mathbb{C}^m, \mathbb{C}^n), B \in \operatorname{Hom}(\mathbb{C}^p, \mathbb{C}^q)$, and $C, D \in \operatorname{Hom}(\mathbb{C}^r, \mathbb{C}^s)$. The Kronecker product then satisfies the following properties:

i) Associativity:

$$(A \otimes B) \otimes C = A \otimes (B \otimes C).$$

ii) Distributivity over addition if $\dim(A) = \dim(B)$:

$$(A + B) \otimes (C + D) = A \otimes C + A \otimes D + B \otimes C + B \otimes D.$$

iii) Complex conjugating respect the product:

$$(A \otimes B)^\dagger = A^\dagger \otimes B^\dagger.$$

[NC10, p. 74]

Proof
To show that it is associative, the Kronecker product is explicit written,

$$\begin{aligned}(A \otimes B) \otimes C &= \begin{bmatrix} a_{11}B & \cdots & a_{1n}B \\ \vdots & \ddots & \vdots \\ a_{m1}B & \cdots & a_{mn}B \end{bmatrix} \otimes C = \begin{bmatrix} (a_{11}B) \otimes C & \cdots & (a_{1n}B) \otimes C \\ \vdots & \ddots & \vdots \\ (a_{m1}B) \otimes C & \cdots & (a_{mn}B) \otimes C \end{bmatrix} \\ &= \begin{bmatrix} a_{11}(B \otimes C) & \cdots & a_{1n}(B \otimes C) \\ \vdots & \ddots & \vdots \\ a_{m1}(B \otimes C) & \cdots & a_{mn}(B \otimes C) \end{bmatrix} = A \otimes (B \otimes C).\end{aligned}$$

To show that it is distributive, consider the i, j-th block matrix in $(A + B) \otimes (C + D)$, which is given as

$$\begin{aligned}\Big((A + B) \otimes (C + D)\Big)_{ij} &= (A + B)_{ij}(C + D) = a_{ij}C + a_{ij}D + b_{ij}C + b_{ij}D \\ &= (A \otimes C)_{ij} + (A \otimes D)_{ij} + (B \otimes C)_{ij} + (B \otimes D)_{ij}.\end{aligned}$$

Since this holds for every entry in $(A+B) \otimes (C + D)$, the Kronecker product is distributive over addition.

To show that complex conjugation respects the Kronecker product, it is once again written out explicitly;

$$(A \otimes B)^\dagger = \begin{bmatrix} a_{11}B & \cdots & a_{1n}B \\ \vdots & \ddots & \vdots \\ a_{m1}B & \cdots & a_{mn}B \end{bmatrix}^\dagger = \begin{bmatrix} \overline{a}_{11}B^\dagger & \cdots & \overline{a}_{m1}B^\dagger \\ \vdots & \ddots & \vdots \\ \overline{a}_{1n}B^\dagger & \cdots & \overline{a}_{mn}B^\dagger \end{bmatrix} = A^\dagger \otimes B^\dagger,$$

which concludes the proof. ■

Another important property of the Kronecker product is that it can be mixed with regular products.

Theorem B.3: Mixed-product Property
Let $A \in \mathrm{Hom}(\mathbb{C}^m, \mathbb{C}^n), B \in \mathrm{Hom}(\mathbb{C}^p, \mathbb{C}^q), C \in \mathrm{Hom}(\mathbb{C}^n, \mathbb{C}^k)$, and $D \in \mathrm{Hom}(\mathbb{C}^q, \mathbb{C}^r)$. Then, AC and BD are well-defined and

$$(A \otimes B)(C \otimes D) = (AC) \otimes (BD).$$

Proof
Since multiplication of block matrices acts similarly as regular matrix multiplication, it follows that

$$\begin{aligned}(A \otimes B)(C \otimes D) &= \begin{bmatrix} a_{11}B & \cdots & a_{1n}B \\ \vdots & \ddots & \vdots \\ a_{m1}B & \cdots & a_{mn}B \end{bmatrix} \begin{bmatrix} c_{11}D & \cdots & c_{1k}D \\ \vdots & \ddots & \vdots \\ a_{n1}D & \cdots & a_{nk}D \end{bmatrix} \\ &= \begin{bmatrix} \sum_{i=1}^n a_{1i}c_{i1}BD & \cdots & \sum_{i=1}^n a_{1i}c_{ik}BD \\ \vdots & \ddots & \vdots \\ \sum_{i=1}^n a_{mi}c_{i1}BD & \cdots & \sum_{i=1}^n a_{mi}c_{ik}BD \end{bmatrix} \\ &= \begin{bmatrix} (AC)_{11}BD & \cdots & (AC)_{1k}BD \\ \vdots & \ddots & \vdots \\ (AC)_{m1}BD & \cdots & (AC)_{mk}BD \end{bmatrix} \\ &= (AC) \otimes (BD).\end{aligned}$$

