

# Fault-tolerant quantum computation

Implementation of a fault-tolerant SWAP  
operation on the IBM 5-qubit device

by

J.S. De Jong

to obtain the degree of Master of Science  
at the Delft University of Technology,  
to be defended publicly on Monday January 28, 2019 at 1:00 PM.

Student number: 4236688  
Thesis committee: Prof. dr. ir. B. M. Terhal, TU Delft, supervisor  
Dr. M. Blaauboer, TU Delft  
Dr. D. Elkouss Coronas, TU Delft

*This thesis is confidential and cannot be made public until January 28, 2019.*

An electronic version of this thesis is available at <http://repository.tudelft.nl/>.



# Abstract

Quantum computing is a field that shows tremendous possibilities and promise. It can provide an exponential speedup compared to classical computers in many computational problems, including simulations of general quantum mechanical systems, pattern finding and solving linear systems.

Quantum computations can be performed by making use of qubits and performing operations, called gates, on them. A physical realisation of a qubit will be faulty, and to combat the errors that will inadvertently but unavoidably occur, the use of a quantum error correction code (QECC) is needed. A QECC is able to correct static errors; errors that occur on the states. However, it can not correct errors that occur on and during the operations.

To make the computer resilient to these errors, the implementation of the QECC has to be performed *fault-tolerantly*. A fault-tolerant implementation of a quantum operation is designed in such a way that errors that happen during the operation, through faulty gates, do not propagate to errors on the states that are not correctable any more by the QECC. Hence, error propagation is a key concept in fault-tolerance. Every element of a general quantum circuit needs to be designed fault-tolerant. One of the elements that will be needed in any realistic quantum computer architecture is the possibility of exchanging the states of two qubits, so that an arbitrary pair of qubits can be brought physically close, allowing the implementation of multi-qubit gates. Exchanging the states of two qubits can be performed by a fault-tolerant SWAP gate, that makes use of one ancilla qubit.

In this thesis, a fault-tolerant SWAP operation is implemented on the IBM 5-qubit ‘Tenerife’ device. It is characterized as a quantum channel by performing quantum process tomography (QPT). The CPTP constraints of quantum channels in the framework of QPT are discussed and a method of obtaining a CPTP estimate of an initial QPT reconstruction is given. The error process of the fault-tolerant SWAP is analysed and subsequently compared to a normal implementation of the SWAP gate, on which tomography is performed as well. The main measure that is used in the comparison is the ratio  $r$  of multi-qubit errors (or correlated) to single-qubit (or uncorrelated) errors in the two implementations.

SPAM errors are an important source of estimated infidelity. To filter these SPAM errors and obtain a better estimate of only the circuits, SPAM errors are approximated to be only measurement errors. The errors are characterized by performing tomography on an arbitrary short identity channel, and subsequently using this as an estimate of the measurement error map. This gives a method to estimate a representation of the circuit itself.

We can report on an error ratio  $r$  of  $0.170 \pm 0.0029$  for the fault-tolerant SWAP operation and an error ratio  $r$  of  $1.256 \pm 0.0129$  for the non-fault-tolerant SWAP operation. However, the total error in the fault-tolerant SWAP is much higher, resulting in a process fidelity  $F_p = 0.608$  for the fault-tolerant implementation and  $F_p = 0.742$  for the normal implementation. This is due to the larger number of gates in the fault-tolerant circuit. This research shows that a SWAP operation can be implemented fault-tolerantly, but that the error rates of the devices need to be reduced before their use becomes viable.



# Preface and Acknowledgements

Before you lies the document that is written by me, Jarn de Jong, in partial fulfillment of the degree of Master of Science in Applied Physics at the Delft University of Technology. This thesis is the final result of a year of research, which was an amazing experience. It definitely was trying at times, but that makes me all the more proud to present this work. However, this work would not have been realized were it for many people, who I would like to thank.

First and foremost, I would like to thank Dr. Ben Criger, my supervisor, mentor and inexhaustible source of help during the past months. Where I heard stories of supervisors that scarcely had time to answer questions even once a week, for me it was exactly the other way around. At the start of my project, Ben asked me to give him two questions a day - and he was rather serious about it too. I can quote him on saying: *“Students questions are my number one priority, Jarn. If I were to have a phone, you would be allowed to call me at three o'clock in the morning to ask me a scientific question”*<sup>1</sup>. Without Ben this document would be very different indeed - I would not have learned half as much or would not have succeeded in half as much if it weren't for him. He has spent countless hours on me, and my enormous gratitude goes to him for it.

Secondly, I would like to thank my professor, Barbara Terhal, for taking me as a master student and including me in the Terhal group. I am honoured to have been part of the group. Furthermore, I would like to thank everyone in the group for welcoming me, and especially Christophe, Daniel and Joel for assisting Ben in helping me whenever he was not available.

I also acknowledge the IBM Q Experience team for providing access to their 5-qubit device. Specifically, I'd like to thank Andrew Cross for providing me the means (i.e. credits) to perform the experiments, and Ali Javadi, Jay Gambetta and Chris Wood for various discussions and help.

About 90% of my time at QuTech I have spent in 2 rooms: where most master students only get to have one master-student room, I was lucky to have two: F218 and B207 (this is because I switched, and I've been banished every since by the people in the first room). It feels like that I have spent more time in those two rooms than any other in the last year, and I would like to thank Bart, Michael and Romy from F218 and Guus, Jonathan, Marius, Mark, Patrick, Remon and Sjoerd from B207 for being my roommates; I will cherish my days with you at QuTech. Especially I want to thank Constantijn, with whom I have had many fruitful discussions.

I actually have not been sleeping under my desk and therefore also have other roommates: Cas, Gijs, Nima, Marc, Sebas and Willem (and Stan pretty much too). Guys, thank you for enduring me whenever I couldn't stop rambling about qubits, proces matrices, error correction or fault-tolerance, and thank you for having a hot dinner ready when I got home at 10 in the evening after a long day of writing.

I am lucky to have many dear friends, and in addition to the people already mentioned I'd like to thank Fleur, Nena, Nico, Stephan and Tom for making me the person that I am today. Of course, no acknowledgement of mine would be complete without thanking Mats, my closest friend, who has been there for me time and time again over the many, many years that I've known him.

Finally, I would like to thank my parents, Roel and Joselien, and my brother Anner, for supporting me no matter what and loving me unconditionally.

*J.S. De Jong  
Delft, January 2019*

---

<sup>1</sup>Ben does, in fact, have a phone, but he does not want anybody to know, to prevent his students calling him at three o'clock in the morning to ask him a scientific question.



# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Classical and quantum computing . . . . .	1
1.2	Goal of this research . . . . .	3
1.3	Structure of this thesis . . . . .	3
<b>2</b>	<b>Introduction to quantum information science</b>	<b>5</b>
2.1	The basic units of quantum computation: Qubits and operations . . . . .	5
2.2	Common unitary operations . . . . .	9
2.3	Quantum circuits . . . . .	12
2.4	Density matrices . . . . .	13
2.5	The Bloch sphere . . . . .	14
2.6	State fidelity . . . . .	15
<b>3</b>	<b>Quantum channels and open quantum systems</b>	<b>17</b>
3.1	Channels as linear maps . . . . .	17
3.2	Representations of $\Lambda$ . . . . .	18
3.3	Relations between different representations . . . . .	20
3.4	Common channels . . . . .	21
3.5	Common error channels . . . . .	22
3.6	Pauli Twirl approximation . . . . .	25
3.7	Process fidelity . . . . .	26
<b>4</b>	<b>Quantum process tomography</b>	<b>27</b>
4.1	Standard quantum process tomography . . . . .	27
4.2	Errors and noise on data . . . . .	30
4.3	Ensuring CP and TP . . . . .	31
4.4	The error matrix $\chi^{\text{err}}$ . . . . .	33
4.5	Other characterization methods . . . . .	35
<b>5</b>	<b>Stabilizer codes</b>	<b>37</b>
5.1	Quantum error correcting codes . . . . .	37
5.2	Stabilizer codes . . . . .	39
5.3	Examples of stabilizer codes . . . . .	42
5.4	Syndrome measurement . . . . .	42
5.5	Clifford operators . . . . .	43
<b>6</b>	<b>Fault-tolerant computation</b>	<b>45</b>
6.1	Error propagation . . . . .	45
6.2	Fault-tolerance . . . . .	46
6.3	Connectivity between qubits . . . . .	50

<b>7</b>	<b>The fault-tolerant SWAP circuit</b>	<b>51</b>
7.1	The fault tolerant SWAP operation . . . . .	51
7.2	Implementation on a quantum chip . . . . .	52
7.3	Simulations of the experiments and circuits . . . . .	53
<b>8</b>	<b>Results of tomography experiments</b>	<b>59</b>
8.1	Preliminary results and SPAM errors . . . . .	59
8.2	Results after filtering . . . . .	60
<b>9</b>	<b>Discussion and conclusion</b>	<b>65</b>
9.1	Discussion of the methods . . . . .	65
9.2	Discussion of the results . . . . .	67
9.3	Conclusion . . . . .	68
9.4	Recommendations . . . . .	68
<b>A</b>	<b>Proofs for chapter 3</b>	<b>71</b>
A.1	Complete positivity in representations of $\Lambda$ . . . . .	71
A.2	Trace-preservation in representations of $\Lambda$ . . . . .	74
A.3	Relations between different representations of $\Lambda$ . . . . .	77
<b>B</b>	<b>Preliminary results on tomography experiments</b>	<b>79</b>
B.1	Other simulation results . . . . .	79
B.2	Results of tomography experiments without SPAM filtering . . . . .	79
B.3	Results of experiments after filtering before CP subroutine . . . . .	79

# Chapter 1

## Introduction

### 1.1 Classical and quantum computing

The idea of using machines to perform computations is very old: mechanical calculators are as old as the start of the 19th century, and various other devices and objects that help with computing have existed for over thousands of years. During the first half of the 20th century, however, the area of automated digital computation emerged - it started with vacuum tubes, but once the transistor was invented in the 40s, the field increased tremendously in size, and modern life can no longer be imagined without digital computers. The basic element of such a computer is the *bit*; a two-state system that is either ‘on’ or ‘off’, ‘plus’ or ‘minus’ or, in its most-used representation, a 0 or a 1. Computations are done by performing *operations* on these bits, which are also called *gates*. Billions of bits are combined into a modern computer, and the power of these machines is ever increasing.

There are, however, limits to the power of these computers. When computer scientists talk about the power or speed of a computation, they are mainly interested in how the computation scales with respect to the size of the input parameter of the computation. If the time that it requires to perform the computation scales polynomially with the size of the input parameter, the computation is said to be *efficient*. If, however, the time that it requires to perform a computation scales exponentially with the size of the input parameter, the computation is inefficient.

To illustrate the distinction between polynomial and exponential scaling, consider a problem where the goal is to check if a given number  $N$  is prime. The most straightforward method to do this is to divide through every number lower than the given number, and check if there is a remainder<sup>1</sup>. If  $N$  is a number with five digits, there are  $\sim 10^5$  different numbers to check. If our computer can check 1000 numbers a second, it will take 100 seconds in total to perform the computation. The input parameter here is the *length* of  $N$ , i.e. its amount of digits. If the input parameter is now twice the size, the number of different numbers to check is  $10^{10}$ : at the same rate of 1000 numbers a second, the computation will take about 115 *days*. Again doubling the input parameter results in  $10^{20}$  different numbers: the same computer now takes more than three *billion years* - about 60% of the age of the earth. Doubling the input parameter is relatively easy to do - we only need to keep track of twice as many digits - but the computation itself is *exponentially hard*.

There are many other known (and important) problems that are not efficiently computable on a digital computer. Physicists and computer scientists in the 80s realized that simulating general quantum mechanical systems with normal, digital computers would be inherently inefficient. They also realized that these quantum mechanical systems could be efficient in simulating each other. Instead of a normal, digital bit, what is now known as a *classical bit*, a quantum mechanical bit or *qubit* would be the basic element of this new computing device, which is now called a quantum computer. The qubit is not only allowed to be either 0 or 1, but it can be in a quantum mechanical *superposition*: a continuous range of different combinations of the two states. Furthermore, combining multiple qubits allows them to be in superpositions together. For certain superpositions the qubits are correlated: a qubit’s state cannot be described without describing the state of other qubits as well. This remarkable phenomenon is known as *quantum entanglement*. Superposition and entanglement both necessary for

---

<sup>1</sup>Of course there are much better ways to do this, but the general idea of scaling still holds.

quantum computing.

Many computations that are inefficient on classical computers are known to be efficient on quantum computers. Apart from simulating quantum systems, other examples in which a quantum computer offers exponential speedup are pattern matching, solving linear systems and prime factorization; algorithms for each can be found in [1]. A quantum computer can perform many of these calculations efficiently, whereas a classical computer can only perform them inefficiently. The field of quantum computing is immensely popular, but there is by no means a working full-scale quantum computer yet. Building a quantum computer with an adequate number of qubits - possibly millions - is not currently doable. The best current devices have about 100 qubits, and these qubits are far from perfect; they are faulty, meaning that errors occur that break the computations performed on them.

It has been said that quantum computing is in its '30s - there is tremendous interest, but so far, there are a multitude of competing architectures for the qubit, and there is no determined single way forward. Even though it is unclear what the eventual physical implementation of a quantum computer is going to be, one thing seems certain: reliable qubits are hard to make, and whatever the physical realization is going to be, it will be prone to errors. These errors will be too large to ignore - any substantial computation will build upon (possibly faulty) intermediate results, and will be rendered useless if nothing is done.

There are various ways of combating these errors. The first (rather obvious) way is to reduce the error rates of the various elements of the quantum computer. Although the importance of reducing error rates is evident, we can not obtain small enough error rates to completely solve the problem. Even if the error rates are very small, in a very big computation one intermediate error propagate to an enormous extent. To further combat errors, quantum error-correcting codes or QECCs exists. This technology provides a way forward, by encoding information or *logical qubits* into combinations of multiple *physical qubits*, thereby making the encoded information more resilient to errors. Quantum error correction is only one side of the coin - QECCs protect states, but they do not provide any correction for faulty operations. Without more, they are insufficient to combat any real errors. To actually perform resilient computations, the use of *fault-tolerance* is needed.

Fault-tolerant computation is, very loosely speaking, the solution to the problem that a physical implementation of a quantum computer will never be perfect. Any interaction with the qubits will be faulty, but a fault-tolerant operation is designed in such a way that the errors do not get out of hand. They will still occur, but by carefully designed methods they can be contained - if the error rate is low enough, the operation is then resilient to errors. The most straightforward implementation of an operation on qubits breaks a QECC - the errors induced by the operation become too abundant for the code to correct. A fault-tolerant implementation of the same operation provides a method to process the information without breaking the QECC; it requires an operation to be performed in such a way that potential errors do not propagate to errors that can not be corrected by the QECC.

Every element of a quantum computer needs a fault-tolerant implementation. The main elements of any quantum computation are the preparation of qubits in a fixed state, the measurements of qubits in a given basis, the implementation of the QECC and the various operations that are performed on the encoded information, which are quantum mechanical logic gates. A quantum gate might be applied to a single qubit, but computations with multiple qubits require multi-qubit gates that connect individual qubits. There exist fault-tolerant designs for all of these elements, the *threshold theorem* [2] proves that all these elements can be combined into a complete fault-tolerant computation. There are many caveats and considerations, and one important assumption is that any pair of qubits can be subjected to a multi-qubit gate. Multi-qubit gates can be implemented on qubits that are physically close, so any real quantum computer architecture will limit the connectivity of the qubits. To perform multi-qubit gates on arbitrary pairs of qubits, qubits (or their states) may need to be swapped around. There is a gate which performs this operation, which is the SWAP gate. This operation itself must also be implemented fault-tolerantly.

## 1.2 Goal of this research

In this thesis, we have investigated a fault-tolerant implementation of the SWAP gate, and have compared it to a standard non-fault-tolerant implementation. The fault-tolerant SWAP is implemented on a quantum device of IBM, that is accessible via the IBM Q Experience [3]. A fault-tolerant implementation of an operation should show certain properties and qualities when compared to a normal implementation, and the comparison is made on these properties. Both implementations of the SWAP gate are fully characterized and the analysis and comparison can be made using these characterizations.

## 1.3 Structure of this thesis

There are nine chapters in this thesis, including this introduction. A summary of every chapter can be found in the following outline.

To be able to discuss errors, QECCs and fault-tolerance, a basic foundation of quantum information science is needed, which is introduced in chapter 2. To combat errors, it is very useful to start by properly defining what the error processes on qubits are, this is discussed in chapter 3. To analyse the error on an actual physical system and characterize it as an error process, one needs the technique of *quantum process tomography*, which will be introduced in chapter 4. The error correcting codes and their properties are introduced in chapter 5, and the theory of fault-tolerance is introduced in chapter 6. The rest of the thesis is concerned with the fault-tolerant SWAP circuit. This circuit, together with simulations on the tomographic methods and the error processes in the circuit, is introduced in chapter 7. The results of the experiments can be found in chapter 8. Finally, chapter 9 concludes with a discussion on the methods and results and gives a main conclusion of this research and recommendations for further research.

Furthermore, there are two appendices. Appendix A provides proofs for various statements in chapter 3, and in appendix B intermediate or additional results of the simulations and experiments can be found.



## Chapter 2

# Introduction to quantum information science

Quantum information science is to quantum computing what (classical) information science is to normal computing. It is the theoretical framework for the computations that will be performed on quantum computers. The basic unit of a classical computer is the *bit*; it gets its name from its possible values, or *states*: either a 0 or a 1, a “binary digit” or “bit”. Classical computations are performed by means of *operations* on those bits. Conversely, the basic unit of a quantum computer is the *quantum bit* or *qubit*. Qubits are not restricted to either 0 or 1 (which are written as  $|0\rangle$  and  $|1\rangle$ ), but they can also be in a quantum mechanical *superposition*: a linear combination of the two states. Just as with classical computing, quantum computations are performed by means of operations on the qubits.

This chapter introduces qubits and the operations that can be performed on them, in section 2.1 and 2.2 respectively. Section 2.2 also introduces the *Pauli matrices*: a very important class of matrices that correspond to certain important qubit operations. In section 2.3, a brief introduction will be given on how to combine qubits and operations into a *quantum circuit*: a specification for an actual computation. Physical realisations of qubits will not be perfect, and to still specify them section 2.4 introduces the notion of *density matrices*. The chapter is concluded by the introduction of two important tools in the analysis of qubits and computations. Section 2.5 introduces the Bloch sphere, a visualisation of qubit states that is very helpful in interpreting qubit states and operations. Finally, to specify how good a quantum state is, section 2.6 introduces the idea of *state fidelity*, a measure of quality for quantum states.

## 2.1 The basic units of quantum computation: Qubits and operations

### Qubits

In quantum mechanics, particles or systems can be in a non-classical combination of classical states, where these states may have different properties such as total energy. This principle, called the *superposition* principle, is one of the cornerstones of quantum mechanics and also one of the properties that is extensively used in quantum computation. The basic unit of a quantum computer, called the qubit, is any (quantum mechanical) system that has two orthogonal states. This superposition or state can be viewed mathematically as a vector in  $\mathbb{C}^2$  where the vector is often written using the ket-notation:

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle = \begin{bmatrix} \alpha \\ \beta \end{bmatrix}, \quad (2.1)$$

with  $\alpha, \beta \in \mathbb{C}$ . Here  $|0\rangle$  and  $|1\rangle$  form a basis for  $\mathbb{C}^2$ ; they represent the two independent levels of the qubit. Furthermore, the qubit space is equipped with a standard (complex) dot product. The values of  $\alpha$  and  $\beta$  have a very important feature, that is best described using Born’s rule: Upon a measurement of the qubit in the  $\{|0\rangle, |1\rangle\}$  basis, the state of the qubit collapses to either the  $|0\rangle$  or  $|1\rangle$  state, with the probability  $\Pr(|0\rangle)$  ( $\Pr(|1\rangle)$ ) that it collapses to the  $|0\rangle$  ( $|1\rangle$ ) state dictated by  $\alpha$

and  $\beta$ :

$$\Pr(|0\rangle) = |\alpha|^2, \Pr(|1\rangle) = |\beta|^2. \quad (2.2)$$

Measurements will be discussed in greater detail in section 2.1.2. These probabilities lead to another important feature that the state must possess, namely that it needs to be normalized:  $|\alpha|^2 + |\beta|^2 = 1$ . In other words, the inner product of  $|\psi\rangle$  with itself ( $\langle\psi|\psi\rangle$ ) needs to be equal to 1. This makes the space of the qubit a Hilbert space, which we denote with  $\mathcal{H}$ .  $|0\rangle$  and  $|1\rangle$  are orthogonal under the inner product on  $\mathcal{H}$ :  $\langle 0|1\rangle = 0$ . Therefore they form an orthonormal basis of  $\mathcal{H}$ , called the *computational* or *canonical* basis.