■

As a direct consequence of the mixed-product property, it follows that if C and D are identity matrix of appropriate size, I_C and I_D, respectively, then

$$(A \otimes I_C)(I_D \otimes B) = A \otimes B = (I_C \otimes B)(A \otimes I_D).$$

This yields some useful tricks, where terms including Kronecker products can be rewritten by multiplying by the identity matrix, or scalar 1 if it is on a vector, appropriately. As an example, let $x \in \mathbb{C}^n, y \in \mathbb{C}^m$. Then

$$x(x^\top \otimes y) = (x \otimes 1)(x^\top \otimes y) = (xx^\top) \otimes y,$$

which may be a more useful expressions if knowledge about $xx^\top$ is given. The mixed-product property generalises naturally.

Corollary B.4: Generalisation of the Mixed-product Property
Let linear transformations $A_1, \dots, A_k, B_1, \dots, B_k$ be given such that $A_i \in \operatorname{Hom}(\mathbb{C}^{n_{i-1}}, \mathbb{C}^{n_i}), B \in \operatorname{Hom}(\mathbb{C}^{m_{i-1}}, \mathbb{C}^{m_i})$ for $i = 1, \dots, k$ and $n_0, \dots, n_k, m_0, \dots, m_k \in \mathbb{N}$. Then

$$\prod_{i=1}^{k}(A_i \otimes B_i) = \left(\prod_{i=1}^{k} A_i\right) \otimes \left(\prod_{i=1}^{k} B_i\right).$$

Proof
The generalisation is proven by induction. The base step is shown in Theorem B.3, so only the update step is needed. Thus, assume that the statement holds for k. For $k+1$, it then follows that

$$\begin{aligned}\prod_{i=1}^{k+1}(A_i \otimes B_i) &= \prod_{i=1}^{k}(A_i \otimes B_i)(A_{k+1} \otimes B_{k+1}) \\ &= \left[\left(\prod_{i=1}^{k} A_i\right) \otimes \left(\prod_{i=1}^{k} B_i\right)\right](A_{k+1} \otimes B_{k+1}) \\ &\overset{(a)}{=} \left(\prod_{i=1}^{k+1} A_i\right) \otimes \left(\prod_{i=1}^{k+1} B_i\right),\end{aligned}$$

where (a) follows from the base step (Theorem B.3), which completes the proof. ■

B.1.1 Decomposition of Unitary Transformations

In this section, a necessary and sufficient condition for when a unitary transformation $U \in \operatorname{End}(\mathscr{H}_A \otimes \mathscr{H}_B)$ for some systems A, B, can be decomposed into two unitary transformation $U_A \in \operatorname{End}(\mathscr{H}_A)$ and $U_B \in \operatorname{End}(\mathscr{H}_B)$, is shown. It is based on [ACFW05, ch. 2.1].

Before giving the result, some notation is introduced. The vectorisation of a matrix $T \in \operatorname{Hom}(\mathbb{C}^m, \mathbb{C}^n)$ is given by

$$\operatorname{vec}(T) = \begin{bmatrix} t_{11} & \cdots & t_{m1} & t_{12} & \cdots & t_{m2} & \cdots & t_{1n} & \cdots & t_{mn} \end{bmatrix}^\top.$$

If T instead is a $m \times n$ block matrix, the blocks can be 'vectorised' similarly as

$$\operatorname{Vec}(T) = \begin{bmatrix} T_{11} & \cdots & T_{m1} & T_{12} & \cdots & T_{m2} & \cdots & T_{1n} & \cdots & T_{mn} \end{bmatrix}^\top,$$

where the capital letters are used to indicate that it is block matrices rather than scalars. The terminology of vectorisation of such a block matrix is misleading as it remains a matrix, however, its meaning should be clear from context. In order for it to become properly vectorised, each block matrix must be vectorised. However, for now it is more interesting to vectorise the block matrices and transposing them, which yields the following matrix:

$$\tilde{T} = \begin{bmatrix} \operatorname{vec}(T_{11})^\top & \cdots & \operatorname{vec}(T_{m1})^\top & \cdots & \operatorname{vec}(T_{1m})^\top & \cdots & \operatorname{vec}(T_{mm})^\top \end{bmatrix}^\top.$$