Although the computational basis encompasses (usually) the two physically distinct levels of the qubit, the state of the qubit can be expressed using any two linearly independent vectors that form a basis for  $\mathcal{H}$ . Normally, two orthogonal vectors are used. Since all states should be normalized, orthonormality of every orthogonal basis follows. Apart from the computational basis, often used bases are the *Hadamard* basis  $\left\{ |+\rangle = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ 1 \end{bmatrix}, |-\rangle = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ -1 \end{bmatrix} \right\}$  and the basis  $\left\{ |+i\rangle = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ i \end{bmatrix}, |-i\rangle = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ -i \end{bmatrix} \right\}$ . If the basis of the qubit is not specified, the computational basis is assumed. All these states form the eigenstates to an important class of matrices called the *Pauli* matrices, which will be introduced in section 2.2.

## Multiple qubits

Multiple qubits together allow for more states. Namely, if there are  $n$  qubits in a system then there are  $2^n$  basis states for that system; all tensor products of the  $|0\rangle$  state and the  $|1\rangle$  state for each qubit separately. Very often, we write  $d$  as a shorthand for the system dimension  $2^n$ . Mathematically, the total state of the qubits is then a vector in a Hilbert space of dimension  $2^n$ :  $H_{2^n} = H^1 \otimes H^2 \dots \otimes H^n$ , with  $H^i$  the Hilbert space of the  $i^{\text{th}}$  qubit and  $\otimes$  the Kronecker tensor product. If two qubits are both in the state  $|0\rangle$ , then together they are in the 2-qubit state  $|0\rangle \otimes |0\rangle$ , often written as  $|00\rangle$ . For 2 qubits, this means that a general state can be written as:

$$|\psi\rangle_2 = \alpha_{00} |00\rangle + \alpha_{01} |01\rangle + \alpha_{10} |10\rangle + \alpha_{11} |11\rangle = \begin{bmatrix} \alpha_{00} \\ \alpha_{01} \\ \alpha_{10} \\ \alpha_{11} \end{bmatrix}, \quad (2.3)$$

with  $\alpha_{ij} \in \mathbb{C}$  and  $\sum_{ij} |\alpha_{ij}|^2 = 1$ . This extends to any number of qubits:

$$|\psi\rangle_n = \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_{2^n} \end{bmatrix}, \quad (2.4)$$

where still  $\alpha_i \in \mathbb{C}$  and  $\sum_{i=1}^{2^n} |\alpha_i|^2 = 1$ . This shows an important feature of qubits hinted at in the introduction: the number of parameters needed to describe the state grows *exponentially* with the number of qubits. A basis for the  $n$ -qubit Hilbert space  $\mathcal{H}_{2^n}$  can be built from any basis of  $\mathcal{H}_2$  as all possible  $n$ -fold Kronecker tensor products of the elements of that single-qubit basis. In this text, I will write the  $n$ -qubit Hilbert space  $\mathcal{H}_{2^n}$  as  $\mathcal{H}_n$  instead for brevity.

## Entanglement

If the state of multiple qubits cannot be written as a tensor product of single-qubit states, the qubits are said to be *entangled*. It is then not possible to describe the state of a single qubit without also describing the state of the other qubit, and thus the qubit states are now correlated. Any state that is not entangled is called *separable*. Quantum computations with separable states are

efficiently simulatable on a classical computer, indicating that entanglement is an important necessary of quantum computation<sup>1</sup>.

An example of an entangled state is the state  $|\Phi_+\rangle = \frac{|00\rangle + |11\rangle}{\sqrt{2}}$ ; the qubits are in an equal superposition of the  $|00\rangle$  and  $|11\rangle$  states. The first qubit is equally in the  $|0\rangle$  and  $|1\rangle$  state, but only when the second qubit is also in the  $|0\rangle$  and  $|1\rangle$  state, respectively.  $|\Phi_+\rangle$  is one of the 4 *Bell* states:

$$\begin{aligned} |\Phi_+\rangle &= \frac{|00\rangle + |11\rangle}{\sqrt{2}}, |\Phi_-\rangle = \frac{|00\rangle - |11\rangle}{\sqrt{2}}, \\ |\Psi_+\rangle &= \frac{|01\rangle + |10\rangle}{\sqrt{2}}, |\Psi_-\rangle = \frac{|01\rangle - |10\rangle}{\sqrt{2}}. \end{aligned} \quad (2.5)$$

Note that all Bell states are orthogonal to each other. Therefore, and because there are 4 of them, they form an orthonormal basis to  $\mathcal{H}_2$ .

Finally, for two qubits, a *maximally entangled state* can be defined as:

$$|\Psi_{\max}\rangle = \frac{1}{\sqrt{2}} \sum_i |\psi_i\rangle \otimes |\phi_i\rangle, \quad (2.6)$$

with the sum over  $a$  pair of bases  $\{|\psi_i\rangle\}$  and  $\{|\phi_i\rangle\}$  of the single-qubit Hilbert space. Note that all Bell states are maximally entangled. A maximally entangled state can be created for any even number of qubits, by summing over  $n$ -fold tensor products. Furthermore, I will denote the specific maximally entangled state where both bases are the canonical basis ( $\{|\psi_i\rangle\} = \{|\phi_i\rangle\} = \{|i\rangle\}$ ) as

$$|\Omega\rangle = \frac{1}{\sqrt{d}} \sum_i |i\rangle \otimes |i\rangle, \quad (2.7)$$

where I do not explicitly specify the number of qubits in both registers. Note that not every entangled state is also maximally entangled (e.g. the state  $\sqrt{\frac{3}{4}}|00\rangle + \sqrt{\frac{1}{4}}|11\rangle$  is not separable but also not maximally entangled).

### 2.1.1 Unitary operations on qubits

An operation on a qubit is a map  $O : \mathcal{H}_1 \rightarrow \mathcal{H}_1$ , meaning that the operation maps qubit states to other qubit states. Likewise, an operation on a set of  $n$  qubits is a map  $O : \mathcal{H}_n \rightarrow \mathcal{H}_n$ . An operation performed on a set of qubits is completely described by its action on a basis of the qubits Hilbert space. Normally, the operation maps a basis  $\{|\psi_k\rangle\}$  to a (possibly different) basis  $\{|\phi_k\rangle\}$  with  $k$  summing over all  $d$  elements in that basis:

$$U = \sum_k |\phi_k\rangle \langle \psi_k|. \quad (2.8)$$

For these operations we have,

$$U^\dagger U = \sum_k \sum_l |\psi_k\rangle \langle \phi_k | \phi_l \rangle \langle \psi_l| = \sum_k |\psi_k\rangle \langle \psi_k| = \mathbb{I}, \quad (2.9)$$

meaning that  $U$  is unitary. These operations on qubits are therefore also called *unitary operations* or also just *unitaries*<sup>2</sup>. Note that the operation  $U^\dagger$  maps the basis  $\{|\phi_k\rangle\}$  back to  $\{|\psi_k\rangle\}$ .

Using the vector notation, unitary operations can be viewed as unitary matrix transformations. With a qubit in the state  $|\psi_{\text{in}}\rangle$  and an output state,  $|\psi_{\text{out}}\rangle = U |\psi_{\text{in}}\rangle$ , they can be linked as a matrix transformation. If  $U$  maps  $\{|\psi_k\rangle\}$  to  $\{|\phi_k\rangle\}$ , then:

$$|\psi_{\text{out}}\rangle = \begin{bmatrix} \beta_1 \\ \beta_2 \\ \vdots \\ \beta_{2^n} \end{bmatrix} = \begin{bmatrix} u_{11} & u_{21} & \dots & u_{2^n 1} \\ u_{12} & u_{22} & \dots & u_{2^n 2} \\ \vdots & \vdots & \ddots & \vdots \\ u_{2^n 1} & u_{2^n 2} & \dots & u_{2^n 2^n} \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_{2^n} \end{bmatrix} = U |\psi_{\text{in}}\rangle, \quad (2.10)$$

<sup>1</sup>Entanglement is, however, not the sole origin of the power of quantum computation. Gottesman-Knill simulation [4] allows for efficient classical simulation of certain highly entangled states, something I have performed in my thesis but is not included in this report. See section 5.5 for a very brief introduction.

<sup>2</sup>In chapter 3, I will discuss operations on qubits that are not unitary. Normally speaking, non-unitary operations, besides measurement, correspond to errors.

with  $u_{ij} = \sum_k \langle i|\phi_k\rangle \langle \psi_k|j\rangle$ . Note that when the unitary  $U$  maps the computational basis to another basis  $\{|\phi_k\rangle\}$  (i.e.  $\{|\psi_k\rangle\}$  is the computational basis), the unitary  $U$  simplifies to (with  $\phi_i$  the column vector representing  $|\phi_i\rangle$  in the computational basis)  $U = [\phi_1 \ \phi_2 \ \dots \ \phi_{2^n}]$ .

Since any unitary matrix can be diagonalized, there are always  $2^n$  eigenvectors to  $U$ , with the corresponding eigenvalues lying on the unit circle, i.e.  $\lambda_i = e^{i\theta_i}$ . This means that, when an operation  $U$  acts on one of its eigenstates, it only introduces an overall phase  $e^{i\theta_i}$ . As we will see in section 2.1.2, such a *global* phase has no physical meaning. This has another implication: any  $U_1$  and  $U_2$  that only differ by a phase ( $U_1 = e^{i\theta}U_2$ ) do not have a physically distinct effect on a qubit. As such, the class of valid operations on a set of  $n$  qubits are exactly the matrices from the group  $SU(2^n)$ , the group of  $d \times d$  unitary matrices with determinant  $|U| = 1$ .

When performing an operation  $U_1$  on one qubit and an operation  $U_2$  on another qubit, the total operation on the combined system is the operation  $U_{12} = U_1 \otimes U_2$ . An important distinction to make for unitary operations acting on multiple qubits is whether that operation can be written as a tensor product of single-qubit operations or not. When this is not the case, the operation is commonly called an entangling operation. Finally, a unitary operation on a qubit is also called a *gate*. Qubits and unitary operations are, together with measurements, combined in quantum circuits to perform quantum computations. Quantum circuits will be briefly discussed in section 2.3, after a discussion on quantum measurements in section 2.1.2 and an overview of important unitary gates in section 2.2.

## 2.1.2 Measurements

The most straightforward measurement that can be performed on a qubit is a *projective* measurement. In a projective measurement, a qubit is projected onto a basis of its Hilbert space, mapping the state of the qubit to any of the elements from that basis after the measurement. The probability of obtaining a given result is dictated by Born's rule, introduced in section 2.1. After the measurement, the qubit state is said to have *collapsed* to either of the two basis states. Furthermore, the measurement outcome indicates which of the two it is; the outcome<sup>3</sup> will be +1 for  $|\psi_{+1}\rangle$  and -1 for  $|\psi_{-1}\rangle$ . The initial information encoded in the qubits' state (i.e.  $\alpha$  and  $\beta$ ) is lost, which is why it is said that measurements destroy quantum information. A projective measurement in the basis  $\{|\phi_{+1}\rangle, |\phi_{-1}\rangle\}$  can be denoted by the measurement observable  $M = |\psi_{+1}\rangle \langle \psi_{+1}| - |\psi_{-1}\rangle \langle \psi_{-1}|$ . It is then said that one performs a measurement of the observable on the qubit state.

The expectation value  $\lambda_{\psi,M}$  of an observable  $M$  on a qubit state  $|\psi\rangle$  is a value between 1 and -1; it is  $\lambda_{\psi,M} = \text{Pr}(+1) - \text{Pr}(-1)$  with  $\text{Pr}(+1)$  and  $\text{Pr}(-1)$  the probability that the state collapses to the +1 or -1 basis state, respectively. Using the observable  $M$ ,  $\lambda_{\psi,M}$  becomes (note that  $\lambda_{\psi,M}$  is always real-valued since  $M$  is Hermitian):

$$\lambda_{\psi,M} = \langle \psi | M | \psi \rangle = \text{tr} [M | \psi \rangle \langle \psi |]. \quad (2.11)$$

It is now evident that a global phase on the qubit can not influence the outcome of any measurement:

$$\lambda_{\phi} = \langle \phi | M | \phi \rangle = \langle \psi | e^{-i\theta} M e^{i\theta} | \psi \rangle = e^0 \langle \psi | M | \psi \rangle = \lambda_{\psi}. \quad (2.12)$$

When a qubit is entangled with another qubit, a measurement on the first qubit has implications for the state of the other qubit. If the first qubit of a Bell pair  $|\Phi_+\rangle = \frac{|00\rangle + |11\rangle}{\sqrt{2}}$  is measured in the computational basis, the state collapses to either the  $|00\rangle$  or  $|11\rangle$  state. This means that when a measurement in the computational basis on the second qubit is performed, the measurement outcome will always be either +1 or -1, indicated by the outcome of the first measurement. The outcomes of the two measurements are therefore perfectly correlated. This is a profound result of the principle of entanglement that quantum mechanics dictates. An important realisation is that the two entangled qubits need not necessarily be in the same physical location or even physically close; the entanglement of the two qubit states will persist if there is no interaction of the qubits with the outside world.

A more general measurement that can be made on qubits will be discussed in section 2.4.1.

---

<sup>3</sup>In principle, any two number can be the outcome of a measurement, but normally +1 and -1 are associated with the outcomes; then the measurement observable has eigenvalues +1 and -1.

## 2.2 Common unitary operations

### 2.2.1 Single-qubit gates

Unitary operations form the cornerstone to quantum computation; they allow for interaction with and manipulation of qubits in such a way that computations can be performed with them. There are myriad unitary operations - in principle uncountably many- but certain operations are more common than others. The gates introduced in this section will be the elements of most quantum circuits.

The identity gate  $I^4$ , also called the memory or wait gate, is the operation that maps every possible state of a set of qubits to itself:

$$I = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}. \quad (2.13)$$

The single qubit Pauli operators  $X$ ,  $Y$  and  $Z$  are the operations corresponding to the three *Pauli matrices*, named after the physicist Wolfgang Pauli:

$$\begin{aligned} I &= \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, & X &= \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \\ Y &= \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}, & Z &= \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}. \end{aligned} \quad (2.14)$$

I have included the identity operation  $I$  in the above equation, because these four matrices together form a group  $\mathcal{P}_1$  under multiplication if they are extended to have a phase of  $\pm 1, \pm i$ . This is because  $\forall P, P^2 = I$  and a product of two different Pauli matrices will result in the third one up to a phase, rendering the following equation for multiplication of Paulis:

$$P_a P_b = \delta_{ab} I + i \epsilon_{abc} P_c, \quad (2.15)$$

with  $\delta_{ab}$  the Kronecker delta and  $\epsilon_{abc}$  the Levi-Civita symbol. Apart from that, the four elements of  $\mathcal{P}_1$  span the single-qubit operator space. Therefore, any single-qubit operation can be decomposed into a linear combination of the identity operator and the Paulis. Furthermore, all Pauli matrices have trace 0. Note that the determinant of the Paulis are not 1 and that therefore they are not elements of  $SU(2)$ . Still, they are normally chosen over their counterparts  $iX, iY, iZ \in SU(2)$  because they are Hermitian.

$X$  is also known as the bit flip operation, because  $X|0\rangle = |1\rangle$  and  $X|1\rangle = |0\rangle$ . Equivalently,  $Z$  is also known as the phase flip operation, because  $Z|0\rangle = |0\rangle$  and  $Z|1\rangle = -|1\rangle$ , introducing a phase on the  $|1\rangle$  state compared to the  $|0\rangle$  state. Since  $Y = iXZ$ , it performs both the bit and phase flip.

All single-qubit Paulis have a  $+1$  eigenvalue and a  $-1$  eigenvalue. The corresponding eigenstates are exactly the 6 states introduced in section 2.1: the  $(|+\rangle, |-\rangle)$ ,  $(|+i\rangle, |-i\rangle)$  and  $(|0\rangle, |1\rangle)$  states are the  $+1$  and  $-1$  eigenstates of the  $X$ ,  $Y$  and  $Z$  matrices, respectively; they can be found in table 2.1. The Hadamard basis is therefore also called the  $X$  basis, and the basis  $\{|+1\rangle, |-i\rangle\}$  is equivalently called the  $Y$  basis.

The Hadamard gate  $H$  is the operation that maps the computational basis to the Hadamard basis:

$$H = |+\rangle \langle 0| + |-\rangle \langle 1|, \quad (2.16)$$

in matrix notation (and the decomposition into the Paulis) this becomes

$$H = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} = \frac{1}{\sqrt{2}} (X + Z). \quad (2.17)$$

Since the Hadamard matrix is a Hermitian matrix, it is its own inverse and therefore idempotent:  $HH = \mathbb{I}$ . This means that an equivalent representation of  $H$  is:

$$H = |0\rangle \langle +| + |1\rangle \langle -|, \quad (2.18)$$

---

<sup>4</sup>In this report, I make a distinction between the identity operation and the identity matrix. With  $I$  I mean the operation that acts on qubits, whereas with  $\mathbb{I}$  I specifically mean the identity matrix.

$P \in \mathcal{P}_1$	+1 eigenstate	-1 eigenstate
$I$	all states	-
$X$	$ +\rangle = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ 1 \end{bmatrix}$	$ -\rangle = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ -1 \end{bmatrix}$
$Y$	$ +i\rangle = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ i \end{bmatrix}$	$ -i\rangle = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ -i \end{bmatrix}$
$Z$	$ 0\rangle = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$	$ 1\rangle = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$

**Table 2.1:** Eigenstates of the single qubit Paulis and the identity operator. Together, these 4 operations form the group  $\mathcal{P}_1$ .

and that therefore the Hadamard gate also maps the Hadamard basis back to the computational basis. This behaviour is inherent to every idempotent unitary operation. Finally, the Hadamard matrix shows an underlying symmetry of qubit states: if any unitary gate  $U$  gives has as action  $U(\alpha|0\rangle + \beta|1\rangle) = \alpha'|0\rangle + \beta'|1\rangle$ , then  $HUH(\alpha|+\rangle + \beta|-\rangle) = \alpha'|+\rangle + \beta'|-\rangle$ . This remains true if  $U$  is instead defined for the Hadamard basis.

The single-qubit  $Z$  matrix adds a phase of  $e^{i\pi} = -1$  to the  $|1\rangle$  state while doing nothing to the  $|0\rangle$  state. This operation can be generalized to an arbitrary phase, resulting in the  $R_z(\theta)$  operation, that adds a phase of  $e^{i\theta}$  to the  $|1\rangle$  state.

Similarly, the  $X$  and  $Y$  operations add a phase of  $e^{i\pi}$  to the  $|-\rangle$  and  $|-i\rangle$  states, respectively, while leaving the other state of their respective eigenbasis unaffected. These too can be generalized to an arbitrary phase, resulting in the  $R_x(\theta)$  and  $R_y(\theta)$  operations, that add a phase of  $e^{i\theta}$  to the  $|-\rangle$  and  $|-i\rangle$ , respectively. In matrix form, they are:

$$\begin{aligned}
R_x(\theta) &= \begin{bmatrix} \cos(\frac{\theta}{2}) & -i \sin(\frac{\theta}{2}) \\ -i \sin(\frac{\theta}{2}) & \cos(\frac{\theta}{2}) \end{bmatrix}, \\
R_y(\theta) &= \begin{bmatrix} \cos(\frac{\theta}{2}) & -\sin(\frac{\theta}{2}) \\ \sin(\frac{\theta}{2}) & \cos(\frac{\theta}{2}) \end{bmatrix}, \\
R_z(\theta) &= \begin{bmatrix} e^{-i\frac{\theta}{2}} & 0 \\ 0 & e^{+i\frac{\theta}{2}} \end{bmatrix}.
\end{aligned} \tag{2.19}$$

Note that these matrices are defined up to a global phase, and that in this form they indeed do add a global phase to the eigenstates of their respective Paulis. Furthermore, note that indeed  $R_i(\pi) = P_i$  (up to this global phase) as expected. Finally, just as with the  $X$  and  $Z$ ,  $R_x(\theta) = HR_z(\theta)H$ . An interpretation of the rotation gates (and the origin of their names) can be found in the Bloch sphere visualization of quantum states, see section 2.5. The  $R_z(\theta)$  gate has specific names for  $\theta = \frac{\pi}{4}$  and  $\theta = \frac{\pi}{8}$ :  $S = R_z(\frac{\pi}{4})$  is called the  $S$ - or phase-gate, and  $T = R_z(\frac{\pi}{8})$  is called the  $T$ -gate.