For clarification, the general expression for $\tilde{T}$ for $T \in \mathrm{End}(\mathbb{C}^4)$ is

$$\begin{aligned}\tilde{T} &= \begin{bmatrix} \mathrm{vec}\left(\begin{bmatrix} t_{11} & t_{12} \\ t_{21} & t_{22} \end{bmatrix}\right)^\top & \mathrm{vec}\left(\begin{bmatrix} t_{31} & t_{32} \\ t_{41} & t_{42} \end{bmatrix}\right)^\top & \mathrm{vec}\left(\begin{bmatrix} t_{13} & t_{14} \\ t_{23} & t_{24} \end{bmatrix}\right)^\top & \mathrm{vec}\left(\begin{bmatrix} t_{33} & t_{34} \\ t_{43} & t_{44} \end{bmatrix}\right)^\top \end{bmatrix}^\top \\ &= \begin{bmatrix} t_{11} & t_{21} & t_{12} & t_{22} \\ t_{31} & t_{41} & t_{32} & t_{42} \\ t_{13} & t_{23} & t_{14} & t_{24} \\ t_{33} & t_{43} & t_{34} & t_{44} \end{bmatrix}.\end{aligned}$$

With simple algebra, it can be shown that

$$U = U_A \otimes U_B \iff \tilde{U} = \mathrm{vec}(U_A)\mathrm{vec}(U_B)^\top. \tag{B.1}$$

Based on this, the necessary and sufficient condition for when such a decomposition is possible can be stated.

Theorem B.5: Kronecker Product Decomposition of a Unitary Transformation
The unitary transformation U can be decomposed with U_A and U_B such that $U = U_A \otimes U_B$ if and only if $\mathrm{rank}(\tilde{U}) = 1$. [ACFW05, p. 3]

Proof
Assume that there exists U_A, U_B such that $U = U_A \otimes U_B$. From (B.1), it follows that $\tilde{U} = \mathrm{vec}(U_A)\mathrm{vec}(U_B)^\top$. The right-hand side is an outer product, hence clearly has rank one, implying that $\mathrm{rank}(\tilde{U}) = 1$.

Conversely, assume that $\mathrm{rank}(\tilde{U}) = 1$. Then there exists some operators X, Y that are not necessarily unitary such that $\tilde{U} = \mathrm{vec}(X)\mathrm{vec}(Y)^\top$, which by (B.1) implies $U = X \otimes Y$. The unitarity of U then implies that

$$I = UU^\dagger = (X \otimes Y)(X \otimes Y)^\dagger = (X \otimes Y)(X^\dagger \otimes Y^\dagger) = (XX^\dagger) \otimes (YY^\dagger).$$

From this relation, it follows that for $c \in \mathbb{C} \setminus \{0\}$, then

$$XX^\dagger = cI, \quad YY^\dagger = \frac{1}{c}I.$$

Similar arguments can be used on $I = U^\dagger U$, yielding that for $k \in \mathbb{C} \setminus \{0\}$, then

$$X^\dagger X = kI, \quad Y^\dagger Y = \frac{1}{k}I.$$

Now notice that $X^\dagger X$ is Hermitian, which has two important implications. Firstly, X is normal meaning that $X^\dagger X = XX^\dagger$, which implies that $c = k$. Secondly, kI is Hermitian as well, which not only implies that k is real, but also that kI is positive semi-definite as it decomposes as $X^\dagger X$. This in turn implies that k is positive.