## 2.2.2 Multi-qubit gates

A very important specific multi-qubit gate is the (2-qubit) *controlled-not*<sup>5</sup>,  $CNOT$  or  $CX$  gate: it performs an  $X$  gate on the second qubit (called the *target* qubit), conditioned on the first qubit (the *control* qubit) being in the  $|1\rangle$  state:

$$CX = |0\rangle\langle 0| \otimes I + |1\rangle\langle 1| \otimes X \tag{2.20}$$

The action of the  $CX$  on the 4 states from the canonical basis  $\mathcal{H}_2$  can be found in table 2.2; its action on any other state can then be calculated using linearity by decomposing that state into the canonical basis. Especially interesting is the action of the  $CX$  gate on two qubits whenever the first qubit is in a superposition of  $|0\rangle$  and  $|1\rangle$ , while the second qubit is in the  $|0\rangle$  state:

$$\begin{aligned}
CX(\alpha|0\rangle + \beta|1\rangle) \otimes |0\rangle &= \alpha CX|0\rangle \otimes |0\rangle + \beta CX|1\rangle \otimes |0\rangle \\
&= \alpha|0\rangle \otimes |0\rangle + \beta|1\rangle \otimes |1\rangle = \alpha|00\rangle + \beta|11\rangle
\end{aligned} \tag{2.21}$$

<sup>5</sup>The name controlled-not comes from classical computing: the classical bit flip operation is also called the NOT gate, and the controlled bit-flip operation is also called the controlled-not.

Input basis state $ \psi\rangle$	$CX \psi\rangle$	$SWAP \psi\rangle$
$ 00\rangle$	$ 00\rangle$	$ 00\rangle$
$ 01\rangle$	$ 01\rangle$	$ 10\rangle$
$ 10\rangle$	$ 11\rangle$	$ 01\rangle$
$ 11\rangle$	$ 10\rangle$	$ 11\rangle$

**Table 2.2:** The action of the CX gate and the SWAP gate on all 4 states in the canonical basis of  $\mathcal{H}_2$ . Note that for CX gate the 2<sup>nd</sup> qubit is only flipped when the first qubit is in the  $|1\rangle$  state, and that for the SWAP gate only the states  $|01\rangle$  and  $|10\rangle$  are swapped.

Note that when  $\alpha = \beta = \frac{1}{\sqrt{2}}$  the initial state on the first qubit is  $|+\rangle$ , the resulting state is the Bell state  $|\Phi_+\rangle$ . This 2-qubit gate can thus create entanglement on a pair of qubits. In matrix notation, the CX gate is:

$$CX = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}. \quad (2.22)$$

Conversely, the *CPHASE* or *CZ* gate is the 2-qubit gate that performs the phase-flip or *Z* gate on the second qubit, conditioned on the first qubit being in the  $|1\rangle$  state. It is a counterpart to the CX gate, and the two are linked by rotating the second qubit to the Hadamard basis.

The 2-qubit SWAP operation swaps the states of 2 qubits:

$$SWAP = |00\rangle\langle 00| + |10\rangle\langle 01| + |01\rangle\langle 10| + |11\rangle\langle 11|. \quad (2.23)$$

If two qubits are in the separable state  $(\alpha|0\rangle + \beta|1\rangle) \otimes (\gamma|0\rangle + \delta|1\rangle)$ , then after applying the SWAP gate the resulting state is  $(\gamma|0\rangle + \delta|1\rangle) \otimes (\alpha|0\rangle + \beta|1\rangle)$ . This extends to any entangled 2-qubit state:

$$SWAP(\alpha_{00}|00\rangle + \alpha_{01}|01\rangle + \alpha_{10}|10\rangle + \alpha_{11}|11\rangle) = \alpha_{00}|00\rangle + \alpha_{10}|01\rangle + \alpha_{01}|10\rangle + \alpha_{11}|11\rangle. \quad (2.24)$$

Note that only  $\alpha_{01}$  and  $\alpha_{10}$  have switched; this is because switching qubits does not alter the  $|00\rangle$  and  $|11\rangle$  states. The action of the SWAP gate on the 4 states from the canonical basis  $\mathcal{H}_2$  can be found in table 2.2; again its action on any other state can then be calculated using linearity by decomposing that state into the canonical basis. In matrix notation, the SWAP gate is:

$$SWAP = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad (2.25)$$

### 2.2.3 A generating set for operations

In principle, to perform all possible qubit operations, one would need to perform every element of the continuous group  $SU(2)$ . There are uncountably many elements in that group, but every element can be written<sup>6</sup> as a multiplication of a generating set of a discrete group that is dense in the continuous group. Therefore, if you are able to perform the gates in this set on the qubits, any other operation can be performed by applying the specific combination of the generating set. Such a generating set is also named a universal set of quantum gates.

A straightforward example of a universal set of 2-qubit gates is the Hadamard gate  $H$ , the  $T$  gate and the CNOT gate. The process of writing an operation as the product of a generating set is known as *compiling*. An example is that the  $X$  gate can be compiled to  $HT^4H$ ; therefore the bit-flip operation can be implemented by first applying the Hadamard operation, then 4 times the  $T$  gate, and then again the Hadamard operation.

<sup>6</sup>To be precise, every element can only be approximated by a finite set of gates. This is because the set of gates  $SU(2)$  is uncountably infinite, whereas the set of products of elements in a finite set is countable. An important theorem, the Solovay-Kitaev theorem[5], states that this approximation can be done efficiently for all gates.

## 2.2.4 The generalized multi-qubit Paulis

The Paulis can be extended to  $n$ -qubit unitaries by the Kronecker tensor product:  $P_n = P_1^1 \otimes \dots \otimes P_1^n$ ; they are also denoted as  $P_1 P_2 \dots P_n$ . These multi-qubit operators are generally called the  $n$ -qubit Paulis or also simply Paulis.  $\text{tr}[A \otimes B] = \text{tr}[A] \text{tr}[B]$ , so  $\text{tr}(P_n)$  is 0 for all Paulis that are not tensor products of only the single qubit  $I$ . Furthermore, they still have only  $+1$  and  $-1$  eigenstates. If all  $n$ -qubit Paulis are extended with an overall phase of  $\pm 1, \pm i$ , then together they form a group, the Pauli group  $\mathcal{P}_n$ . I will denote an element of  $\mathcal{P}_n$  with  $P_n$  or simply  $P$ .

For a  $P \in \mathcal{P}_1$ , there is just one  $+1$  and  $-1$  eigenstate; for a general  $n$ -qubit (non-identity) Pauli there are for both the  $+1$  and  $-1$  eigenspace  $2^{n-1}$  vectors in its basis. Let Pauli  $P_n = P_1^1 \otimes \dots \otimes P_1^i \otimes \dots \otimes P_1^n$  be any Pauli in  $\mathcal{P}_n$ ,  $|\lambda_i^1\rangle$  be the eigenstates of the single-qubit Pauli  $P_1^i$  and  $\lambda_i^1$  its corresponding eigenvalue. Then we can write any of the eigenstates  $|\lambda_j^n\rangle$  of  $P_n$  as:

$$|\lambda_j^n\rangle = \bigotimes_i^n |\lambda_i^1\rangle, \quad (2.26)$$

with a corresponding eigenvalue  $\lambda_j^n = \prod_i \lambda_i^1$ . So, any eigenstate of a  $n$ -qubit Pauli  $P_n$  is a tensor product of the eigenstates of the 1-qubit Paulis of which  $P_n$  is a tensor product. The eigenvalue of a corresponding eigenstate of  $P_n$  is then the product of all eigenvalues of the individual 1-qubit eigenstates.

The Pauli operators are also linked to the Bell pairs via the single qubit operations on one of the two qubits. This also allows the definition of the generalized Bell states:

$$|B_n\rangle = (P_n \otimes I) |\Omega\rangle. \quad (2.27)$$

These states together form a new basis for  $\mathcal{H}_n$ , which we will use occasionally throughout this report. Note that for  $\mathcal{P}_1$  this indeed generates the four Bell states.

## 2.2.5 The weight of an operator

The *weight*  $\text{wt}(P)$  of a Pauli operator  $P$  is the number of qubits on which it acts non-trivially. If the  $n$ -qubit Pauli  $P$  can be decomposed into tensor products of single-qubit Paulis with  $r$  identity gates, then  $\text{wt}(P) = n - r$ . E.g. the operator  $O = X \otimes I \otimes X$  has  $\text{wt}(O) = 2$ , whereas the operator  $Q = Z \otimes Y \otimes X$  has  $\text{wt}(Q) = 3$ . Furthermore, the *support* of an operator is the set of qubits on which it acts non-trivially.

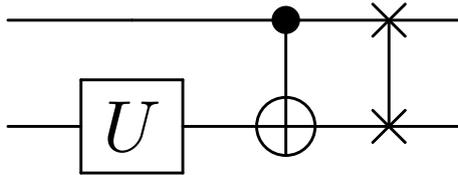
In the framework of quantum error correcting codes, which will be discussed in chapter 5, it is useful to define the weight of a non-Pauli operator in terms of its decomposition into the Paulis. If an operator  $A$  can be decomposed into Paulis of at most weight  $k$ , then  $\text{wt}(A) = k$  as well. So, the operator  $CX = \frac{1}{2}(II + ZI + IX - ZX)$  has  $\text{wt}(CX) = 2$ , but the 3-qubit gate  $S_{1,3} = \frac{1}{2}(III + XIX + YIY + ZIZ)$ , that swaps the state of the 1st and 3rd qubit, also has  $\text{wt}(S_{1,3}) = 2$ . Note that the weight of an operator is not preserved after composition. For instance, the Paulis  $P_1 = XY$  and  $P_2 = ZY$  both have  $\text{wt}(P_1) = \text{wt}(P_2) = 2$ , but their product  $P_1 P_2 = YI$  (up to a global phase) has weight 1.

## 2.3 Quantum circuits

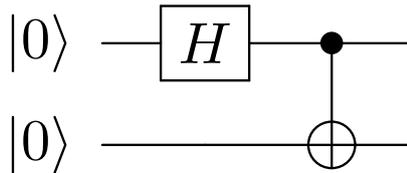
A quantum circuit is a set of qubits (often called a register) and a set of ordered operations. It is a representation of a composite operation that is to be performed on the set of qubits. The operations are applied to the register in the specified order, which can be understood as running the circuit. Any computation done on a quantum computer is usually represented as a quantum circuit (For an overview of many quantum algorithms, see [1]). Normally, the qubits are initialized in a specific state, meaning that this state is prepared on the qubits before the circuit runs. Often the qubits are all initialized in the  $|0\rangle$  state, so that any non-trivial preparation of the qubits is included as operations in the circuit. Moreover, usually the qubits are measured at the end of the circuit, although measurements can in principle take place anywhere in the circuit. Any element of the operations within the circuit can

be referred to as a *location*. If a qubit is ‘waiting’ for operations on other qubits to be done, this is referred to as a wait-location, which can be represented by performing the identity gate on that qubit.

A quantum circuit has an often-used visual representation, called the circuit diagram. Every qubit in the register is drawn as a horizontal line, and every operation in the circuit is drawn as a rectangle on the lines of the supported qubits; within that rectangle is then written that operation. The CNOT operation has a specific representation in a diagram as a black dot on the control qubit, a vertical line from that qubit to the target qubit and the modulo-2 addition sign  $\oplus$  on the target qubit. Moreover, the SWAP operation is drawn as a cross on both supported qubits with a vertical line connecting them. These representations can be seen in figure 2.1. Furthermore, a very basic quantum circuit representing the preparation of the Bell state  $|\Phi_+\rangle$  can be found in figure 2.2. For a more in-depth discussion on quantum circuits and how they are drawn I refer to [6].



**Figure 2.1:** The visual representation of a single qubit gate  $U$  on the bottom qubit (left), the CNOT operation with the control qubit at the top (middle) and the SWAP operation (right).



**Figure 2.2:** A simple quantum circuit that prepares the Bell state  $|\Phi_+\rangle = \frac{|00\rangle + |11\rangle}{\sqrt{2}}$ . The two qubits, both initialized in the  $|0\rangle$  state, are represented by the two horizontal lines. The Hadamard operation is performed on the top qubit, after which the CNOT operation is performed with the top qubit as the control qubit (represented by the black dot) and the bottom qubit as the target qubit (represented by the modulo-2 addition sign  $\oplus$ ).

## 2.4 Density matrices

If an entangled pair of qubits  $\frac{|00\rangle + |11\rangle}{\sqrt{2}}$  is shared by two people, named Alice and Bob, neither can describe the state of their own qubit without knowing the state of the other qubit. When Alice performs a measurement in the computational basis, Alice knows that Bob’s qubit collapses to the  $|0\rangle$  or  $|1\rangle$  state, depending on her measurement outcome. As long as Bob doesn’t know the measurement outcome - or does not even know whether a measurement was performed by Alice, Bob can describe the state of his qubit as a *statistical mixture* of the  $|0\rangle$  and  $|1\rangle$  states; it is described using a *density operator*  $\rho$ , in this case  $\rho = \frac{1}{2}(|0\rangle\langle 0| + |1\rangle\langle 1|)$ .

More generally, if the state of a qubit is  $|\psi_j\rangle$  with probability  $p_j$ , the density operator or density matrix describing that state is:

$$\rho = \sum_j p_j |\psi_j\rangle\langle \psi_j|, \quad (2.28)$$

with  $\sum_j p_j = 1$ . Therefore,  $\text{tr}(\rho) = 1$ . Note that the different states  $|\psi_j\rangle$  need not necessarily be orthogonal to each other.

Any density operator whose sum is over just one element is called a *pure* state, whereas if the sum is over more than one element it is called a *mixed* state, and  $\rho^2 = \rho$  only for pure states. A density matrix is not an element of  $H_n$ , but an element of a set of operators  $\{M \in \mathcal{M}_{2^n \times 2^n} | M \geq 0 \ \& \ \text{tr}(M) = 1\}$  that acts upon elements of  $H_n$ . Note that this is not the same space as the space of operations  $SU(2)$ ; a density operator is never a unitary matrix.

When a qubit is in a mixed state of all elements of a basis  $\{|\psi_i\rangle\}$  of  $H_n$ , with equal probability, the resulting density operator  $\rho_{\text{mm}}$  represents the *maximally mixed* state:

$$\rho_{\text{mm}} = \sum_{\{|\psi_i\rangle\}} \frac{1}{d} |\psi_i\rangle \langle \psi_i| = \frac{\mathbb{I}}{d}. \quad (2.29)$$

If a qubit is in the maximally mixed state then no information that might have initially been encoded is left on the qubit; it is then essentially a random bit.

A density matrix can be decomposed into a weighted sum<sup>7</sup> of an orthonormal basis of  $\mathcal{M}$ . Most notably, a  $n$ -qubit density matrix can be decomposed into a linear combination of the elements  $P_i \in \mathcal{P}_n$ .

## 2.4.1 Operations on density matrices

### Unitary operations

A unitary operation  $U$  on a pure state  $|\psi\rangle$  is denoted by  $U|\psi\rangle$ ; as such unitary operation on a pure density matrix is  $U|\psi\rangle\langle\psi|U^\dagger$ . This can be easily generalized to any density matrix. Applying a unitary operation  $U$  on a qubit in the density state  $\rho_{\text{in}}$  gives a state  $\rho_{\text{out}}$ :

$$\rho_{\text{out}} = U\rho_{\text{in}}U^\dagger. \quad (2.30)$$

As will be introduced in chapter 3, more general operations can be defined for density matrices; exactly these operations can make mixed states from pure states.

### Measurements

Performing a projective measurement of an observable  $M$  on a qubit on a pure state gives an expectation value  $\lambda_{\psi,M} = \langle\psi|M|\psi\rangle = \text{tr}[M|\psi\rangle\langle\psi|]$ . This can be extended to any density state  $\rho$ :

$$\lambda_{\rho,M} = \text{tr}[M\rho]. \quad (2.31)$$

A generalization of projective measurements can be made, where the qubit state is not projected upon a set of orthogonal states but on a set of orthogonal subspaces that together span the whole Hilbert space. It does not necessarily have to be a projection; for a set of operators  $\{A_i\}$ , the qubit state will collapse with probability  $\text{tr}[A_i\rho A_i^\dagger]$  to any of the states

$$\rho_{A_i} = \frac{A_i\rho A_i^\dagger}{\text{tr}[A_i\rho A_i^\dagger]}. \quad (2.32)$$

Furthermore, we need  $\sum_i A_i^\dagger A_i = \mathbb{I}$ , so that if the measurement outcome is not recorded, the mixed state  $\sum_i A_i\rho A_i^\dagger$  will be obtained. Such a generalized measurement is called a POVM or positive-operator valued measure.

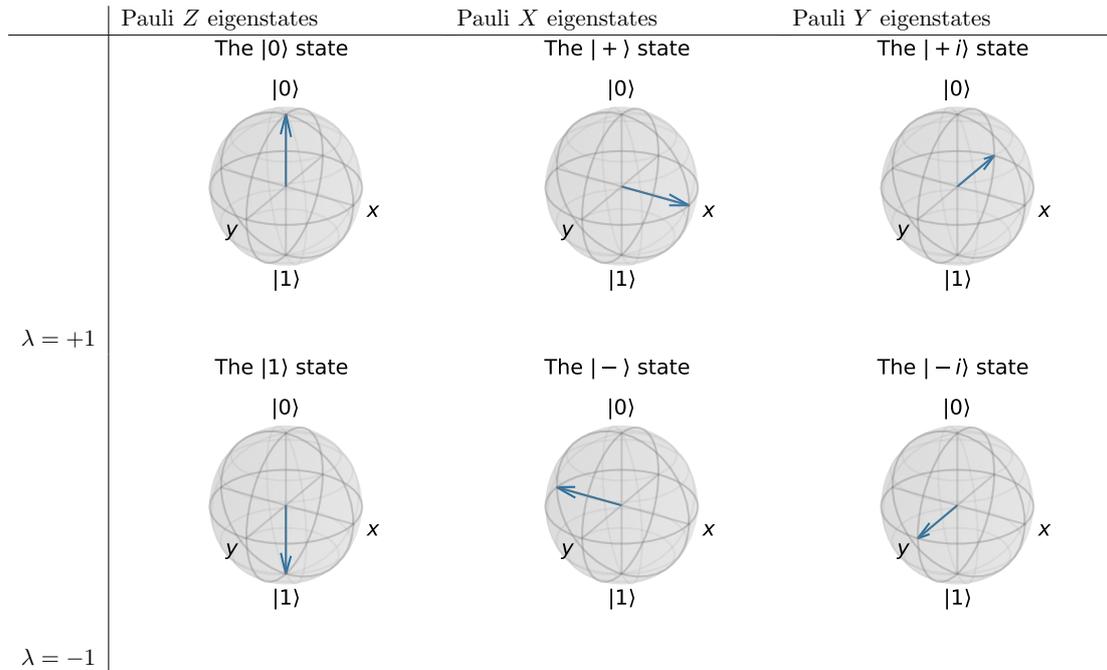
## 2.5 The Bloch sphere

Visualizing a state of a qubit can be done using the Bloch sphere. The Bloch sphere is a sphere of unit radius, for which pure states are points on that sphere. The  $|0\rangle$  state is the north pole of the sphere, whereas the  $|1\rangle$  state is the south pole of the sphere. Any superposition of the two states is a point on the surface of the sphere; it can be specified (up to a global phase) using the parameters  $\theta$  and  $\phi$  ( $0 \leq \theta \leq \pi$  and  $0 \leq \phi \leq 2\pi$ ) as

$$|\theta, \phi\rangle = \cos \frac{\theta}{2} |0\rangle + e^{i\phi} \sin \frac{\theta}{2} |1\rangle. \quad (2.33)$$

---

<sup>7</sup>This is not necessarily a convex combination. Note for instance that the pure state  $|1\rangle\langle 1| = \frac{1}{2}(I - Z)$



**Figure 2.3:** The  $+1$  (top) and  $-1$  (bottom) eigenstates for the Pauli matrices drawn on the Bloch sphere. Any pure state is represented by a point on the Bloch sphere. The  $|0\rangle$  and  $|1\rangle$  states are on the north- and south pole, respectively. Any superposition is a point in between. The azimuthal and polar angle are  $\theta$  and  $\phi$  from Eq. (2.33) respectively.

By using spherical coordinates, the corresponding point on the Bloch sphere to  $|\theta, \phi\rangle$  is then exactly the point with radius 1, azimuthal angle  $\theta$  and polar angle  $\phi$ .  $\phi$  is the relative phase of the qubit state; it is often just called the phase.

The eigenstates of the Pauli  $X$ ,  $Y$  and  $Z$  matrices are exactly the intersections of the Bloch sphere with the  $x$ -,  $y$ - and  $z$ -axes; the  $+1$  and  $-1$  eigenstates lie on the positive and negative ends respectively.

Unitary operations on a qubit can be represented as rotations on the Bloch sphere; this reflects the idea that any unitary operation can be viewed as a change of basis. The  $R_x$ ,  $R_y$  and  $R_z$  are all rotations over an angle  $\theta$  about the  $x$ -,  $y$ - and  $z$ -axes respectively. As such, the Pauli operations  $X$ ,  $Y$  and  $Z$  are the operations that rotate any state on the Bloch sphere half of a rotation around their respective axis. Note that it follows that any eigenstate of a unitary operation should lie exactly on the axis of rotation and that therefore eigenstates of unitary operations always are antipodal on the Bloch sphere.