Using this, X and Y can be normalised in order to turn them unitary. Explicitly, defining the unitary operators

$$U_A = \frac{1}{\sqrt{k}}X, \quad U_B = \sqrt{k}Y,$$

gives that $U = U_A \otimes U_B$, which concludes the proof. ■

A very simple example of a unitary transformation $U \in \mathrm{End}(\mathbb{C}^4)$ with $\mathrm{rank}(\tilde{U}) \neq 1$, hence U cannot be decomposed into a Kronecker product of two unitary transformation $U_A, U_B \in \mathrm{End}(\mathbb{C}^2)$, is

$$U = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & -1 & 0 & 0 \\ 0 & 0 & 1 & -1 \\ 0 & 0 & 1 & 1 \end{bmatrix}.$$

Even without Theorem B.5 is it easily verifiable that it cannot be decomposed. For $U = U_A \otimes U_B$ to hold, it is straightforward to see that $(U_A)_{12} = (U_A)_{21} = 0$. However, this implies that $U_A \in \{\pm I, \pm iI\}$, which then implies that there is no U_B satisfying that $U = U_A \otimes U_B$.

B.2 Exponentiation Identity for the Matrix Exponential

In this section, it is proven that the exponential identity $\mathrm{e}^a\mathrm{e}^b = \mathrm{e}^{a+b}$ for $a, b \in \mathbb{R}$ also holds for commuting matrices.

> **Theorem B.6: Exponentiation Identity**
> If the matrices A, B commutes, then $\mathrm{e}^{\alpha A}\mathrm{e}^{\beta B} = \mathrm{e}^{\alpha A+\beta B}$.

Proof
By using the power series of the matrix exponential, it follows that

$$\begin{aligned}\mathrm{e}^{\alpha A}\mathrm{e}^{\beta B} &= \left(\sum_{n=0}^{\infty}\frac{1}{n!}(\alpha A)^n\right)\left(\sum_{m=0}^{\infty}\frac{1}{m!}(\beta B)^m\right) \overset{(a)}{=} \sum_{n=0}^{\infty}\sum_{m=0}^{n}\frac{1}{m!}(\alpha A)^m\frac{1}{(n-m)!}(\beta B)^{n-m}\\ &= \sum_{n=0}^{\infty}\frac{1}{n!}\sum_{m=0}^{n}\binom{n}{m}(\alpha A)^m(\beta B)^{n-m} \overset{(b)}{=} \sum_{n=0}^{\infty}\frac{1}{n!}(\alpha A+\beta B)^n = \mathrm{e}^{\alpha A+\beta B},\end{aligned}$$

where (a) follows from the Cauchy product and (b) from the binomial theorem, which only holds when A, B commutes. ■

B.3 Pauli Matrices

Since the Pauli matrices defined in Definition 2.9 are important in quantum mechanics, some basic properties regarding them are derived in this section. The Pauli matrices are repeated here for readability:

$$X = \begin{bmatrix}0 & 1\\ 1 & 0\end{bmatrix}, \quad Y\begin{bmatrix}0 & -\mathrm{i}\\ \mathrm{i} & 0\end{bmatrix}, \quad Z\begin{bmatrix}1 & 0\\ 0 & -1\end{bmatrix}.$$

The properties are stated in the following theorem.

> **Theorem B.7: Properties of Pauli matrices**
> The Pauli matrices X, Y, Z satisfies the following properties:
>
> i) They are Hermitian and unitary (and thereby involutory)
>
> ii) They anti-commute with each other,
>
> iii) They have eigenvalues ± 1,
>
> iv) Together with I, they constitute an orthogonal basis of $\mathbb{C}^{2\times 2}$ over $\mathbb{C}$ with respect to the Hilbert-Schmidt inner product,
>
> v) Together with I, they constitute an orthogonal basis of Hermitian matrices in $\mathbb{C}^{2\times 2}$ over $\mathbb{R}$ with respect to the Hilbert-Schmidt inner product.
>
> [NC10, pp. 71,77,78]

Before proving the theorem, recall that the Hilbert-Schmidt inner product of the matrices $A, B \in \mathrm{End}(\mathbb{C}^n)$ is, assuming that it is well-defined, given as

$$\langle A, B\rangle_{\mathrm{HS}} = \mathrm{Tr}\big(A^\dagger B\big).$$

Proof
i) The Pauli matrices are easily seen to be Hermitian as X, Z are real-valued and symmetric, while Y is purely imaginary and anti-symmetric. The unitarity is also easily seen as the columns of each matrix constitute an orthonormal basis of $\mathbb{C}^2$.

ii) This follows by direct computation. Explicitly:

$$XY = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} 0 & -\mathrm{i} \\ \mathrm{i} & 0 \end{bmatrix} = \begin{bmatrix} \mathrm{i} & 0 \\ 0 & -\mathrm{i} \end{bmatrix} = \mathrm{i}Z, \quad YX = \begin{bmatrix} 0 & -\mathrm{i} \\ \mathrm{i} & 0 \end{bmatrix} \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} = \begin{bmatrix} -\mathrm{i} & 0 \\ 0 & \mathrm{i} \end{bmatrix} = -\mathrm{i}Z,$$
$$XZ = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} = -\mathrm{i}Y, \quad ZX = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} = \mathrm{i}Y,$$
$$YZ = \begin{bmatrix} 0 & -\mathrm{i} \\ \mathrm{i} & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} = \begin{bmatrix} 0 & \mathrm{i} \\ \mathrm{i} & 0 \end{bmatrix} = \mathrm{i}X, \quad ZY = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \begin{bmatrix} 0 & -\mathrm{i} \\ \mathrm{i} & 0 \end{bmatrix} = \begin{bmatrix} 0 & -\mathrm{i} \\ -\mathrm{i} & 0 \end{bmatrix} = -\mathrm{i}X.$$

iii) It is easily seen that $\mathrm{Tr}(X) = \mathrm{Tr}(Y) = \mathrm{Tr}(Z) = 0$ while $\det(X) = \det(Y) = \det(Z) = -1$. Thus, the sum of eigenvalues is zero, while the product of them is minus one. Hence, it can be concluded the eigenvalues of X, Y, Z are ± 1.

iv) In order to show that it is an orthogonal basis, it is sufficient to show that the matrices are orthogonal and span $\mathrm{End}(\mathbb{C}^2)$. Orthogonality is firstly shown. As shown in iii), the Pauli matrices have zero trace, hence they are trivially orthogonal to the identity matrix. By combining i) and ii), it follows that the Pauli matrices are mutually orthogonal; i) gives that the Pauli matrices are Hermitian, hence it is sufficient to consider products of Pauli matrices in the Hilbert-Schmidt inner product, that is, with no conjugate transpose. But then ii) gives that the product of two Pauli matrices simply is a scaling of the remaining Pauli matrix, hence the trace remains zero. Thus, $\{I, X, Y, Z\}$ are mutually orthogonal.

To show that they span $\mathrm{End}(\mathbb{C}^2)$, there must always be a solution to the following equation for an arbitrary matrix $T \in \mathrm{End}(\mathbb{C}^2)$:

$$\begin{bmatrix} t_{11} & t_{12} \\ t_{21} & t_{22} \end{bmatrix} = aI + bX + cY + dZ = \begin{bmatrix} a+d & b-\mathrm{i}c \\ b+\mathrm{i}c & a-d \end{bmatrix}.$$

Some simple algebra shows that the solution is

$$a = \frac{t_{11}+t_{22}}{2}, \quad b = \frac{t_{12}+t_{21}}{2}, \quad c = \frac{\mathrm{i}(t_{12}-t_{21})}{2}, \quad d = \frac{t_{11}-t_{22}}{2},$$

which concludes they they span $\mathrm{End}(\mathbb{C}^2)$.

v) This is a specific case of iv). Explicitly, if T is Hermitian, its diagonal entries are real, while the off-diagonal entries are each others' complex conjugate. By using this in the expressions for a, b, c, d, it follows directly that $a, b, c, d \in \mathbb{R}$. This concludes the proof. ■

Another useful result regarding the Pauli matrices are their Frobenius covariants. Recall that for a diagonalisable matrix A with eigenvalues $\lambda_1, \ldots, \lambda_n$, the i'th Frobenius covariant is given as

$$A_i = \prod_{\substack{j=1 \\ j\neq i}}^{n} \frac{1}{\lambda_i - \lambda_j}(A - \lambda_j I).$$

Since the eigenvalues of the Pauli matrices are ± 1, the covariants have similar structure. For example, considering X, the covariants are:

$$X_{+1} = \frac{1}{1-(-1)}(X - (-1)I) = \frac{I+X}{2}, \quad X_{-1} = \frac{1}{-1-1}(X - 1I) = \frac{I-X}{2}.$$

The Frobenius covariants for the other Pauli matrices follow analogously.

B.4 Partial Trace over Orthonormal Basis

Given an orthonormal basis $\{v_n\}_n$ for some vector space V and an operator $O \in \mathrm{End}(V)$, the trace of O can be calculated as

$$\mathrm{Tr}(O) = \sum_n v_n^\dagger O v_n.$$

A similar result holds true for the partial trace.