Since density matrices are convex combinations of pure states, they can be viewed as points lying in the Bloch sphere. The vector  $\mathbf{v}$  that points to the mixed state is calculated as

$$v_i = \text{tr} [P_i \rho], \quad (2.34)$$

with  $\{i\} \in \{x, y, z\}$ . Therefore, a statistical mixture of the  $Z$ -eigenstates is a point on the  $z$ -axis, and the maximally mixed state is the point at the origin.

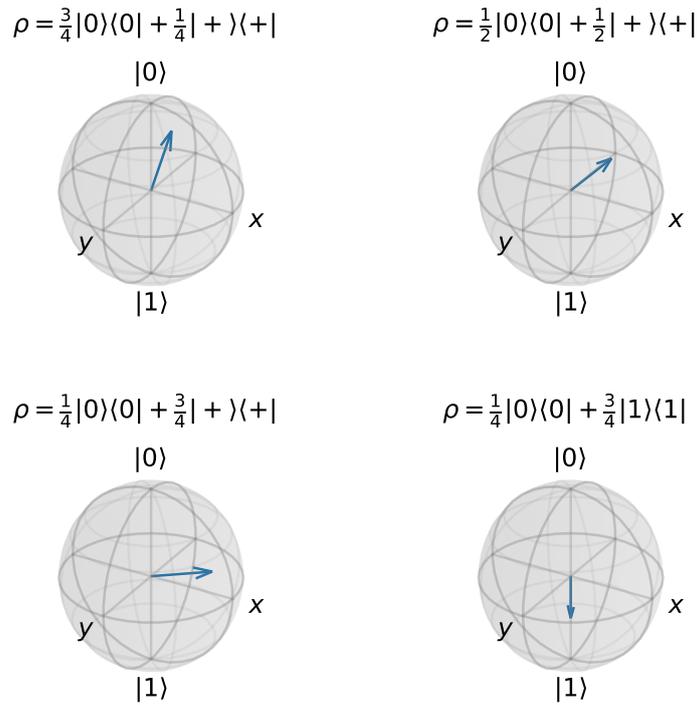
## 2.6 State fidelity

Comparing two quantum states can be done by calculating their state fidelity, which is a measure of the overlap of two quantum states. For two states  $\rho$  and  $\sigma$ , the fidelity is

$$F(\rho, \sigma) = \text{tr} \left( \sqrt{\sqrt{\rho} \sigma \sqrt{\rho}} \right)^2 = \text{tr} \left( \sqrt{\sqrt{\sigma} \rho \sqrt{\sigma}} \right)^2, \quad (2.35)$$

with the last identity indicating that the fidelity is symmetric. If one of the states is pure, e.g. if  $\rho = |\psi\rangle\langle\psi|$ , the fidelity simplifies to

$$F(\rho, \sigma) = \langle\psi|\sigma|\psi\rangle. \quad (2.36)$$



**Figure 2.4:** Various mixed states represented in the Bloch sphere. The decomposition of  $\rho$  into the weighted sum of pure states determines the vector that represents the mixed state. Note that the states do not lie on, but rather within the Bloch sphere.

Furthermore, if  $\sigma = |\phi\rangle\langle\phi|$  is also pure, the fidelity simplifies further to  $F(\rho, \sigma) = |\langle\psi|\phi\rangle|^2$ . The fidelity is bounded by 0 from below and 1 from above; hence it is also often expressed as a percentage.

The state fidelity is commonly used as a measure of the quality of a preparation apparatus, entangling apparatus or any other experimental state; the fidelity is then calculated between the obtained (often mixed) state, and the ideal or desired (often pure) state. Finally, there is also a fidelity for operations, the process fidelity, which will be introduced in section 3.7.

## Chapter 3

# Quantum channels and open quantum systems

In a perfect world pure quantum states would always stay pure quantum states, and there would not be any use for density matrices. In practice, however, many processes influence pure quantum states in such a way that the formalism of pure quantum states, called *closed quantum systems*, is not adequate. In general, a (possibly mixed) quantum state  $\rho_{in}$  can undergo a unitary transformation, a non-unitary transformation, an error process or a coupling with its surroundings (often called the environment of the system), among other things. Generally speaking, this will produce a density matrix  $\rho_{out}$ , and  $\rho_{in}$  is said to have gone through the *quantum system* or *quantum channel*. Due to the possible coupling with its surroundings, this is called an *open quantum system*. Quantum Channels can be characterized as linear maps, where they map density matrices to density matrices.

Not any linear map can be a quantum channel; there are some constraints and restrictions put on them in the framework of open quantum systems, which are introduced in section 3.1. There exist various representations of the maps  $\Lambda$ , of which some are introduced in section 3.2. Some common channels are introduced in section 3.4, and section 3.5 does the same for error channels specifically. Finally, two tools that help in analysing error maps are introduced at the end of the chapter: section 3.6 introduces an approximation of error maps called the *Pauli Twirl*, and section 3.7 introduces a measure of how good a process is when it is compared to the ideal or desired operation, that is the equivalent of state fidelities for processes.

### 3.1 Channels as linear maps

If we have a quantum system  $\Lambda$ , with an input state  $\rho_{in}$  of dimension  $n \times n$  and an output state  $\rho_{out} = \Lambda(\rho_{in})$  of dimension  $m \times m$ , we can view this system or channel as a linear map from the space of  $n \times n$  hermitian matrices to the space of  $m \times m$  Hermitian matrices:  $\Lambda : \mathcal{M}_{n \times n} \rightarrow \mathcal{M}_{m \times m}$ . Note that this map is not necessarily an isomorphism, however for every unitary operation this is the case.

#### 3.1.1 Restrictions: *Complete Positivity & Trace preservation*

Since both the input and output state are physical density matrices, there are extra constraints on  $\Lambda$ . There are two important constraints imposed; complete positivity and trace preservation. Any physical density matrix should be positive-semidefinite (i.e. should have only non-negative eigenvalues). This should hold for any (possibly mixed) state put through  $\Lambda$ , but also if only part of a state is put through the system. Thus, if we have an extended density matrix space  $\mathcal{M}_{n_1 \times n_1} \otimes \mathcal{M}_{n_2 \times n_2}$  and for the map  $(\Lambda \otimes I_{n_2}) : \mathcal{M}_{n_1 \times n_1} \otimes \mathcal{M}_{n_2 \times n_2} \rightarrow \mathcal{M}_{m_1 \times m_1} \otimes \mathcal{M}_{n_2 \times n_2}$  we should have  $(\Lambda \otimes I_{n_2})(\rho_{ext,in}) = \rho_{ext,out} \geq 0$ , for any extended  $\rho_{ext}$ . It can be shown [7] that it is enough to show this only for  $n_2 = n_1$ . Any map that has this property is said to be *completely positive*. The extension of  $\mathcal{M}_{n_1 \times n_1}$  to  $\mathcal{M}_{n_1 \times n_1} \otimes \mathcal{M}_{n_2 \times n_2}$  can be viewed as coupling of the system space to an environment. Thus, the first subspace is the system space, and the second subspace is the environment.

Furthermore, any physical state  $\rho$  should have unit trace [6] for normalization. Thus, the condition  $\text{tr}(\Lambda(\rho)) = \text{tr}(\rho)\forall\rho$  must hold. This is exactly the criterion for  $\Lambda$  to be *trace preserving*.

Any map that is completely positive (CP) and trace preserving (TP) is called a CPTP-map. Furthermore, it is often so that  $m = n$  and therefore that the map  $\Lambda$  maps  $\mathcal{M}_{n \times n}$  to itself. This will be the case in this text unless otherwise noted.

## 3.2 Representations of $\Lambda$

There exist various equivalent representations of a channel  $\Lambda$ . In this section I state various representations and their conditions for CP and TP. All proofs for CP and TP are in appendix A.1 and A.2 respectively.

### 3.2.1 The $\chi$ matrix

Let  $\{B_i\}$  be any orthogonal basis for  $\mathcal{M}_{d \times d}$ , s.t.  $\langle B_i, B_j \rangle = d\delta_{ij}$ , with  $d = 2^n$ . The map  $\Lambda$  can always be written in the form:

$$\Lambda(\rho) = \sum_{m,n=0}^{d^2-1} \chi_{m,n} B_m \rho B_n^\dagger. \quad (3.1)$$

Here the  $d^2 \times d^2$  matrix  $\chi$  is called the *process matrix* of  $\Lambda$ .  $\chi$ , together with the basis  $\{B_i\}$ , completely characterises the map  $\Lambda$ . Very often the Pauli basis is used for  $\{B_i\}$ ; the representation of  $\Lambda$  then simplifies somewhat, since that basis is unitary. We will use the Pauli basis unless otherwise specified.

If a system acts on a set of  $n$  qubits with a process matrix  $\chi_1$ , and on another set of  $n'$  qubits with a process matrix  $\chi_2$ , a representation of the full system working on all  $n + n'$  qubits is  $\chi_1 \otimes \chi_2$ , in the basis  $\{B_{1,i}\} \otimes \{B_{2,i}\}$ . If the Pauli basis is used for both  $\chi_1$  and  $\chi_2$ , the new basis becomes exactly the  $\mathcal{P}_{n+n'}$  group.

### Complete Positivity and Trace Preservation

Any channel  $\Lambda$  is completely positive if and only if its  $\chi$  matrix is positive-semidefinite, implying all non-negative real eigenvalues and Hermiticity for  $\chi$ . For  $\Lambda$  to be TP, it is necessary and sufficient for  $\chi$  to obey the following equality:

$$\sum_{m,n} \chi_{m,n} B_n^\dagger B_m = I. \quad (3.2)$$

As a direct consequence,  $\chi$  always has trace 1, but this is not a strong enough statement for trace preservation. There are  $d^2$  constraints coming from Eq. (3.2), which is consistent with the number of constraints for the Choi matrix (sect. 3.2.2) to be TP.

### Calculating $\chi$ matrix of a unitary process

Any unitary process  $U$  can be decomposed into a sum of the Pauli matrices:  $U = \sum_{P_k \in \mathcal{P}_n} p_k P_k$ , with  $p_k = \langle P_k, U \rangle = \text{tr}[P_k^\dagger U]$ . Since a map of a unitary operation  $U$  can be written as  $\Lambda(\rho) = U\rho U^\dagger = \sum_{P_m, P_n \in \mathcal{P}_n} p_m p_n^* P_m \rho P_n^\dagger$ , it is evident that for a unitary operation  $\chi = |U\rangle_{\mathcal{P}} \langle U|_{\mathcal{P}}$ , with  $|U\rangle_{\mathcal{P}}$  the vector of all weights of the Pauli decomposition of  $U$ , so  $\chi$  has rank 1.

### 3.2.2 Choi matrix

For a map  $\Lambda$  acting on  $n$  qubits, the Choi matrix  $\rho_{\text{Choi}}$  is the density matrix obtained after putting half of the maximally entangled state  $|\Omega\rangle$  through the channel  $\Lambda$ , while doing nothing on the other half. Let  $d = 2^n$  be the system dimension. Then:

$$\rho_{\text{Choi}} = (\Lambda \otimes I)(|\Omega\rangle \langle \Omega|) = \sum_{i,j} \frac{1}{d} \Lambda(|i\rangle \langle j|) \otimes |i\rangle \langle j|. \quad (3.3)$$

The Choi matrix characterizes  $\Lambda$  completely; this is a result of the Choi-Jamiolkowski isomorphism [8], that forms the basis of the channel-state duality. It states that there is a link between the space of CP maps and the space of density operators. There is a straightforward interpretation to the Choi matrix; it is a block matrix with at the  $(i, j)$ -th block the matrix  $\Lambda(|i\rangle\langle j|)$ . By linearity, the action of the map on any state  $\rho_{in}$  can be calculated from these elements  $\Lambda(|i\rangle\langle j|)$ , thereby completely specifying the map.

### Complete Positivity and Trace Preservation

Since  $\rho_{\text{Choi}}$  is a density matrix, it is positive-semidefinite. This is also a sufficient condition for the corresponding  $\Lambda$  to be completely positive, based on Choi's theorem on completely positive maps [7].

The condition for  $\Lambda$  to be trace preserving is that if one traces out the system of  $\rho_{\text{Choi}}$  one should obtain the maximally mixed state:

$$\text{tr}_1 [\rho_{\text{Choi}}] = \frac{1}{d}\mathbb{I}. \quad (3.4)$$

Again, there are  $d^2$  constraints for TP coming from Eq. (3.4), consistent with Eq. (3.2) for the  $\chi$  matrix. Thus,  $\rho_{\text{Choi}}$  always has trace 1, but this is not a strong enough statement for trace preservation.

The extra constraint of trace preservation on the map  $\Lambda$  also has consequences for the Choi-Jamiolkowski isomorphism. The isomorphism still holds, but it is now between the space of CPTP maps and a subspace of the space of density operators: the space of density operators for which constraint (3.4) holds.

### 3.2.3 Kraus decomposition

Any linear map  $\Lambda(\rho)$  can be written as  $\Lambda(\rho) = \sum_k L_k \rho R_k^\dagger$  for some set of operators  $\{L_k\}$  and  $\{R_k\}$  by summing over enough  $k^1$ . The Kraus representation of a map is when  $\{L_k\} = \{R_k\}$  (these are then called the Kraus operators  $\{A_k\}$ ):

$$\Lambda(\rho) = \sum_i A_i \rho A_i^\dagger. \quad (3.5)$$

A Kraus representation of a map is only possible if that map is CP. Therefore, complete positivity is inherent to a Kraus representation. The condition on the Kraus operators for the corresponding  $\Lambda$  to be trace preserving is  $\sum_k A_k^\dagger A_k = I$ . As such, if there is only one Kraus operator, that operator is necessarily unitary for a trace preserving map.

The Kraus- or operator-sum representation is useful for the following way: for every  $A_k$  there is a probability  $\text{tr}[A_k \rho A_k^\dagger]$  that it will happen, and therefore the resulting state is a statistical mixture of all the resulting states  $A_k \rho A_k^\dagger$ . Any channel  $\Lambda$  can be represented by a Kraus representation with at most  $d^2$  Kraus operators. Note the similarities with a POVM.

However, the set of operators  $\{A_k\}$  is not unique; there are always multiple sets of operators that represent the same channel. Any two sets of Kraus operators  $\{A_k\}$  and  $\{B_k\}$  that correspond to the same system  $\Lambda$  are linked via a unitary matrix  $U$ :  $B_i = \sum_j U_{ij} A_j$  [6].

### 3.2.4 Superoperator representation

The superoperator representation is essentially a vectorized version of the  $\chi$  matrix, or the  $\chi$  matrix in a different basis. Using the identity  $|ABC\rangle\rangle = (C^T \otimes A)|B\rangle\rangle$ , a vectorized version of any  $B_m \rho_{in} B_n^\dagger$  can be seen as a matrix product on the vectorized version of  $\rho_{in}$ :  $(B_n^{\dagger T} \otimes B_m)|\rho_{in}\rangle\rangle$ . This allows for summation of all individual elements of the  $\chi$  matrix to obtain one matrix  $S \in \mathcal{M}_{d^2 \times d^2}$  that completely characterizes  $\Lambda$ :

$$|\rho_{out}\rangle\rangle = S|\rho_{in}\rangle\rangle = \left( \sum_{m,n} \chi_{m,n} \overline{B_n} \otimes B_m \right) |\rho_{in}\rangle\rangle. \quad (3.6)$$

<sup>1</sup>This can be achieved by using the set of all  $d^4$  combinations  $L_k, R_k = B_i, B_j$  for  $\{B_i\}$  the canonical basis of  $\mathcal{M}_d$ , up to a scalar multiplication. Then  $|\Lambda(\rho)\rangle\rangle = \sum_k (R_k^T \otimes L_k)|\rho\rangle\rangle \hat{=} \sum_{mn,ij} A_{mn,ij} \rho_{ij}$ , so any linear combination of the elements of  $\rho$  can be written like this.

This representation allows for easy calculation of composite maps, because  $\Lambda_2(\Lambda_1(\rho))$  can now (in vectorized form) be written as  $S_2 S_1 |\rho\rangle\rangle$ ; the action of multiple maps can thus also be represented by a single matrix  $S = S_2 S_1$ .

The complete positivity and trace preservation conditions are less intuitive and well defined for the superoperator than they are for other representations of  $\Lambda$ . To check complete positiveness or trace preservation, it is easier to map back to another representation and check the CPTP constraints in that representation. This representation is not very intuitive overall, but it is very useful to find a representation of composite maps. As such, it is best used as an intermediate and to convert to a different representation in the end.

### 3.3 Relations between different representations

Although all representations are in principle equivalent, some representations are better suited than others for certain situations. There are relations in between all representations, such that every representation can be obtained. Sometimes it is easier to use an intermediate representation to go from one to another. The proofs of the following relations can be found in appendix A.3.

#### 3.3.1 The Choi matrix and $\Lambda(\rho)$

The actual outcome  $\rho_{out}$  after applying a map to a density matrix can be obtained from the Choi matrix using the following relation:

$$\Lambda(\rho) = d \text{tr}_2 [\rho_{\text{Choi}}(I \otimes \rho^T)]. \quad (3.7)$$

This has little experimental merit, it is useful however when the outcome of a map is to be calculated when the only characterization of the map that is known is its Choi matrix. To calculate the Choi matrix from  $\Lambda$  itself is straightforward using the definition of the Choi matrix.

#### 3.3.2 The Choi matrix and $\chi$ matrix

The Choi matrix can be calculated as the  $\chi$  matrix in the basis  $\{|B_i\rangle\rangle\} = \{(B_i \otimes I) |\Omega\rangle\rangle\}$ . Thus:

$$\rho_{\text{Choi}} = \sum_{m,n} \chi_{m,n} |B_m\rangle\rangle \langle\langle B_n|. \quad (3.8)$$

This equation shows that, in principle, the Choi matrix is actually the  $\chi$  matrix in a specific basis  $\{|B_i\rangle\rangle\}$ .

To obtain the coefficients of  $\chi$  from the Choi matrix is straightforward:

$$\chi_{m,n} = \langle\langle B_m | \rho_{\text{Choi}} | B_n \rangle\rangle. \quad (3.9)$$

When working with the Pauli basis, the Choi matrix is thus the  $\chi$  matrix in the Bell basis.

#### 3.3.3 The $\chi$ matrix and the Kraus decomposition

To calculate the  $\chi$  matrix from the Kraus representation is straightforward; the Kraus operators need only be decomposed into the desired basis  $\{B_m\}$  using the Hilbert-Schmidt inner product. After summing over all Kraus operators, the desired representation is obtained:

$$\chi_{m,n} = \sum_k \langle B_m, A_k \rangle \langle A_k, B_n \rangle = \sum_k \text{tr} [B_m^\dagger A_k] \text{tr} [B_n^\dagger A_k]^* \quad (3.10)$$

A Kraus representation of  $\Lambda$  can be obtained from the  $\chi$  matrix using its eigendecomposition.  $\chi$  (with a specified basis  $\{B_m\}$ ) can always be diagonalised, because it is Hermitian. Each eigenvector  $|\psi_i\rangle$  weighed by its corresponding eigenvalue  $\lambda_i$  then represents a Kraus operator  $A_i$ , with the weights of the decomposition of  $A_i$  into  $\{B_m\}$  the elements of  $|\psi_i\rangle$ :  $A_i = \sqrt{\lambda_i} \sum_j \langle j, \psi_i \rangle B_j$ . Thus, the number of Kraus operators is directly linked with the number of non-zero eigenvalues of  $\chi$ . Since the eigenvalues of a matrix do not depend on the basis in which it is expressed, the Kraus operators do not depend which basis  $\{B_m\}$  is chosen.

### 3.3.4 The Choi matrix and the Kraus decomposition

The Choi matrix can be calculated from the Kraus operators using the definition of the Choi matrix:

$$\rho_{\text{Choi}} = \frac{1}{d} \sum_{i,j,k} A_k |i\rangle \langle j| A_k^\dagger \otimes |i\rangle \langle j|. \quad (3.11)$$

Obtaining a set of Kraus operators from the Choi matrix can be done by diagonalizing  $\rho_{\text{Choi}}$ . From its eigenvalues and eigenvectors  $\lambda_k$  and  $|\psi_k\rangle$  one can obtain a set of Kraus operators obeying  $|A_k\rangle\rangle = \sqrt{\lambda_k} |\psi_k\rangle$ . Furthermore, these Kraus operators are all orthogonal to each other:  $\text{tr}[A_i^\dagger A_j] = \sqrt{\lambda_i \lambda_j} \delta_{ij}$ .

### 3.3.5 The $\chi$ matrix and the superoperator

As mentioned, using the identity  $|ABC\rangle\rangle = (C^T \otimes A)|B\rangle\rangle$  the superoperator matrix  $S$  can be calculated from  $\chi$ :

$$S = \sum_{m,n} \chi_{m,n} \overline{B_n} \otimes B_m. \quad (3.12)$$

Obtaining the  $\chi$  matrix from  $S$  can be done using the Hilbert-Schmidt inner product:

$$\chi_{m,n} = \text{tr}[(\overline{B_n} \otimes B_m)^\dagger S]. \quad (3.13)$$

When using the superoperator representation to calculate the total product of multiple maps, special care needs to be taken when going back to the  $\chi$  (or any other) representation. Using multiple superoperator-matrices after each other on a state does preserve the trace, but only if a non-normalized basis as introduced in 3.2.1. However, if a different basis is used, this results only in a scalar offset, so the  $\chi$  matrix can be normalised to obtain the proper state.

## 3.4 Common channels

Here I will introduce some channels that are very common. When determining representations of maps  $\Lambda$  they play an important role, and they are applicable in many other representations.

### 3.4.1 Unitary channels

A unitary operation on a density matrix  $\rho$  can be seen as a map  $\Lambda(\rho) = U\rho U^\dagger$ . This is essentially a Kraus representation of  $\Lambda$ , with only one Kraus operator. Therefore, both the  $\chi$  and the Choi matrix only have 1 non-zero eigenvalue, which is equal to 1.

Calculating the  $\chi$  matrix of a unitary process works essentially the same when mapping a Kraus representation to a  $\chi$  representation of the same system. Since there is now only one Kraus operator, the process is straightforward. First, calculate the support of  $U$  on every element of the desired basis  $\{B_m\}$ , and put those in a vector  $|u\rangle$ . That is,  $\langle i, u \rangle = \text{tr}[B_i^\dagger U]$ . Then, the  $\chi$  matrix of  $U$  in that basis is  $\chi_U = |u\rangle \langle u|$ .

Calculating the Choi matrix for a unitary operation works much like calculating the  $\chi$  matrix, but here  $U$  itself can be vectorized immediately to obtain  $|U\rangle\rangle$ . This reflects the idea that the Choi matrix is the  $\chi$  matrix in a specific basis. So, for a unitary  $U$ , the Choi matrix is of the form  $\frac{1}{d} \sum_{i,j} U |i\rangle \langle j| U^\dagger \otimes |i\rangle \langle j|$ . This leads to  $\rho_{\text{Choi}} = |U\rangle\rangle \langle\langle U|$ , with  $|U\rangle\rangle$  the vectorized version of  $U$ .

### 3.4.2 Pauli channels

Any CPTP channel of the form  $\Lambda(\rho) = \sum_{P_k \in \mathcal{P}_n} p_k P_k \rho P_k$  (with  $\sum_k p_k = 1$ ) is a Pauli channel; all its Kraus operators are Pauli operators, each with their own specific weight  $p_k$ . The corresponding  $\chi$  matrix in the Pauli basis is a diagonal matrix, with  $\chi_{k,k} = p_k$ .

A Pauli channel is often easy to work with, because it has concrete distinct operations on  $\rho$  just as with a Kraus representation, with the added qualities that those operations are the well-understood

Pauli operations, and that its representation in those operators is unique. Furthermore, Pauli channels are classically efficiently simulatable<sup>2</sup>.

The Choi matrix of a Pauli channel also has a nice interpretation. Since states  $|P_k\rangle = (P_k \otimes I) |\Omega\rangle$  (with  $P_k \in \mathcal{P}_n$ ) are all the different generalized Bell states, the Choi matrix for a Pauli channel becomes a statistical mixture of Bell states:  $\rho_{\text{Choi}} = \sum_k p_k |P_k\rangle \langle P_k|$ .

### 3.4.3 Partial trace as a channel

Taking the partial trace of a system can also be viewed as a channel. Tracing out the  $i$ th qubit of an  $n$ -qubit system has the following Kraus operators:

$$A_k = I^{\otimes i-1} \otimes \langle k| \otimes I^{\otimes n-i}, \quad (3.14)$$

with  $I^{\otimes i-1}$  the identity matrix tensor itself  $i - 1$  times, and  $\{|k\rangle\}$  is the computational basis for the  $i$ th qubit. Note that these Kraus operators are not square, indicating that this channel is a linear map from the operator space  $\mathcal{M}_{d \times d}$  to a smaller operator space  $\mathcal{M}_{(d-1) \times (d-1)}$ . Tracing out more qubits can be done in succession or by replacing the  $I$  operators in the Kraus operators by  $\langle k|$  for those specific qubits.

## 3.5 Common error channels

A typical error in a quantum circuit arises when it is influenced by an environment. Therefore, errors are hard to describe in the formalism of closed quantum systems, but within open quantum systems they can be described easily; an error on a quantum system can be viewed as a channel acting on that system. Two common errors that happen on qubits are the *dephasing* and *depolarizing* errors. Both these channels are Pauli channels. Another common channel is the amplitude damping channel, which is however not a Pauli channel.

### 3.5.1 Systematic errors

An implementation  $U_{\text{act}}$  of a unitary operation might always differ from the desired operation  $U_{\text{ideal}}$  by a constant operation  $U_{\text{err}}$ . Such an error is called a systematic error:

$$U_{\text{act}} = U_{\text{err}} U_{\text{ideal}}, \quad (3.15)$$

and normally the error is small:  $U_{\text{err}} \approx I$ . Systematic errors are usually an indication of a poorly calibrated device or gate set.

### 3.5.2 Dephasing channel

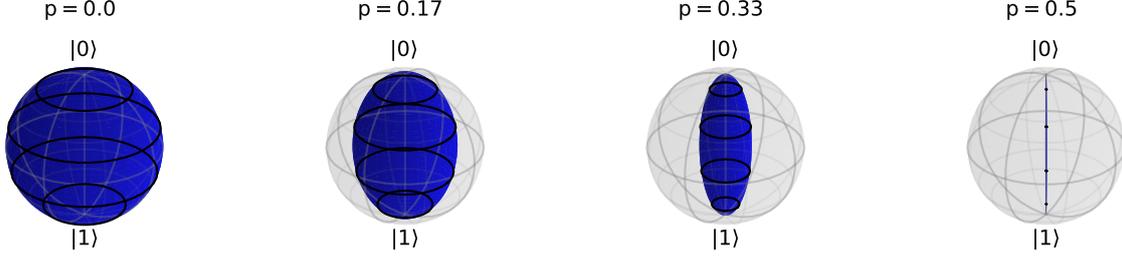
When there is a probability  $p$  that the (relative) phase of a qubit flips, the phase becomes gradually less defined. Such a process is a *dephasing* error, and it can be described as a channel with a Kraus decomposition of two operators:  $A_1 = \sqrt{1-p}I$  and  $A_2 = \sqrt{p}Z$ . The channel acting on a qubit in the state  $\rho = \begin{bmatrix} a & b \\ b^* & d \end{bmatrix}$  yields:

$$\Lambda_{\text{deph}(p)}\left(\begin{bmatrix} a & b \\ b^* & d \end{bmatrix}\right) = (1-p) \begin{bmatrix} a & b \\ b^* & d \end{bmatrix} + p \begin{bmatrix} a & -b \\ -b^* & d \end{bmatrix} = \begin{bmatrix} a & (1-2p)b \\ (1-2p)b^* & d \end{bmatrix}. \quad (3.16)$$

Thus, the dephasing channel brings the off-diagonal elements gradually down; when  $p = \frac{1}{2}$ , the channel  $\Lambda_{\text{deph}(\frac{1}{2})}$  is known as the complete dephasing channel. Since pure states that are superpositions have off-diagonal elements, whereas purely statistical mixtures have no off-diagonal elements, the complete dephasing channel destroys any coherence that exists between the two basis states of the qubit. The complete dephasing channel destroys any coherent quantum information, and as such the

<sup>2</sup>This can be done by performing *Gottesman-Knill* simulation, see section 5.5 for more details.

dephasing channel *decoheres* the qubit. This can be viewed on the Bloch sphere as a process that gradually maps the full Bloch sphere (and any coherent state on it) to an ellipsoid centered around the Z axis; the complete dephasing channel then projects every state on the Bloch sphere to the Z axis.



**Figure 3.1:** Action of the dephasing channel on the entire Bloch ball for  $\{p = 0, \frac{1}{6}, \frac{1}{3}, \frac{1}{2}\}$ . With probability  $p$  the relative phase of the qubit is flipped, resulting in a loss of coherence between the  $|0\rangle$  and  $|1\rangle$  state, thereby mapping the qubit state to the  $Z$ -axis in the Bloch sphere. For  $p = \frac{1}{2}$  the relative phase is completely randomized, resulting in a projection onto the  $Z$ -axis. If the initial qubit state was  $|0\rangle$  or  $|1\rangle$  there is no effect from the channel, but if the qubit state was a superposition then the coherent information is gradually lost.

A qubit may undergo a map that decoheres the qubit steadily; the off-diagonal terms are going to zero as  $e^{-t/T_2}$ . Here,  $T_2$  is the rate of qubit decoherence. The gradual decay of coherence can then be viewed as a process that every  $\frac{1}{T_2}$  unit time the qubit can decohere completely, obtaining a statistical mixture on the qubit. There exist a relation of the  $T_2$  time with the  $T_1$  relaxation time, see section 3.5.4.

Furthermore, if the Pauli basis is used, the  $\chi$  matrix for the dephasing channel is a diagonal matrix:  $\chi_{I,I} = 1 - p$ ,  $\chi_{Z,Z} = p$ , and all other terms are 0. When two qubits both undergo a dephasing process separately, the resulting map can be characterized by a Kraus representation where the Kraus operators are all possible tensor products of the 1-qubit operators:  $A_1 = (1 - p)II$ ,  $A_2 = \sqrt{p(1 - p)}IZ$ ,  $A_3 = \sqrt{p(1 - p)}ZI$  &  $A_4 = pZZ$ . This generalizes to more qubits: the set of Kraus operators for  $n$  qubits is then  $\{A\}^{\otimes n}$ , resulting in  $2^n$  Kraus operators.

### 3.5.3 Depolarizing channel

In the depolarizing channel, the Pauli  $X, Y$  and  $Z$  operators are all applied to the state with equal probability  $\frac{p}{3}$ , while the state can also be left intact. For one qubit, The Kraus operators are:

$$\{A_k\} = \{\sqrt{1 - p}I, \sqrt{\frac{p}{3}}X, \sqrt{\frac{p}{3}}Y, \sqrt{\frac{p}{3}}Z\}. \quad (3.17)$$

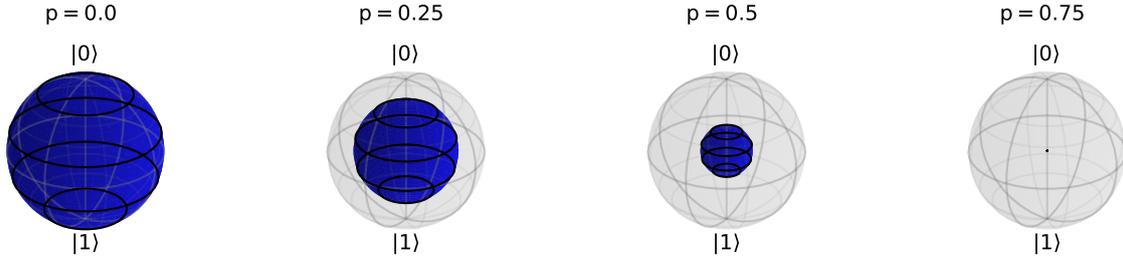
$\Lambda_{\text{dep}(p)}$  acting on a qubit in the state  $\rho = \begin{bmatrix} a & b \\ b^* & d \end{bmatrix}$  yields:

$$\Lambda_{\text{dep}(p)}\left(\begin{bmatrix} a & b \\ b^* & d \end{bmatrix}\right) = \begin{bmatrix} (1 - \frac{2p}{3})a + \frac{2p}{3}d & (1 - \frac{4p}{3})b \\ (1 - \frac{4p}{3})b^* & (1 - \frac{2p}{3})d + \frac{2p}{3}a \end{bmatrix}. \quad (3.18)$$

Since  $\text{tr}(\rho) = 1$ , this can be rewritten as:

$$\Lambda_{\text{dep}(p)}(\rho) = (1 - \frac{4p}{3})\rho + \frac{4p}{3} \frac{I}{2}. \quad (3.19)$$

The depolarizing channel takes a convex combination of  $\rho$  with the maximally mixed state, regardless of the initial state  $\rho$ . In the Bloch sphere this can be viewed as gradually going down to the origin; essentially deflating the Bloch sphere. For  $p = \frac{3}{4}$ , we get the completely depolarizing channel;  $\Lambda_{\text{dep}(\frac{3}{4})}(\rho) = \frac{1}{2}I$  for any input state  $\rho$ . The qubit is arbitrarily rotated in any direction, all with equal probability, and none of the initial information remains. The Bloch sphere is then completely mapped to the origin.



**Figure 3.2:** Action of the depolarizing channel on the entire Bloch ball for  $p = \{0, \frac{1}{4}, \frac{2}{4}, \frac{3}{4}\}$ . A  $R_X(\pi)$ ,  $R_Y(\pi)$  or  $R_Z(\pi)$  rotation is applied to the qubit with equal probability  $\frac{p}{3}$ . Therefore, the qubit state is generally randomized, resulting in a mapping to the maximally mixed state in the origin of the Bloch sphere. For  $p = \frac{3}{4}$ , the qubit state is in an equal mixture of rotations over all axes, destroying any information initially encoded on the qubit, and thus obtaining the maximally mixed state.

When using the Pauli basis, the  $\chi$  matrix is again a diagonal matrix, with  $\chi_{I,I} = 1 - p$  and  $\chi_{X,X} = \chi_{Y,Y} = \chi_{Z,Z} = \frac{p}{3}$ .

A 2-qubit depolarizing channel can also be defined, with Kraus operators  $A_1 = \sqrt{1-p}I$  and all other elements of  $\mathcal{P}_2$  with equal weight  $\sqrt{\frac{p}{15}}$ , for a total of 16 Kraus operators. This depolarizing channel can be used in simulations as an error map for 2-qubit unitary operations.

### 3.5.4 Amplitude damping channel

The amplitude damping channel gradually maps a qubit to the  $|0\rangle$  state. It is a useful representation of a qubit for which the  $|0\rangle$  state is the ground state and the  $|1\rangle$  state is an excited state. The qubit can go from the excited state to the ground state, called relaxation; the inverse is far less probable and is not modelled here.

There are 2 Kraus operators for this channel:

$$\begin{aligned}
 A_1 &= \begin{bmatrix} 1 & 0 \\ 0 & \sqrt{1-p} \end{bmatrix} = \frac{1 + \sqrt{1-p}}{2}I + \frac{1 - \sqrt{1-p}}{2}Z, \\
 A_2 &= \begin{bmatrix} 0 & \sqrt{p} \\ 0 & 0 \end{bmatrix} = \sqrt{p}(X + iY).
 \end{aligned} \tag{3.20}$$

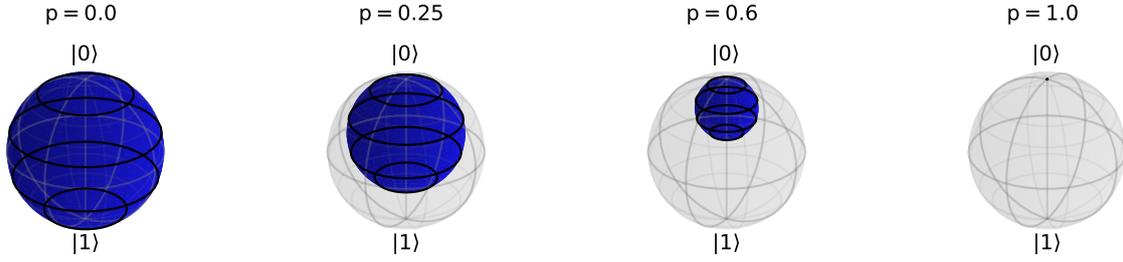
$A_2$  is the lowering operator, that maps the  $|1\rangle$  state to the  $|0\rangle$  state.  $A_1$  can be interpreted as the operator that corresponds to nothing happening to the qubit. If after some time the qubit has *not* decayed, it means that the qubit was more likely to be in the  $|0\rangle$  state to begin with, resulting in  $\sqrt{1-p}$  as the lower right element. Note that  $A_1$  is also essential to uphold the trace-preservation condition  $\sum_k A_k^\dagger A_k = \mathbb{I}$ .

$\Lambda_{\text{amp}(p)}$  acting on a qubit in the state  $\rho = \begin{bmatrix} a & b \\ b^* & d \end{bmatrix}$  yields:

$$\Lambda_{\text{amp}(p)} \left( \begin{bmatrix} a & b \\ b^* & d \end{bmatrix} \right) = \begin{bmatrix} a + pd & \sqrt{1-p}b \\ \sqrt{1-p}b^* & (1-p)d \end{bmatrix}. \tag{3.21}$$

In the Bloch sphere representation, amplitude damping also deflates the Bloch sphere. However, the  $|0\rangle$  is left untouched, so the north pole of the Bloch sphere (instead of the origin as with the depolarizing channel) stays fixed as it deflates, see figure 3.3. For  $p = 1$ , we get the complete amplitude damping channel, and the entire Bloch sphere is mapped to the  $|0\rangle$  state. When expressed in the Pauli basis, the  $\chi$  matrix corresponding to the amplitude-damping channel is not diagonal; but the coefficients can be calculated from the decomposition of  $A_1$  and  $A_2$  into the Paulis.

The characteristic time that is associated with amplitude damping is known as the *relaxation* time  $T_1$ , corresponding to the decoherence time  $T_2$ . If a qubit state undergoes relaxation, it also gradually loses its coherence. Therefore,  $T_2$  is bounded by  $T_1$ :  $T_2 \leq 2T_1$  [6].



**Figure 3.3:** Action of the amplitude damping channel on the entire Bloch ball for  $p = \{0, \frac{1}{4}, \frac{3}{5}, 1\}$ . The qubit state is gradually mapped to the  $|0\rangle$  state; the  $|1\rangle$  state is flipped to the  $|0\rangle$  state with probability  $p$ , whereas the inverse does not happen. This can be interpreted as the qubit relaxing to its ground state  $|0\rangle$ . For  $p = 1$  the qubit is completely relaxed, obtaining the  $|0\rangle$  state. Note that this model is not completely physically motivated; a qubit with finite temperature will generally relax to a statistical mixture of its ground- and excited state, and not only to its ground state.

The amplitude damping model described here is not necessarily very physical: a qubit will generally decay to a statistical distribution on the  $|0\rangle$  and  $|1\rangle$  state, with the distribution on those two states determined by the temperature of the qubit. Generally, there will be a non-zero occupation of the  $|1\rangle$  state after complete damping. To model this, the raising operator must also be included as a Kraus operator; this allows for excitation of the qubit from the  $|0\rangle$  to the  $|1\rangle$  state. Generally speaking, the amplitude damping channel (with 2 Kraus operators) models a qubit with temperature  $T = 0$ , whereas the generalized model (with 3 Kraus operators) models a qubit with a finite temperature.

### 3.6 Pauli Twirl approximation

An approximation of a map  $\Lambda$  can be generated by taking the Pauli twirl operation. The Pauli twirl generates a map  $\bar{\Lambda}$ :

$$\bar{\Lambda}(\rho) = \frac{1}{d^2} \sum_m P_m^\dagger \Lambda(P_m \rho P_m^\dagger) P_m, \quad (3.22)$$

with  $\{P_m\} = \mathcal{P}_n$ .

When  $\Lambda$  is represented by its  $\chi$  matrix in the Pauli basis, the Pauli twirl of  $\Lambda$  is exactly taking the diagonal of  $\chi$  and setting all other elements to 0. This operation gives a simple way of obtaining an approximation of  $\Lambda$  which has only Pauli elements as its Kraus operators, generating a Pauli channel.

#### 3.6.1 Error Weight ratio

A possible error  $E$  on a quantum state is a  $t$ -qubit error if it is a linear combination of operators  $E_i$  all with weight  $wt(E_i) \leq t$ . These individual  $E_i$ 's do not necessarily have to have their support on the same qubits; in principle a  $t$ -qubit error can act non-trivially on all qubits of a system. Furthermore, if the Kraus operators of an error channel are all  $t$ -qubit errors themselves, the channel is a  $t$ -qubit error channel.

1-qubit errors are generally easier to correct than multi-qubit errors. Therefore, the ratio of multi-qubit errors to single-qubit errors is an important value. Even if a certain error channel is a multi-qubit channel, if all but one of its Kraus operators are 1-qubit errors, and if the probability of that particular multi-qubit error happening is lower, that channel is at least ostensibly less severe than a general multi-qubit error channel.