Theorem B.8: Partial Traces given Orthonormal Bases
Let $\mathcal{H} = \mathcal{H}_A \otimes \mathcal{H}_B$ and $O \in \operatorname{End}(\mathscr{H})$. Furthermore, let $\{|a_n\rangle\}_n$ and $\{|b_m\rangle\}_m$ be orthonormal bases of $\mathcal{H}_A$ and $\mathcal{H}_B$, respectively. The partial trace over $\mathcal{H}_B$ is then given as

$$\operatorname{Tr}_B(O) = \sum_m (I_A \otimes \langle b_m|) O (I_A \otimes |b_m\rangle).$$

The partial trace over $\mathcal{H}_A$ is given analogously.

Proof
First, notice that

$$\begin{aligned}
O &= (I_A \otimes I_B) O (I_A \otimes I_B) \\
&= \sum_n \sum_m \Big(\big(|a_n\rangle \langle a_n| \big) \otimes \big(|b_m\rangle \langle b_m| \big) \Big) O \Big(\big(|a_n\rangle \langle a_n| \big) \otimes \big(|b_m\rangle \langle b_m| \big) \Big) \\
&= \sum_n \sum_m \big(|a_n\rangle \otimes |b_m\rangle \big) \big(\langle a_n| \otimes \langle b_m| \big) O \big(|a_n\rangle \otimes |b_m\rangle \big) \big(\langle a_n| \otimes \langle b_m| \big) \\
&= \sum_n \sum_m \big(\langle a_n| \otimes \langle b_m| \big) O \big(|a_n\rangle \otimes |b_m\rangle \big) \big(|a_n\rangle \langle a_n| \big) \otimes \big(|b_m\rangle \langle b_m| \big).
\end{aligned}$$

Thus, taking the partial trace of O over $\mathcal{H}_B$ yields

$$\begin{aligned}
\operatorname{Tr}_B(O) &= \sum_n \sum_m \big(\langle a_n| \otimes \langle b_m| \big) O \big(|a_n\rangle \otimes |b_m\rangle \big) \big(|a_n\rangle \langle a_n| \big) \operatorname{Tr}(|b_m\rangle \langle b_m|) \\
&= \sum_n \sum_m |a_n\rangle \big(\langle a_n| \otimes \langle b_m| \big) O \big(|a_n\rangle \otimes |b_m\rangle \big) \langle a_n| \\
&= \sum_n \sum_m \Big(\big(|a_n\rangle \langle a_n| \big) \otimes \langle b_m| \Big) O \Big(\big(|a_n\rangle \langle a_n| \big) \otimes |b_m\rangle \Big) \\
&= \sum_m (I_A \otimes \langle b_m|) O (I_A \otimes |b_m\rangle).
\end{aligned}$$

■

C | Introduction to Group Theory

A gentle introduction to group theory needed for the stabilizer formalism is given in this appendix, which is based upon [DF04, ch. 1,2] and [NC10, app. 2].

Algebra gives an abstract description of things that posses the same structure, which enables results regarding one thing to translate to the others. One of the most fundamental of such structures is a group.

> **Definition C.1: Group**
> Let G be a set and $*$ be a binary operator on G, i.e., $* : G \times G \to G$. The ordered pair $(G, *)$ is called a group if it satisfies the following axioms:
>
> i) Associativity: $(a * b) * c = a * (b * c)$ for all $a, b, c \in G$,
>
> ii) Identity: There exists $e \in G$ such that $e * a = a * e = a$ for all $a \in G$,
>
> iii) Inverse: For each $a \in G$, there exists $a^{-1} \in G$ such that $a * a^{-1} = a^{-1} * a = e$.
>
> If the binary furthermore is commutative such that $a * b = b * a$ for all $a, b \in G$, the group is called abelian. [DF04, pp. 16-17]

A few remarks on notation is on order. Firstly, it is common to simply say that G is a group under $*$ rather than giving the ordered pair $(G, *)$. Secondly, the operator is occasionally omitted in expressions, e.g., it is often written with juxtaposition when the operator is multiplication.

There are also conceptual remarks that are noteworthy. The existence of an identity in the second axiom implies that $G \neq \emptyset$. Furthermore, the axioms required for a group are only partly those of a field, hence a group is more general than a field. Specifically, a field requires two operations that satisfy similar axioms as well as being distributive. This implies that every field is a group, which already provide many groups, e.g., $\mathbb{R}, \mathbb{C}$ under addition, $\mathbb{R} \setminus \{0\}, \mathbb{C} \setminus \{0\}$ under multiplication, and Galois fields restricted to one of its operators. Another example of a simple group that will serve as an example throughout this appendix is provided in Example C.2.