The Pauli twirl gives an Pauli channel approximation  $\bar{\Lambda}$  of any  $\Lambda$ , with a corresponding  $\bar{\chi} = \text{diag}(p_1, p_2, \dots, p_{d^2})$ . A measure of the amount of single-qubit and multi-qubit errors of the map is the sum of all eigenvalues of  $\bar{\chi}$  that correspond to the weight-1 and weight-2 operators. So, with  $\mathcal{P}_s : \{P \in \mathcal{P} | wt(P_i) = 1\}$  and  $\mathcal{P}_m : \{P \in \mathcal{P} | wt(P_i) > 1\}$ ,

$$s_0 = p_I, \quad s_1 = \sum_{P_k \in \mathcal{P}_s} p_k, \quad s_{2+} = \sum_{P_k \in \mathcal{P}_m} p_k. \quad (3.23)$$

The ratio of multi-qubit errors to single-qubit errors can then be calculated as  $r = \frac{m}{s}$ . Note that  $r$  does not indicate any information of how much error there is in total, and furthermore that  $s + m \leq 1$ , where there is only an equality if  $p_I = 0$ , something that will generally not happen often, especially not when error channels are concerned.

### 3.7 Process fidelity

A measure that is used to test the quality of a map is the *process fidelity*  $F_p$ . If there is a desired unitary process represented by  $\chi_{\text{ideal}}$ , then the process fidelity of a system or channel represented by  $\chi$  is:

$$F_p = \text{tr} [\chi_{\text{ideal}}\chi] = \text{tr} [\chi\chi_{\text{ideal}}] = \text{tr} [\rho_{\text{Choi}}\rho_{\text{Choi,ideal}}] \quad (3.24)$$

Note here that the product  $\chi_{\text{ideal}}\chi$  is indeed the matrix product. The last identity follows from the fact that the Choi matrix is the  $\chi$  matrix in a specific basis. The process fidelity is thus the same as the state fidelity for the Choi matrix.

The desired operation is a unitary operation  $U$  and its process matrix  $\chi_{\text{ideal}}$  can be diagonalized resulting in only one non-zero eigenvalue (see section 3.4.1). The process fidelity then becomes:

$$F_p = \langle u | \chi | u \rangle \quad (3.25)$$

with  $|u\rangle$  as defined in section 3.4.1:  $|u\rangle$  is that vector for which  $\langle i, u \rangle = \text{tr} [B_i^\dagger U]$ .

Another used quality measure is the *channel fidelity*. For a channel with a Choi matrix  $\rho_{\text{Choi}}$ , with a desired operation  $U$ , the channel fidelity is:

$$F_c = \langle \Omega | (U^\dagger \otimes I) \rho_{\text{Choi}} (U \otimes I) | \Omega \rangle \quad (3.26)$$

Using the Choi matrix to calculate  $F_p$  and making use of Eq. (3.11), it is easy to see that for unitary processes the process fidelity and channel fidelity are the same.

To analyse or characterize a system, or specifically the errors on a system, a representation of that system that characterizes it as a map  $\Lambda$  must be obtained. This can be done by performing quantum process tomography, which will be introduced in chapter 3.

## Chapter 4

# Quantum process tomography

When designing or implementing a quantum circuit, there is a specific operation meant to be performed. However, imperfections in the implementation can give rise to imperfections on that operation. Without knowing what the system does, adequately performing any task with the system is difficult indeed. Being able to analyse or characterize a system is therefore an important tool. Sometimes it is enough to only characterize a specific output of a system (or any other quantum state). To fully characterize any quantum state its density matrix needs to be determined. When many copies of the state can be prepared, these copies can be used to measure the state in a full set of measurement observables. Typically the Pauli basis is used, with observables  $X, Y$  &  $Z$  for every qubit, resulting in a total of  $3^n$  different measurements. By repeating, the expectation value of each observable can be determined, and from this the density matrix state can be reconstructed. This process is known as quantum state tomography or QST.

When the action of a quantum mechanical operation on an *arbitrary* input state needs to be specified completely, it essentially means that a description needs to be found that characterizes the operation as an open quantum system. As discussed in chapter 3, any representation introduced there will do, as they all are equivalent. Such undertakings of characterizing a system as an open quantum system is known as *quantum process tomography* or QPT. There are various ways of doing this, with the most straightforward method being standard quantum process tomography or SQPT. Quantum process tomography is very costly; there are at least  $12^n$  different experiments to specify an  $n$ -qubit system, which all need to be performed a large number ( $\sim 1000 - 10000$ ) of times to estimate the expectation values of those experiments.

In this chapter, I will introduce the method of SQPT to characterise the operations as quantum channels as introduced in chapter 3 via one of the representations, which is in section 4.1. This is done by inversion of a linear system of equations. Errors introduced by this method are discussed in section 4.2. These errors can lead to estimated descriptions of the channel that are not CPTP, so a method of solving these problems is discussed in section 4.3. A method to characterize the errors induced by a channel is introduced as the error matrix  $\chi^{\text{err}}$  in section 4.4. The fidelity of a process is discussed and linked to the eigenvalues of the  $\chi$  matrix. Furthermore, a brief discussion on other methods of characterizing quantum systems can be found at the end of the chapter in section 4.5.

### 4.1 Standard quantum process tomography

In standard quantum process tomography, the expectation values of measurements are linked to a description of the operation like its  $\chi$  matrix or the Choi matrix. This gives a system of equations to be solved. I will denote the expectation value of a measurement observable  $M$  on a qubit in the state  $\rho$  with  $E_{\rho, M} = \text{tr}[M\rho]$ . First I will show how  $\chi$  and the expectation values are related in section 4.1.1. Then, in section 4.1.2, I will show how to obtain the expectation values from actual measurement data.

### 4.1.1 SQPT: Linear inversion

$E_{\rho,M}$  is of the form  $E_{\rho,M} = \text{tr}[M\rho]$ . If the expectation values of a full set of observables are known, the full output state  $\rho$  can be specified; a full set of observables is any set of operators that spans the operator space for  $\rho$ . Most often, the Pauli basis  $\mathcal{P}_n$  is used, resulting in  $4^n$  different expectation values that are needed to characterize a density matrix. The reconstruction of  $\rho$  is then straightforward:

$$\hat{\rho} = \sum_{P_k \in \mathcal{P}_n} \frac{1}{d} E_{\rho,P_k} P_k. \quad (4.1)$$

This is known as quantum state tomography.

If the goal is not to characterize a state but a process, we can accomplish this by characterizing the action of the operation on an informationally complete set of input states, by performing QST on all corresponding outputs, where an informationally complete set of states is a set for which the density matrices within that set span the entire operator space; this is minimally  $4^n$ . However, another used option is the set of Pauli eigenstates. Note that this regards all eigenstates of all Paulis, for a total of  $6^n$ .

The expectation value of a state  $\rho_{\text{out}} = \Lambda(\rho_{\text{in}})$  is then:

$$E_{\rho_{\text{in}},M} = \text{tr}[M\rho_{\text{out}}] = \text{tr}[M\Lambda(\rho_{\text{in}})] = \text{tr}\left[M \sum_{m,n} \chi_{mn} P_m \rho_{\text{in}} P_n\right] = \sum_{m,n} \chi_{m,n} \text{tr}[M P_m \rho_{\text{in}} P_n]. \quad (4.2)$$

Now, if the set of input states is informationally complete, we can rewrite all expectation values of the form  $E_{\rho_{\text{in}},M}$  (as obtained from measurements) into a form  $\lambda_{ij} = \text{tr}[P_j \Lambda(P_i)]$ . Here both the measurement observable and the input density matrix is regarded as a Pauli operator. We then get the simplified expression:

$$\lambda_{ij} = \sum_{m,n} \chi_{m,n} \text{tr}(P_j P_m P_i P_n), \quad (4.3)$$

with  $i$  corresponding to the input Pauli, and  $j$  corresponding to the measurement basis.

This is a set of linear equations, that links the  $\lambda_{ij}$ 's with the process matrix  $\chi$ . When the Paulis are used for  $\lambda_{ij}$ , there are just as many elements in  $\lambda_{ij}$  as there are elements in  $\chi_{m,n}$ : ( $d^4$ ). This set of equations can therefore be specified by a square matrix, when  $\lambda_{ij}$  and  $\chi_{m,n}$  are both put in vector form:

$$\boldsymbol{\lambda} = A\boldsymbol{\chi}, \quad (4.4)$$

with  $A$  the  $d^4 \times d^4$  matrix that links  $\chi$  and  $\lambda$ :  $A_{ij,mn} = \text{tr}[P_j P_m P_i P_n]$ .  $A$  is Hermitian<sup>1</sup> and unitary. This means that  $A^{-1} = A^\dagger = A$  and thus that  $\boldsymbol{\chi} = A\boldsymbol{\lambda}$ ; by de-stacking  $\boldsymbol{\chi}$  the process matrix  $\chi$  is then obtained.

Note that the Pauli operators  $P_i$  cannot actually be prepared, because they are not valid density operators. However, by linearity of quantum mechanics they can be written as linear combinations of their eigenstates, which can be prepared experimentally. Therefore, the expectation values  $\lambda_{ij}$  from Eq. (4.3) can be written as linear combinations too.

### 4.1.2 Obtaining the expectation values from measurement data

#### Writing the input operators as sums of actual input states

As stated in the previous section, the expectation values  $\lambda_{ij}$  from Eq. (4.3) are not directly accessible by measurement. For example, it is impossible to prepare a trace 0 operator, so we can not prepare a Pauli basis as an input state  $\rho_{\text{in}}$  as in Eq. (4.3). However, using the linearity of quantum mechanics, different experimentally attainable expectation values can be combined to calculate all  $\lambda_{ij}$ .

To see this, note that any Pauli operator can always be written as its eigendecomposition:  $P = \sum_i \lambda_i |\psi_i\rangle \langle \psi_i|$ . Since any Pauli only has +1 and -1 eigenvalues, this is equal to:

$$P = \sum_{\lambda_i=1} |\psi_+\rangle \langle \psi_+| - \sum_{\lambda_i=-1} |\psi_-\rangle \langle \psi_-|, \quad (4.5)$$

<sup>1</sup>This follows from:  $\overline{A_{mn,ij}} = \overline{\text{tr}[P_n P_i P_m P_j]} = \text{tr}[(P_n P_i P_m P_j)^\dagger] = \text{tr}[P_j^\dagger P_m^\dagger P_i^\dagger P_n^\dagger] = \text{tr}[P_j P_m P_i P_n] = A_{ij,mn}$ , with the overline denoting the complex conjugate of the scalar.

with  $\{|\psi_+\rangle\}$  a basis for the +1 eigenspace and  $\{|\psi_-\rangle\}$  a basis for the -1 eigenspace of  $P$ . The individual eigenvectors  $|\psi_+\rangle\langle\psi_+|$  and  $|\psi_-\rangle\langle\psi_-|$  can now be prepared on a system of qubits. If the basis as introduced in section ?? is used, the states that need to be prepared are just the 6 1-qubit Pauli eigenstates on every qubit, for a total of  $6^n$  preparation states. All Paulis, including the identity operator, can be written as a linear combination of these  $6^n$  states.

A specific expectation value  $\lambda_{ij} = \text{tr}[P_j\Lambda(P_i)]$  can then be decomposed in the eigenbasis of  $P_i$ , because both the map and the trace is linear:

$$\lambda_{ij} = \sum_{\{|\psi_+\rangle\}} \text{tr}[P_j\Lambda(|\psi_+\rangle\langle\psi_+|)] - \sum_{\{|\psi_-\rangle\}} \text{tr}[P_j\Lambda(|\psi_-\rangle\langle\psi_-|)]. \quad (4.6)$$

Expectation values can not be measured themselves, they can only be estimated from measuring observables on qubit states. So, actual experiments allow us to estimate  $\text{tr}[P_j\Lambda(|\psi\rangle\langle\psi|)]$ , for any  $P_j \in \mathcal{P}_n$ . If a measurement apparatus provides this specific form data, the expectation values  $\lambda_{ij}$  are easily calculated using Eq. (4.6).

Extra care has to be taken when the decomposition of  $P_j$  into single qubit Paulis contains one or more identity operators, because then  $P_j$  is not a valid measurement observable. However, from the outcomes of the measurements of observables that are tensor products of only the Pauli  $X, Y$  and  $Z$  operators,  $\lambda_{ij}$  for any  $P_j$  can be determined, as explained in the next section.

### Determining the expectation value of a measurement from data

When a single qubit in a state  $\rho$  is measured in the eigenbasis of a Pauli  $P_j$ , the outcome of that measurement indicates whether that qubit is in the +1 or -1 eigenstate of that basis. If the measurement is repeated  $N_{\text{tot}}$  times,  $N_+$  times it will be in the +1 eigenstate, and  $N_-$  in the -1 eigenstate, with  $N_+ + N_- = N_{\text{tot}}$ . The expectation value  $\lambda_{\rho, P_j}$  can now be estimated from these counts:  $\lambda_{\rho, P_j} \approx \frac{N_+ - N_-}{N_{\text{tot}}}$ . This can be thought of as decomposing the measurement observable into its eigenbasis:

$$\lambda_{\rho, P_j} = \text{tr}[P_j\rho] = \text{tr}[|\psi_+\rangle\langle\psi_+|\rho] - \text{tr}[|\psi_-\rangle\langle\psi_-|\rho] \approx \frac{N_+}{N_{\text{tot}}} - \frac{N_-}{N_{\text{tot}}}. \quad (4.7)$$

Extending to multiple qubits is now straightforward. The measurement Pauli  $P_j$  can be decomposed into its eigenspaces ( $P_j = \Pi_{+j} - \Pi_{-j}$ ), and if the counts of the +1 and -1 eigenspace projections are known, the individual terms  $\text{tr}[P_j\Lambda(|\psi\rangle\langle\psi|)]$  in Eq. (4.6) can be estimated from these counts:

$$\text{tr}[P_j\Lambda(|\psi\rangle\langle\psi|)] = \text{tr}[\Pi_{+j}\Lambda(|\psi\rangle\langle\psi|)] - \text{tr}[\Pi_{-j}\Lambda(|\psi\rangle\langle\psi|)] \approx \frac{N_+}{N_{\text{tot}}} - \frac{N_-}{N_{\text{tot}}}. \quad (4.8)$$

Performing a non-destructive measurement to determine if a multi-qubit state is in the +1 or -1 eigenspace of  $P_j$  is generally not trivial<sup>2</sup>. However,  $P_j$  can, just like  $P_i$ , be decomposed into a basis of its +1 and -1 eigenspaces,  $\{|\phi_+\rangle\}$  and  $\{|\phi_-\rangle\}$  resp.:

$$\text{tr}[P_j\Lambda(|\psi\rangle\langle\psi|)] = \sum_{\{|\phi_+\rangle\}} \text{tr}[|\phi_+\rangle\langle\phi_+|\Lambda(|\psi\rangle\langle\psi|)] - \sum_{\{|\phi_-\rangle\}} \text{tr}[|\phi_-\rangle\langle\phi_-|\Lambda(|\psi\rangle\langle\psi|)]. \quad (4.9)$$

All of these decompositions can in principle be performed for any set that makes an eigenbasis; we need only ensure that the two sets spans the whole operator space. However, by using the tensor products of the single qubit Pauli eigenstates as a basis (i.e. the eigenstates introduced in section 2.2.4), the measurements become separable: for a measurement of  $P_j = P_1^1 \otimes \dots \otimes P_1^n$ , the  $i$ th qubit can be measured in the basis specified by  $P_1^i$  (if  $P_1^i \neq I$ ), and from those measurement outcomes, the corresponding multi-qubit eigenvalue can be determined. When a  $P_1^i = I$ , the measurements of any other Pauli observable can be used; all measurement outcomes then correspond to a +1 eigenstate because every state is a +1 eigenstate of  $I$ .

There is no need to collapse the state to an eigenspace of  $P_j$  in one measurement; this can be done in steps by collapsing to all eigenspaces of the separate single qubit Paulis separately. Measuring all

<sup>2</sup>One way of doing this is by entangling the system with an ancilla in the same way as a stabilizer measurement. This has several drawbacks, but might also have advantages. See 9.1.

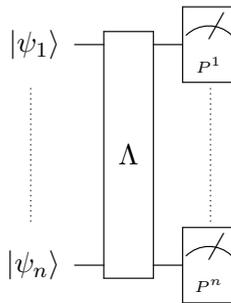
qubits separately gives rise to a bitstring  $s$  of length  $n$ , that then corresponds to a specific eigenstate  $|\phi_s\rangle$ , for a total of  $6^n$  different bitstrings, coming from  $3^n$  different measurement observables, namely  $X$ ,  $Y$  and  $Z$  for every qubit.

Now, by running the experiment  $N_{\text{tot}}$  times, we get counts  $\{N_s\}$  for every measurement outcome  $s$  corresponding to eigenstate  $|\phi_s\rangle$ , with  $\sum_s N_s = N_{\text{tot}}$ . We can subdivide all bitstrings  $s$  into the sets  $\{s_+\}$  and  $\{s_-\}$  corresponding to the  $+1$  and  $-1$  eigenstates, respectively. Using these, we can estimate any  $\text{tr}[P_j \Lambda(|\psi\rangle\langle\psi|)]$ :

$$\text{tr}[P_j \Lambda(|\psi\rangle\langle\psi|)] \approx \sum_{s \in \{s_+\}} \frac{N_s}{N_{\text{tot}}} - \sum_{s \in \{s_-\}} \frac{N_s}{N_{\text{tot}}}. \quad (4.10)$$

### 4.1.3 Calculating $\lambda_{ij}$

Using both Eq. (4.6) and Eq. (4.10) we can now calculate  $\lambda_{ij} = \text{tr}[P_j \Lambda(P_i)]$  for all  $16^n$  combinations of  $P_i$  and  $P_j$ . All combinations can be built up from the  $6^n$  different input states and  $3^n$  different measurements, resulting in  $18^n$  different experiments for full process tomography. A circuit representing this experiment can be found in figure 4.1. Furthermore, each experiment needs to be performed  $N_{\text{tot}}$  times, to obtain an estimate  $\frac{N_s}{N_{\text{tot}}}$  for every individual term in Eq. (4.9).



**Figure 4.1:** A SQPT experiment. There are  $n$  physical qubits; they are all individually prepared in specific eigenstates of the single-qubit Pauli operators, resulting in a state  $\rho_{\text{in}}$  on all qubits together. The map  $\Lambda$  is then applied to the qubits. The observable  $P_j = P_1^1 \otimes \dots \otimes P_1^n$ , with  $P_1^i$  the single-qubit Paulis that make the  $n$ -qubit Pauli  $P_j$ , is then measured on the output state  $\rho_{\text{out}} = \Lambda(\rho_{\text{in}})$ . This is done by performing measurements of  $P^i$  on the  $i$ th qubit for all  $n$  qubits. This experiment is repeated for different  $\rho_{\text{in}}$  and different  $P_j$ , so that the set of input states  $\rho_{\text{in}}$  and measurement observables  $P_j$  both span the operator space individually. Here, for the set of input states is chosen the set of all Pauli eigenstates, and for the set of measurement observables is chosen all  $3^n$  tensor products of the single qubit traceless Paulis.

If the experiments were perfect, than this would conclude SQPT;  $\chi$  can now be determined from these experiments. However, error and noise will influence the estimate of  $\chi$ . These effects will be discussed in section 4.2.

## 4.2 Errors and noise on data

Error on the data will give rise to an imperfect estimation of  $\chi$ . There will be both statistical noise (due to finite sampling) and imperfections on the preparations and measurements. These imperfections give rise to errors generally named SPAM errors (short for state-preparation and measurement). Generally speaking, statistical and SPAM errors can be considered uncorrelated, resulting in an estimate of  $\lambda$  that is a sum of the two:  $\hat{\lambda} = \lambda + \Delta\lambda_{\text{stat}} + \Delta\lambda_{\text{SPAM}}$ . Note that the natures of  $\Delta\lambda_{\text{stat}}$  and  $\Delta\lambda_{\text{SPAM}}$  are very different:  $\Delta\lambda_{\text{stat}}$  is a purely mathematical, indicating that the finite sample size of the estimation is not perfect, whereas  $\Delta\lambda_{\text{SPAM}}$  is a physical error happening on the circuit because the physical implementation of the preparations and measurements are not perfect.

### 4.2.1 Statistical errors: standard deviation of $\chi$ and confidence bounds

Statistical noise on  $\chi$  arises from the fact that the expectation values of each term  $\text{tr}[|\phi_s\rangle\langle\phi_s| \Lambda(|\psi\rangle\langle\psi|)]$  in Eq. (4.9) can only be estimated from a finite number of experiments  $N_{\text{tot}}$ . The measurements that

are performed to determine these expectation values are essentially samples drawn from a binomial distribution; the expectation value  $p_s$  of such a variable is estimated by  $\frac{N_{\text{success}}}{N_{\text{tot}}}$ , and the standard deviation  $\Delta p_s$  on  $p_s$  is  $\Delta p_s = \sqrt{\frac{p_s(1-p_s)}{N_{\text{tot}}}}$ . Since  $0 \leq p_s \leq 1$ , we have  $\Delta p_s \leq \frac{1}{2\sqrt{N_{\text{tot}}}}$ . Then, for the expectation values calculated in Eq. (4.10), by standard methods [9]:

$$\Delta(\text{tr}[P_j \Lambda(|\psi\rangle\langle\psi|)]) = \sqrt{\sum_{s \in \{s_+, \{s_-\}\}} \Delta p_s^2} = \frac{1}{\sqrt{N_{\text{tot}}}} \sqrt{\sum_{\{s\}} p_s(1-p_s)} \leq \frac{2^{\frac{n}{2}}}{2\sqrt{N_{\text{tot}}}}. \quad (4.11)$$

Then, using Eq. (4.6), the standard deviation on  $\lambda_{ij}$  can be calculated:

$$\Delta \lambda_{ij} = \sqrt{\sum_{\{\psi_+\}} (\Delta \text{tr}[P_j \Lambda(|\psi_+\rangle\langle\psi_+|)])^2 + \sum_{\{\psi_-\}} (\Delta \text{tr}[P_j \Lambda(|\psi_-\rangle\langle\psi_-|)])^2} \leq \frac{2^n}{2\sqrt{N_{\text{tot}}}}. \quad (4.12)$$

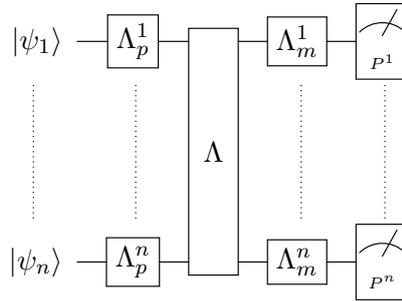
Any individual  $\lambda_{ij}$  is a sum of at most  $d^2$  terms, with every term bounded by  $-1$  and  $+1$ . Therefore,  $\lambda_{ij}$  is bounded too:  $-2^{2n} \leq \lambda_{ij} \leq 2^{2n}$ , and the bound from Eq. (4.12) can only be saturated if  $\lambda_{ij} = 0$ . Finally, the standard deviation on  $\chi$  is:

$$\Delta \chi = \sqrt{(A \odot A)(\Delta \lambda \odot \Delta \lambda)} \leq \frac{(4)^n}{2\sqrt{N_{\text{tot}}}}, \quad (4.13)$$

where  $\odot$  denotes the Hadamard or elementwise product, and the bound on  $\Delta \chi$  is per element. These bounds are not very tight, and the bound on  $\Delta \chi$  is especially loose, since it adds deviations on many different  $p_s$ , which are all assumed to be equal to  $\frac{1}{2}$ . Using these calculated standard deviations, confidence bounds on  $\chi$  can be calculated. Furthermore, in any subsequent analysis on  $\chi$  these standard deviations are used as well.

## 4.2.2 SPAM errors

SPAM errors are those errors associated with imperfect state preparation and imperfect measurements. These influences can be represented by (unknown) error maps on every single qubit right after their (perfect) preparation and right before their (perfect) measurement. A schematic of this representation can be found in figure 4.2. The methods introduced in section 4.1 do not provide any means to correct for these errors. In section 8.1 I will consider SPAM errors and introduce an approximate method to correct them.



**Figure 4.2:** An actual SQPT experiment with SPAM errors. Immediately after (perfectly) preparing every qubit, an individual (unknown) error map  $\Lambda_p^i$  is applied to the  $i$ th qubit, for every qubit. Thereafter, the map  $\Lambda$  is applied to all qubits. Then, before a perfect measurement of each separate qubit, an individual (unknown) error map  $\Lambda_m^i$  is applied to the  $i$ th qubit, for every qubit. Compare with figure 4.1.

## 4.3 Ensuring CP and TP

Problems when determining  $\chi$  via the methods described in 4.1 arise whenever there is noise on  $\lambda$ . Particularly, the found solution  $\chi^{\text{meas}}$  will be of the form  $\chi^{\text{meas}} = \chi + \chi^{\text{noise}}$ , with  $\chi^{\text{noise}}$  dependent

on the noise on  $\lambda$ . In principle,  $\chi$  should represent a CPTP map, but  $\chi^{\text{meas}}$  might not, due to the influence of the noise. This results in an estimate of  $\chi$  that is not in the CPTP-subspace of the space of process matrices; to obtain a valid representation of the channel, we need to find a CPTP map which is close to  $\chi^{\text{meas}}$  in some sense.

Since CP and TP are two different criteria, they will be discussed separately. We will first consider TP, and show that the method of obtaining  $\chi$  introduced in section 4.1 will always give a TP channel when a loose condition on the measurement data is fulfilled. Thereafter, we will introduce a method of obtaining a CP  $\chi$  close to the original  $\chi^{\text{meas}}$ . This method preserves TP, and thus provides a CPTP  $\chi$  if  $\chi^{\text{meas}}$  was TP to begin with.

### 4.3.1 TP as condition on the measurements

For measurement data of an SQPT experiment a strong statement can be made about whether the estimated representation of  $\Lambda$  is trace preserving or not. If  $\sum_{\{s\}} N_s = N_{\text{tot}}$  then the estimated representation will be a TP channel. To prove this, the condition of TP for the Choi matrix is rewritten to a form that can be directly linked to the expectation values  $\lambda_{ij}$  for  $P_j = I$ . The proof is as follows:

For trace preservation, a necessary and sufficient condition is Eq. (3.4) found in section 3.2.2. It is:

$$\text{tr}_1 [\rho_{\text{Choi}}] = \frac{1}{d} I. \quad (4.14)$$

Since  $\rho_{\text{Choi}} = (\Lambda \otimes I)(|\Omega\rangle\langle\Omega|) = \sum_{i,j} \frac{1}{d} \Lambda(|i\rangle\langle j|) \otimes |i\rangle\langle j|$ , for Eq. (4.14) to hold we get the identity:

$$\text{tr} [\Lambda(|i\rangle\langle j|)] = \delta_{ij}. \quad (4.15)$$

We can expand  $|i\rangle\langle j|$  into the Pauli basis:  $|i\rangle\langle j| = \sum_k \frac{1}{\langle P_k, P_k \rangle} \langle j| P_k |i\rangle P_k$ , with  $\frac{1}{\langle P_k, P_k \rangle}$  a normalisation factor. We then get for  $\text{tr} [\Lambda(|i\rangle\langle j|)]$  (with  $\langle P_k, P_k \rangle = d$ ):

$$\text{tr} [\Lambda(|i\rangle\langle j|)] = \frac{1}{d} \text{tr} \left[ \Lambda \left( \sum_k \langle j| P_k |i\rangle P_k \right) \right] = \frac{1}{d} \sum_k \langle j| P_k |i\rangle \text{tr} [\Lambda(P_k)]. \quad (4.16)$$

The very last term  $\text{tr} [\Lambda(P_k)]$  is  $\lambda_{ij}$  from section 4.1, with  $P_j = I$  and  $P_i = P_k$ .

Following the same reasoning as with Eq. (4.6),  $P_k$  can be expanded into a sum of its positive and negative eigenspaces, obtaining:

$$\text{tr} [I\Lambda(P_k)] = \sum_+ \text{tr} [I\Lambda(|\psi_+\rangle\langle\psi_+|)] - \sum_- \text{tr} [I\Lambda(|\psi_-\rangle\langle\psi_-|)], \quad (4.17)$$

with  $\{|\psi_+\rangle\}$  and  $\{|\psi_-\rangle\}$  a basis for the +1 and -1 eigenspaces of  $P_k$ .

Since  $I$  only has a +1 eigenspace,  $\text{tr} [I\Lambda(|\psi\rangle\langle\psi|)]$  is, by using Eq. (4.10), equal to:

$$\text{tr} [I\Lambda(|\psi\rangle\langle\psi|)] = \sum_{s \in \{s\}} \frac{N_s}{N_{\text{tot}}}, \quad (4.18)$$

where  $\{s\}$  represents a basis for the entire space. In other words, it is summing all counts  $N_s$  of all individual outcomes  $\text{tr} [|\phi_+\rangle\langle\phi_+| \Lambda(|\psi\rangle\langle\psi|)]$  for  $|\phi_+\rangle$  all +1 eigenstates of the identity operator.

If there are no preparations unaccounted for, i.e. if every prepared shot is also measured and the measurement always returns a value, then  $\sum_{\{s\}} N_s = N_{\text{tot}}$ . This results in  $\text{tr} [I\Lambda(|\psi\rangle\langle\psi|)] = 1$ .

Every Pauli that is not the identity has equally large +1 and -1 eigenspaces, whereas the identity only has a +1 eigenspace. Therefore, Eq. (4.17) results in  $\text{tr} [\Lambda(P_k)] = d$  for  $P_k = I$  and 0 otherwise. It is now evident that Eq. (4.16) becomes:

$$\frac{1}{d} \sum_k \langle j| P_k |i\rangle \text{tr} [\Lambda(P_k)] = \frac{1}{d} \langle j| I |i\rangle \text{tr} [\Lambda(I)] = \delta_{ij}. \quad (4.19)$$

These derivations work from bottom to top as well, so Eq. (4.15) holds if  $\sum_{\{s\}} N_s = N_{\text{tot}}$  for every prepared input state, thereby ensuring trace preservation of the estimated process.

### 4.3.2 Finding a CP representation of $\chi$

If  $\chi^{\text{meas}}$  does not represent a CP map, then at least 1 of its eigenvalues is negative. The goal now is to estimate a physical process  $\chi^{\text{est}}$  that is close to  $\chi^{\text{meas}}$  and fulfils the CPTP conditions. Here, close means that the distance  $d(\chi^{\text{meas}}, \chi^{\text{est}})$  is small, where the distance is usually defined using some norm. Many norms are possible, here the *Frobenius* norm is used:

$$d(\chi^{\text{meas}}, \chi^{\text{est}}) = \|\chi^{\text{meas}} - \chi^{\text{est}}\|_2 = \sqrt{\sum_{ij} |\chi_{ij}^{\text{meas}} - \chi_{ij}^{\text{est}}|^2} = \sqrt{\text{tr}[(\chi^{\text{meas}} - \chi^{\text{est}})^2]}, \quad (4.20)$$

where the last identity follows from the fact that both  $\chi^{\text{meas}}$  and  $\chi^{\text{est}}$  are Hermitian.

This norm is a specific choice, many others are possible. It is basis-independent [10], meaning that  $\chi^{\text{meas}}$  and  $\chi^{\text{est}}$  can be expressed in any basis (as long as both are expressed in the same basis). Therefore we can change to the eigenbasis of  $\chi^{\text{meas}}$ , where  $\chi^{\text{meas}}$  is a diagonal matrix. From Eq. (4.20) it is obvious that  $\chi^{\text{est}}$  expressed in this basis should have off-diagonal elements equal to zero if the distance is to be minimized. The diagonal elements can then be viewed as the eigenvalues of  $\chi^{\text{est}}$ ; we impose the CP constraint that these are all non-negative. Note that now  $\chi^{\text{meas}}$  and  $\chi^{\text{est}}$  have the same eigenvectors. We now get the following function for the distance that is to be minimized:

$$d(\chi^{\text{meas}}, \chi^{\text{est}}) = \sqrt{\sum_i |\lambda_i^{\text{meas}} - \lambda_i^{\text{est}}|^2}, \quad (4.21)$$

with  $\lambda_i^{\text{meas}}$  the eigenvalues of  $\chi^{\text{meas}}$  and the constraint  $\lambda_i^{\text{est}} \geq 0, \forall i$ .

Finding the closest CPTP  $\chi^{\text{est}}$  is generally hard to do, but finding a relatively close CP  $\chi^{\text{est}}$  can be done by taking a convex combination of  $\chi^{\text{meas}}$  with the identity channel. This works as follows:

- Express  $\chi^{\text{meas}}$  in the Pauli basis.
- Diagonalize  $\chi^{\text{meas}}$  to obtain a set of eigenvalues  $\lambda_i^{\text{meas}}$  and their corresponding eigenvectors  $|\lambda_i^{\text{meas}}\rangle$ . Form the diagonalization  $\chi^{\text{meas}} = UDU^\dagger$ .
- If  $\lambda_i^{\text{meas}} \geq 0 \forall i$  then  $\chi^{\text{meas}}$  represents a CP map.
- If not, get the smallest eigenvalue  $\lambda_{\min}^{\text{meas}}$  and add this to  $D$ :  $\hat{D} = D + |\lambda_{\min}^{\text{meas}}|I_{d^2}$ .
- Obtain a new matrix  $\hat{\chi}^{\text{meas}} = U\hat{D}U^\dagger$ . This matrix has only non-negative eigenvalues by construction.
- $\text{tr}[\hat{\chi}^{\text{meas}}] = \text{tr}[\chi^{\text{meas}}] + \text{tr}[|\lambda_{\min}^{\text{meas}}|I_{d^2}] = 1 + |\lambda_{\min}^{\text{meas}}|d^2$ . Any map should have trace 1, so set  $\chi^{\text{est}} = \frac{\hat{\chi}^{\text{meas}}}{\text{tr}[\hat{\chi}^{\text{meas}}]} = \frac{\hat{\chi}^{\text{meas}}}{1 + d^2|\lambda_{\min}^{\text{meas}}|}$

We have  $\sum_{m,n} \chi_{m,n}^{\text{est}} P_n^\dagger P_m = \frac{1}{\text{tr}[\hat{\chi}^{\text{meas}}]} (\sum_{m,n} \chi_{m,n}^{\text{meas}} P_n P_m + \sum_n |\lambda_{\min}^{\text{meas}}| P_n P_n) = \frac{1 + |\lambda_{\min}^{\text{meas}}|d^2}{\text{tr}[\hat{\chi}^{\text{meas}}]} I_d = I_d$ , so

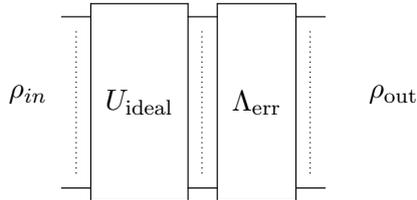
Eq. (3.2) for TP still holds. This means that  $\chi^{\text{est}}$  is a process matrix that represents a CPTP map, if it is derived from a process matrix  $\chi^{\text{meas}}$  that is TP.

## 4.4 The error matrix $\chi^{\text{err}}$

If tomography is performed on a circuit that should be a unitary transformation  $U_{\text{ideal}}$ , then if both the circuit is perfectly implemented and the tomography is perfect (implying  $N_{\text{tot}} \rightarrow \infty$ ), the obtained  $\chi^{\text{meas}}$  will correspond perfectly to that unitary. In practise, there will be errors in the implementation, resulting in an estimate  $\chi^{\text{est}}$ , which I will now just write as  $\chi$ . Characterizing this error can in principle be done directly from  $\chi$ , but this is less than ideal. However, the whole circuit can be modelled (see figure 4.3) as a process that first performs the perfect unitary and then applies an error channel:

$$\rho_{\text{out}} = \Lambda(\rho_{\text{in}}) = \sum_{m,n} \chi_{m,n}^{\text{err}} P_m U_{\text{ideal}} \rho_{\text{in}} U_{\text{ideal}}^\dagger P_n^\dagger \quad (4.22)$$

Here  $\chi^{\text{err}}$  is the *error matrix* [11].  $\chi^{\text{err}}$  can be calculated from  $\chi$  as  $\chi^{\text{err}} = V\chi V^\dagger$  with  $V_{m,n} = \text{tr}[P_m^\dagger P_n U_{\text{ideal}}^\dagger] \frac{1}{d}$ , with  $d = \text{tr}[P_m^\dagger P_m]$ . The error matrix is a direct characterization of the error map. In principle any basis can be used for the error matrix, but it is very useful to use the Pauli basis for  $\chi^{\text{err}}$ . By using the Pauli basis, the top left element of  $\chi^{\text{err}}$  is the element corresponding to the trivial part  $U_{\text{ideal}}\rho_{\text{in}}U_{\text{ideal}}^\dagger$ ; any non-zero element of  $\chi^{\text{err}}$  that is not in the top left corner therefore indicates an imperfection of the total process that is the composition of  $U$  and  $\Lambda_{\text{err}}$ .



**Figure 4.3:** For a desired unitary operation  $U_{\text{ideal}}$ , the total obtained map  $\Lambda$  can be modelled as the perfect unitary and thereafter an error map  $\Lambda_{\text{err}}$ .  $\Lambda_{\text{err}}$  then has a representation  $\chi^{\text{err}}$ , which describes the error induced by the channel.

It can be shown [11] that for systematic unitary errors  $U^{\text{err}}$  all non-zero elements of  $\chi^{\text{err}}$  are, to first order, the imaginary parts of the elements in the top row and leftmost column:  $\chi_{m,n}^{\text{err}} = 0$  for  $m, n > 1$ . Since  $\chi^{\text{err}}$  is Hermitian, specifying either set of elements is adequate. Furthermore, the elements in this row (or column) indicate the decomposition of  $U^{\text{err}}$  on the channel into the basis  $\{P_k\}$ :  $U^{\text{err}} = \sum_k \text{Im}\{\chi_{0,k}^{\text{err}}\} P_k$ .

#### 4.4.1 Fidelity of the process

The process fidelity  $F_p$  of a unitary map represented by  $\chi$  is  $\text{tr}[\chi\chi_{\text{des}}]$  (see Eq. (3.25)). Because the error matrix is essentially the  $\chi$  matrix in a different basis, and because the trace is basis independent, we can use the error matrix to calculate the process fidelity more easily:

$$F_p = \text{tr}[\chi\chi_{\text{des}}] = \text{tr}[V^\dagger\chi^{\text{err}}VV^\dagger\chi_{\text{des}}^{\text{err}}V] = \text{tr}[\chi^{\text{err}}\chi_{\text{des}}^{\text{err}}]. \quad (4.23)$$

The desired error matrix  $\chi_{\text{des}}^{\text{err}}$  is of course the identity channel denoted by  $\chi_I$ .  $\chi_I$  has only one non-zero element, in its top-left corner it equals 1. Therefore, the process fidelity  $F_p$  is exactly the top left element of  $\chi^{\text{err}}$ :

$$F_p = \text{tr}[\chi^{\text{err}}\chi_{\text{des}}^{\text{err}}] = \text{tr}[\chi^{\text{err}}\chi_I] = \chi_{0,0}^{\text{err}}. \quad (4.24)$$

By expanding  $\chi$  into its eigendecomposition and noting that for unitary desired process  $\chi_{\text{ideal}}$  only has one non-zero eigenvalue (see section 3.3.3), the process fidelity can be calculated in terms of the eigenvalues of  $\chi$ :

$$F_p = \text{tr}[\chi\chi_{\text{des}}] = \sum_i \lambda_i \text{tr}[|\psi_i\rangle\langle\psi_i|u\rangle\langle u|] = \sum_i \lambda_i |\langle\psi_i, u\rangle|^2. \quad (4.25)$$

Furthermore, if  $\chi$  has reasonable fidelity with respect to  $\chi_{\text{des}}$ , one of its eigenvectors is at least close to  $U_{\text{ideal}}$ 's corresponding eigenvector  $|u\rangle$ , with the corresponding eigenvalue  $\lambda_{\text{max}}$  being the largest one. That means that  $|\langle\psi_{\text{max}}, u\rangle|^2 \approx 1$ . Moreover, all  $|\psi_i\rangle$ 's are orthogonal to each other, so for  $i \neq i_{\text{max}}$ , we have  $|\langle\psi_i, u\rangle|^2 \ll 1$ . Therefore, approximately:

$$F_p \approx \lambda_{\text{max}}, \quad (4.26)$$

with  $\lambda_{\text{max}}$  the maximum eigenvalue of  $\chi$ .

#### 4.4.2 Pauli Twirl of $\chi^{\text{err}}$

Since  $\chi^{\text{err}}$  describes all imperfections of the map, it can be used to analyse the weight of the errors of  $\chi$ . Taking the Pauli twirl of  $\chi^{\text{err}}$  results in an approximation of the error channel as a Pauli channel; from this channel the support on the single-qubit and multi-qubit Paulis can be calculated, together with their ratio.

## 4.5 Other characterization methods

The main drawback of SQPT is that it is very costly. There are other characterization methods that are better suited to specific situations. I will briefly introduce various other methods, though they will not be considered later.

### 4.5.1 Ancilla-assisted Quantum Process Tomography

In Ancilla-assisted Quantum Process Tomography [12],[13], or AAPT, the system of interest is first connected to an ancillary system of (at least) the same size. The maximally entangled state  $|\Omega\rangle$  is prepared on the  $2n$  qubits;  $\Lambda$  is now applied to half of the system. This results in the Choi matrix density state on the composite space (see sec 3.2.2). By performing state tomography on this state, the Choi matrix can be determined, thereby also characterizing the map  $\Lambda$ . The results in a preparation of a single state. State tomography prescribes measuring all qubits in the 3 different bases, resulting in a total of  $3^{2n} = 9^n$  different measurements.