> **Example C.2: Group over Integers**
> Consider the set of integers $\mathbb{Z}$. It forms a group under addition; i) scalar addition is associative, ii) $0 \in \mathbb{Z}$ is the additive identity, and iii) for every $z \in \mathbb{Z}$ then there exists $-z \in \mathbb{Z}$, hence an additive inverse exists. Particularly, $(\mathbb{Z}, +)$ is an abelian group since scalar addition is commutative.
>
> However, $\mathbb{Z} \setminus \{0\}$ does not form a group under multiplication since the only element in the set that has an multiplicative inverse is 1.

An important characteristic of a group is the cardinality of the underlying set G, which is denoted $|G|$. If it is finite, the group G is said to be finite. In that case, $|G|$ define the order of the group.

By considering the group $\mathbb{Z}$ given in Example C.2, it it clear that it is in fact possible to remove some of the elements, e.g., the odd integers $\mathbb{Z}_{\text{odd}}$, without breaking any of the group axioms. This implies that the even integers $\mathbb{Z}_{\text{even}}$ is a group. A group created in such a manner is called a subgroup.

> **Definition C.3: Subgroup**
> Let $(G, *)$ be a group and H be a subset of the set G. The subset H is called a subgroup of $(G, *)$ if $(H, *)$ is a group. [DF04, p. 46]

Notice that if $(G, *)$ is a group, then $*$ is clearly also associative for $H \subset G$. Thus, it is easier to test whether or not H is a subgroup of a group G than to test if H is a group. This is done by showing that

H is closed under $*$, it contains the identity, and that every of its elements has inverses. This is equivalent to the following simpler criterion.

Theorem C.4: The Subgroup Criterion
Let $(G, *)$ be a group. Then $H \subset G$ is a subgroup under $*$ if and only if it satisfies:

i) $H \neq \emptyset$

ii) For all $a, b \in H$, then $a * b^{-1} \in H$.

If H is finite, the second condition can be simplified to $a * b \in H$ for all $a, b \in H$. [DF04, p. 47]

Proof
Assume that H is a subgroup of G under $*$. By definition, H is then itself a group under $*$, and the group axioms then immediately implies that the two conditions are satisfied.

Conversely, assume that the two conditions holds. The first condition implies that there exists some $a \in H$, so let this element be given. The second condition then gives that $a * a^{-1} = e \in H$. Hence, the identity is contained in H. This in turn implies that $1 * a^{-1} = a^{-1} \in H$, hence H contains inverses. By picking an arbitrary $b \in H$, it therefore follows that $b^{-1} \in H$, which by the second condition implies that $a * (b^{-1})^{-1} = a * b \in H$. This concludes that H is a group, thus also a subgroup of G.

If H is finite then for every $a \in H$, there exists only finite distinct elements $a, a * a, a * a * a, \ldots$. Thus there exists $m > n$ such that $a *$ itself n times is equal to $a *$ itself m times, i.e., $a^n = a^m$ if the operation is multiplication. This then gives that $a^{m-n} = 1$, which in turn yields that $a^{m-n-1} = a^{-1} \in H$, hence H is closed under inverses. ■

It is sometimes more convenient to characterise a group G by how they act as a mapping on another set S. Hence it is useful to consider a mapping $A : G \times S \to S$. If such a mapping have an identity element and composition is equivalent to a group operation ($A(a * b, s) = A(a, A(b, s))$ for $a, b \in G$ and $s \in S$), it is called a group action. One particularly interesting group action is the stabilizer, which contains elements in G that acts as an identity on some element in S.

Definition C.5: Stabilizer
Let $(G, *)$ be a group, S be some set, and $A : G \times S \to S$ be a group action. For a particular $s \in S$, the stabilizer of s by G is defined as the set

$$G_s = \{g \in G \,|\, A(g, s) = s\}.$$

[DF04, p. 51]

The stabilizer is also commonly called the stabilizer group, which is justified by the following theorem.

Theorem C.6: The Stabilizer is a Subgroup
For each $s \in S$, the stabilizer of s by G is a subgroup of G. [DF04, p. 51]

Proof
In order to show that it is a subgroup, the subgroup criterion must be satisfied. To do so, let $s \in S$ be given.

It is firstly proven that $G_s \neq \emptyset$. By construction, every group action contains an identity, i.e., there exists $e \in G$ such that $A(e, s) = s$, hence $e \in G_s$.

It is secondly proven that G_s is closed under $*$ and inverses. If $a \in G_s$, then

$$s = A(e, s) = A(a^{-1} \times a, s) \overset{(a)}{=} A(a^{-1}, A(a, s)) = A(a^{-1}, s) \implies a^{-1} \in G_s,$$

where (a) follows from the condition that composition and group operations are equivalent.

If $a, b \in G_s$, then

$$A(a \times b, s) = A(a, A(b, s)) = A(a, s) = s \implies a \times b \in G_s.$$

Combining these implications shows that G_s is closed under $*$ and inverses. ■

Two other important subgroups are the centraliser and the normaliser.

Definition C.7: Centraliser and Normaliser
Let $(G, *)$ be a group and S some non-empty subset of G. The centraliser of S in G is the subset

$$C(S) = \{g \in G \,|\, gsg^{-1} = s \,\forall\, s \in S\}.$$

Furthermore, let $gSg^{-1} = \{gsg^{-1} \,|\, s \in S\}$. The normaliser of S in G is the subset

$$N(S) = \{g \in G \,|\, gSg^{-1} = S\}.$$

[DF04, p. 50]

Notice that $C(S)$ is different from $N(S)$ since $g \in C(S)$ implies that g commutes with every element in S, while $g \in N(S)$ only implies that g commutes with the set S, but not necessarily element-wise. Loosely speaking, $g \in C(S)$ implies $gsg^{-1} = s$, while $g \in N(S)$ implies $gsg^{-1} = s'$ for some $s' \in S$. From the distinction, it is clear that $C(S) \subseteq N(S)$. Actually, both of these subsets are also subgroups of G_n, for which a proof can be found in [DF04, ch. 2.2].

When working with groups it can sometimes be tedious to explicitly write all the terms every time. Hence it would be convenient if a group could be denoted using a smaller set, which exactly is the concept of generators.

Definition C.8: Generator
Let G be a group and S be a subset of the elements in G. If every element of G can be written as a combination of elements in S and there inverses, S is called a generator of G, which is denoted $G = \langle S \rangle$.

[DF04, p. 26]

If S is a generator of G, it is also said that S generates G. By continuing on Example C.2, a generator of $\mathbb{Z}$ can be found.

Example C.9: Generator of Integers
Since every integer can be written as a sum of ± 1, it follows that $\mathbb{Z} = \langle 1 \rangle$. Similar reasoning implies that another generator could be given, e.g., $\mathbb{Z} = \langle 1, 5 \rangle$.

Example C.9 gives two different generators for $\mathbb{Z}$. In the second case where $\langle 1, 5 \rangle$, the elements are said to be dependent since 5 can be written in terms of 1. Thus, it is generally desirable to have independent generators, meaning that no generating element can be written in terms of the other generating elements, as this leads to a more compact description of the generator. For finite groups is it actually possible to determine a bound on the size of the generator.

Theorem C.10: Size of Generating Set
Let $(G, *)$ be a group of order $|G|$. Then there exists a set of generators, $\{g_1, \dots, g_m\}$, of G such that $m \leq \log_2(|G|)$. [NC10, p. 611]

Proof
Let $G = \langle g_1, \dots, g_m \rangle$ be a finite group. Furthermore, for $i \in \{1, \dots, m\}$, denote $G_i = \langle g_1, \dots, g_i \rangle$. Now, assume that $g_{n+1} \notin G_n$. Then for every $g_i \in G_n$, it holds that $g_{n+1} * g_i \notin G_n$, since this would imply that $g_{n+1} * g_i * g_i^{-1} = g_{n+1} \in G_n$, which is a contradiction to the assumption. Since $g_{n+1} * g_i \neq g_{n+1} * g_j$ for $i, j \in \{1, \dots, n\}$ satisfying $i \neq j$, it follows that $|G_n| \leq 2|G_{n+1}|$. By recursion it then follows that

$$|G| = |G_m| \geq 2|G_{m-1}| \geq \cdots \geq 2^m |G_1| \geq 2^m.$$

Thus $m = \log_2(|G|)$, which concludes the proof. ■