This method has both drawbacks and advantages when compared to SQPT. The number of qubits that is needed to perform the tomography is twice as high for SQPT. Furthermore, on these  $2n$  qubits there needs to be a highly entangled state; this is generally much harder to accomplish than preparing the separable states.

On the other hand, there is only one prepared state, resulting in less complexity in the preparations. The total number of experiments is  $9^n$ , less than the minimum of  $12^n$  different experiments for SQPT. This is a clear advantage - especially for a large number of qubits.

AAPT is especially helpful when running diagnostics on a small subsystem within a quantum system or quantum computer [14]. In that case, there are enough qubits, and preparing the maximally entangled state is something that should be doable. A more in-depth discussion of AAPT can be found in [15].

### Necessity of entanglement in the input state

It is not necessary to prepare the maximally entangled state  $|\Omega\rangle\langle\Omega|$  on the composite space of qubits. Any state  $\rho$  with Schmidt number (see [6])  $2^{2n}$  [14] is adequate; this is also a necessary condition. Note that imperfections in the preparation of  $\rho$  directly leads to error in the estimate of  $\Lambda$ . This condition can be interpreted [15] as being a condition of invertibility of the resulting output state. This means that there is no intrinsic need for entanglement; even many states are adequate. This takes away the disadvantage in AAPT that is the preparation of the maximally entangled state. Specifically, a separable state of the following form (called a Werner state) can be shown [14] to be adequate:

$$\rho = \frac{1-\epsilon}{d^2}I + \epsilon|\Psi_{-}\rangle\langle\Psi_{-}|, \epsilon \leq \frac{1}{1+d} \quad (4.27)$$

with  $|\Psi_{-}\rangle$  the specific (extended) Bell state.

In the specific case that the maximally entangled state is indeed used, as introduced in the previous paragraph, this method is also called EAPT or Entanglement-Assisted Process Tomography.

### 4.5.2 Randomized Benchmarking

Randomized benchmarking is a method that does not determine a complete representation of an error map  $\Lambda$ . It does, however, quantify the error rates of gates in a system by randomly applying gates in such a way that the resulting total process should be an identity gate (i.e. all randomly applied gates compile to the identity channel). Any infidelity on the resulting output state indicates errors within the gates, or in the preparations or measurements. By altering the number of gates, and determining how the fidelity depends on the number of gates, the influence of the measurement and preparation errors can be filtered out. The method is much less costly and easier to implement than QPT but it does not provide any characterization of the induced maps. As such, for the characterization of error maps it is not adequate.

### 4.5.3 Direct Characterization of Quantum Dynamics

The main idea of Direct Characterization of Quantum Dynamics or DCQD is that when using the right input states, the only measurements that need to be performed are Bell state measurements, e.g. projections onto the Bell states. This, in general, needs to be only a single fixed measurement apparatus. The input states that need to be prepared are a (generalized) Bell state and  $d^2-1$  other non-maximally entangled states. Furthermore, the method is *direct*, meaning the measurement outcomes (or rather the expectation values of the measurements estimated by the measurement outcome counts) are directly indicative of the elements of the  $\chi$  matrix, without the need for linear inversion methods as for example in Eq. (4.4). Since these inversions add problems this directness is favourable. A more in-depth discussion of DCQD can be found in [15].

However, Bell state measurements are generally hard to perform. A measurement apparatus that performs these measurements will necessarily need to perform multi-qubit gates to prepare the qubits in the right states. 2-qubit gates are generally very faulty, influencing the measurement error rate. This is why this technique was not used in the experiments discussed in this report.

## Chapter 5

# Stabilizer codes

The errors that can occur within systems, as described in chapter 3, can introduce irreparable damage that limits the possibilities and computing power of a system. Quantum error correcting codes (QECCs) offer a way to combat these errors, by encoding a generally low number of qubits into a large number of actual qubits; the encoded qubits are then called the *logical* qubits, and the actual qubits are called the *physical* qubits. Detecting and correcting for errors in quantum states is hard to do, especially if the quantum state itself is to be left untouched. In quantum computation this generally will be the case, because the quantum states are elaborate superpositions - accidentally measuring them collapses the states and destroys the encoded information.

There exist many different QECCs, but most popular are the codes known as *Stabilizer codes*, introduced by Daniel Gottesman in his thesis [16]. Stabilizer codes are a special class of codes that make use of the many properties of Paulis; the error detection and correction happens by projecting upon the eigenspaces of many different Paulis together. They are the leading form of QECCs, and future quantum computation will (very, very) likely make use of some sort of stabilizer code.

This chapter introduces QECCs and stabilizer codes specifically. Properties and conditions general to any QECC are introduced in section 5.1, and to interpret these ideas a straightforward example of a code is given in section 5.1.1. That example is actually a stabilizer code, which are then introduced more generally in section 5.2. It is good to know exactly *which* errors a stabilizer code can (and can't) correct, so every kind of error on a stabilizer code is discussed in section 5.2.1. After their general introduction, section 5.3 introduces two very well known stabilizer codes, the so-called *perfect code* and the *Steane code*. Section 5.4 briefly discusses how to perform the necessary measurements to determine the errors that have occurred, without affecting the encoded information, and the chapter is closed by section 5.5 that introduces a set of operators called the Clifford operators.

### 5.1 Quantum error correcting codes

The general idea behind a QECC is to encode the logical qubits into a subspace  $\mathcal{H}_k$  (called the code space) of  $\mathcal{H}_n$ . The basisvectors of  $\mathcal{H}_k$  then form a basis for the logical qubits; an element from  $\mathcal{H}_k$  is called a *code word*. The dimension of  $\mathcal{H}_k$  is generally a power of 2: if  $|\mathcal{H}_k| = 2^k$  then the code encodes  $k$  logical qubits. Often  $k = 1$ , so that a code encodes one logical qubit into  $n$  physical qubits.

A QECC is designed in such a way that (typical) errors that can happen map the code space to other orthogonal subspaces of  $\mathcal{H}_n$ . The internal structure of  $\mathcal{H}_k$  is not distorted, so that encoded information is preserved. Because the errors are isomorphisms, they can be corrected without altering the state of the logical qubits. For this to work, there has to be a method of determining the error and corresponding subspace, without affecting the internal structure of the subspace; such a thing can be realized by a POVM, and the outcome of that measurement is called the *error syndrome*.

A code can never correct every possible error: there can only be  $2^{n-k}$  different orthogonal subspaces, including the code space itself, but each subspace necessarily has dimension  $2^k$ , and therefore there are more than one errors that map to this subspace. Two errors  $E_1$  and  $E_2$  might map to the same orthogonal subspace, while rearranging the codewords respective to each other. This means that  $E_1$  and  $E_2$  need to be corrected for differently:  $E_1^\dagger \neq E_2^\dagger$ . The syndrome measurement indicates which

orthogonal subspace it is, but nothing more. Hence, only one of the errors can be corrected for; the set of correctable errors  $\mathcal{E}$  for the code contains only one of the two. When designing the code, either of the two has to be chosen.

The conditions for a code to work can be summarized in the *Knill-Laflamme conditions* [17][18]:

A code  $\mathcal{H}_k$  together with a set of correctable errors  $\mathcal{E}$  is a QECC iff  $\forall |\psi\rangle, |\phi\rangle \in \mathcal{H}_k, \forall E, F \in \mathcal{E}$ , the following condition holds:

$$\langle \psi | E^\dagger F | \phi \rangle = C(E, F) \langle \psi | \phi \rangle. \quad (5.1)$$

The conditions say that the errors  $E_1$  and  $E_2$  can affect the codewords, mapping them to orthogonal subspaces ( $C(E, F)$  depends on the errors), but that the internal structure of the code space should be untouched ( $C(E, F)$  is not allowed to depend on the codewords).

A QECC is designed in such a way that the set of correctable errors  $\mathcal{E}$  has certain properties. An error with a high weight is less likely to occur than a lower weight error. So, if  $E_1$  and  $E_2$  both have the same error syndrome, but need to be corrected differently, then the lower weight error is assumed to have occurred. If a code is properly designed, all errors with  $\text{wt}(E) \leq w$ , for a certain value  $t$ , are included within the set of correctable errors  $\mathcal{E}$ . Then, the code is said to have distance  $d = 2w + 1$ ; the distance of a code is an often used measure for the correction capabilities of the code. It has an important interpretation: it is the minimum weight of an error that maps a code word to a different code word, i.e. a *logical error*.

If an error occurs on a system, the qubits are subjected to an error channel  $\Lambda$ . The Kraus operators  $\{A_k\}$  that represent this channel can be any element of  $SU(2)$  scaled by some factor; to distinct all of them by error syndrome would result in the need for uncountably many error syndromes. However, any element of  $SU(2)$  can be written as a linear combination of a basis of the space. Using the Pauli basis, we can decompose any  $A_k = \sum_{P_i \in \mathcal{P}_n} \alpha_i^k P_i$ . Applying  $A_k$  to a codeword  $|\psi\rangle$  results in

$$A_k |\psi\rangle = \sum_{P_i \in \mathcal{P}_n} \alpha_i^k P_i |\psi\rangle, \quad (5.2)$$

which is a coherent superposition of states  $P_i |\psi\rangle$ . If  $A_k$  can be decomposed into only the elements of  $\mathcal{E}$ , the resulting state is a superposition of erroneous codewords  $E_i |\psi\rangle$ , which is a superposition of states all in different orthogonal subspaces. The error syndrome measurement then projects the superposition onto any of these orthogonal subspaces, resulting in a single state  $P_i |\psi\rangle$ . Furthermore, the error syndrome measurement indicates which of these subspaces it is. The error syndrome measurement effectively discretizes the error, resulting that if a code corrects a set  $\mathcal{E}$ , it also corrects  $\text{Span}(\mathcal{E})$ .

The fact that for more than 1 Kraus operators a map  $\Lambda$  creates a statistical mixture, is no problem as well. If the Kraus operators are constrained to be linear combinations of only elements of  $\mathcal{E}$ , then the error syndrome measurement will project the state  $\sum_k A_k \rho A_k^\dagger$  onto a subspace (corresponding to a  $E \in \mathcal{E}$ ); it then does not matter any more what  $A_k$  was and the resulting state after syndrome measurement is the known state  $E |\psi\rangle^1$ . So, if an error map  $\Lambda$  that consists of Kraus operators that are strictly linear combinations of the elements of a QECC's correctable set  $\mathcal{E}$ , then that code can correct for that error map  $\Lambda$ .

Finally, a stabilizer code that encodes  $k$  logical qubits on  $n$  physical qubits, with a distance of  $d$ , is denoted as an  $[[n, k, d]]$  code. It is desirable for both  $\frac{k}{n}$  and  $\frac{d}{n}$  to be as high as possible. Clearly  $\frac{k}{n} < 1$  and  $\frac{d}{n} < \frac{1}{2}$ , but there exist much stronger bounds on the relations between  $n$ ,  $k$  and  $d$  [19][18]. Moreover, the set of physical qubits on which a code is implemented is also called the *codeblock*; often  $k = 1$  and the codeblock then represents one logical qubit. A larger quantum computing system will then have more than one codeblock.

### 5.1.1 Example: The 3-qubit bit-flip code

A typical example of a QECC is the 3-qubit bit-flip code. In this code, one logical qubit ( $k = 1$ ) is encoded into 3 physical qubits ( $n = 3$ ). The  $|0\rangle$  state is encoded into the state  $|000\rangle$ , and the  $|1\rangle$  state

<sup>1</sup>To be precise, it can be any  $E \in \mathcal{E}$  with the same error syndrome. But all those errors are the same unitary up to a stabilizer.

is encoded into the state  $|111\rangle$ , resulting in a logical codespace  $\mathcal{H}_k = \text{Span}\{|000\rangle, |111\rangle\}$ . When a line is written over a ket, it indicates that it is a logical state, so we write  $|000\rangle = |\bar{0}\rangle$  and  $|111\rangle = |\bar{1}\rangle$  for the logical  $|0\rangle$  and  $|1\rangle$  states. A general state  $\alpha|0\rangle + \beta|1\rangle$  is then encoded using the code as

$$\alpha|0\rangle + \beta|1\rangle \rightarrow \alpha|000\rangle + \beta|111\rangle = \alpha|\bar{0}\rangle + \beta|\bar{1}\rangle. \quad (5.3)$$

Now, if there is an  $X$  error on one of the qubits, resulting in a bit-flip, the erroneous qubit state is orthogonal to the codespace. If, for instance, the error  $E_1 = X \otimes I \otimes I$  occurs, the resulting state is

$$E_1(\alpha|000\rangle + \beta|111\rangle) \rightarrow \alpha|100\rangle + \beta|011\rangle. \quad (5.4)$$

Note that the new basis vectors  $|100\rangle$  and  $|011\rangle$  span a space orthogonal to  $\mathcal{H}_k$ . These states are relatively close to the original codewords, but we can not measure the qubits due to the projective nature of measurements in quantum mechanics. However, by measuring only the parity of the qubits, we keep the encoded information intact. The parity of the 1st and 2nd qubit is odd, whereas the parity of the 2nd and 3rd is even. These two results indicate that a error on the first qubit has happened. Note that the parity of the 1st and 3rd qubit does not offer any extra information; it can be calculated from the other two parities.

The outcome of the parity measurements acts as the error syndrome, which indicates what error has happened. It is the only information that we can obtain before destroying any encoded information. However, the error  $E_1$  is not the only error with this error syndrome. This shows the fundamental problem with QECCs: error syndromes can never distinct between all different errors than can occur on a set of qubits. The error  $E_2 = I \otimes X \otimes X$  has the same error syndrome as  $E_1$ , but it has to be corrected differently. Correcting  $E_1$  can be done by applying  $X \otimes I \otimes I$  to the qubits, but applying this to  $E_2$  results in  $X \otimes X \otimes X$ , which maps  $|\bar{0}\rangle$  to  $|\bar{1}\rangle$  and vice-versa, thus acting as the logical operation  $\bar{X}$ . By assuming that  $E_1$  occurred, and correcting for it, we have created a logical error, something we can never hope to correct without actually measuring the encoded information. We say that the 3-qubit bit-flip code corrects single-qubit  $X$  errors, and that it does not correct double  $X$  errors. This code does not correct any phase-flip errors, but we can instead encode into the logical states  $|+++ \rangle$  and  $|--- \rangle$  to correct for those, by making use of the same technique. With just three qubits, we can do either, but not both.

The 3-qubit bit-flip code corrects weight-1 bit-flip errors. Nevertheless, it does not have a distance of  $2 \times 1 + 1 = 3$ , because it does not correct *any* weight-1 error; only the bit-flips. Since it only corrects all weight-0 errors, the code has distance 0.

## 5.2 Stabilizer codes

The parity check measurements of the bit-flip code can be interpreted as determining if the qubits are in the  $+1$  or  $-1$  eigenspaces of the Paulis  $Z \otimes Z \otimes I$  and  $I \otimes Z \otimes Z$ , for the first and second parity check respectively:  $|00\rangle$  and  $|11\rangle$  are  $+1$  eigenstates of  $Z \otimes Z$ , and  $|01\rangle$  and  $|10\rangle$  are  $-1$  eigenstates. The codewords are  $+1$  eigenstates of all these Paulis, called the *stabilizers* of the code, and if an error<sup>2</sup> occurs, the state of the qubits becomes a  $-1$  eigenstate of one or more of the stabilizers.

This behaviour is not special to the bit-flip code; there are many codes which all of its codewords are a  $+1$  eigenstate of a set of Paulis. This set of Paulis is called the stabilizer of the code. We can define a stabilizer set  $\mathcal{S}$  for a codespace  $\mathcal{H}_k$  (also called the stabilizer of  $\mathcal{H}_k$ ) as the set:

$$\mathcal{S} = \{S \in P_n | S|\psi\rangle = |\psi\rangle \forall |\psi\rangle \in \mathcal{H}_k\}, \quad (5.5)$$

where we thus restrict the elements of  $\mathcal{S}$  to be Paulis. Any stabilizer  $\mathcal{S}$  has two basic properties:

- *The set  $\mathcal{S}$  is closed under multiplication, making it a group.* If  $S_1, S_2 \in \mathcal{S}$ , then  $\forall |\psi\rangle \in \mathcal{H}_k$  we have  $S_1 S_2 |\psi\rangle = S_1 |\psi\rangle = |\psi\rangle$ , so  $S_1 S_2 \in \mathcal{S}$ .

<sup>2</sup>To be precise: an error that maps the state to a space orthogonal to the codespace. If an error maps a codeword to another codeword, it is a logical error, and the resulting state is still a  $+1$  eigenstate of all stabilizers.

- $\mathcal{S}$  is an Abelian group. If  $S_1, S_2 \in \mathcal{S}$ , then  $[S_1, S_2]|\psi\rangle = 0$ . This can only be true if  $[S_1, S_2] = 0$ , since Paulis either commute or anticommute.

Instead of defining a stabilizer in terms of a code, one can also define a codespace in terms of a stabilizer group. For a given stabilizer group  $\mathcal{S}$ , define the codespace as the intersection of the  $+1$  eigenspaces of all elements of  $\mathcal{S}$ . Then, every codeword is an eigenstate of every element of  $\mathcal{S}$ , and every eigenstate of every element of  $\mathcal{S}$  is a codeword. Such a code is called a *stabilizer code*.

A stabilizer code is normally defined by a set of generators  $\mathcal{G}$  for its stabilizer group. The number of logical qubits  $k$  that is encoded in the stabilizer code follows directly from the number of generators. Let  $r = |\mathcal{G}|$  be the number of generators for  $\mathcal{S}$  such that  $|\mathcal{S}| = 2^r$  (note that  $r \leq n$ ). Then the projector  $\Pi_+$  of the codespace can be defined as

$$\Pi_+ = \frac{1}{2^r} \prod_{G \in \mathcal{G}} (I + G) = \frac{1}{2^r} \sum_{S \in \mathcal{S}} S, \quad (5.6)$$

where the last identity follows from the fact that all different products of the generators are all elements of  $\mathcal{S}$ . Since the dimension of a subspace is the trace of its projector, we have

$$\dim(\mathcal{H}_k) = \text{tr}[\Pi_+] = \frac{1}{2^r} \sum_{S \in \mathcal{S}} \text{tr}[S] = \frac{1}{2^r} \text{tr}[I] = 2^{n-r}, \quad (5.7)$$

because every Pauli that is not the identity is traceless, and  $I$  is always in  $\mathcal{S}$ . We thus see that  $k = n - r$ ; every added generator cuts the code space's dimension exactly in half.

### 5.2.1 Errors on stabilizer codes

As discussed in section 5.1, if a set  $\mathcal{E}$  is correctable, then its span is too. Thus, we need only discuss a basis for  $SU(2)$ , and we will use  $\mathcal{P}_n$  as a basis. Since  $\forall P \in \mathcal{P}_n$  we have  $P^\dagger = P$ , any of these errors can be corrected by applying the same Pauli to the erroneous codeword.

An element  $E \in \mathcal{P}_n$  can, with respect to the code  $\mathcal{S}$ , be strictly divided into three different subsets of  $\mathcal{P}_n$ :  $E \in \mathcal{S}$ ,  $E \notin N(\mathcal{S})$  or  $E \in N(\mathcal{S}) \setminus \mathcal{S}$ , with  $N(\mathcal{S})$  the normalizer of  $\mathcal{S}$ . The stabilizer code treats an error depending on in which of these subsets it is, so they are discussed separately below.

#### The set for which $E \in \mathcal{S}$

If  $E$  is an element of  $\mathcal{S}$ , then it acts as a stabilizer. In other words, the error  $E$  acts as the identity on the codewords, and they have no non-trivial effect on the code words. Clearly, for any  $E$  and  $F$  in this set and any pair of codewords  $|\psi\rangle$  and  $|\phi\rangle$ ,  $\langle\psi|EF|\phi\rangle = \langle\psi|\phi\rangle$ , so Eq. (5.1) holds for these errors with  $C(E, F) = 1$ .

#### The set for which $E \notin N(\mathcal{S})$

If  $E \notin N(\mathcal{S})$ , then  $E$  does not commute with at least 1 of the stabilizers. Because Paulis either commute or anticommute, this means that  $\exists S_i \in \mathcal{S}$  s.t.  $\{S_i, E\} = 0$ . Then  $S_i$  acting upon a codeword  $|\psi\rangle$  means that,  $\forall |\psi\rangle \in \mathcal{H}_k$ ,

$$S_i E |\psi\rangle = -E S_i |\psi\rangle = -E |\psi\rangle. \quad (5.8)$$

In other words,  $E|\psi\rangle$  is now a  $-1$  eigenstate of the stabilizer  $S_i$ , regardless of the codeword  $|\psi\rangle$ . In other words,  $E$  now maps the the codespace to some other subspace. The orthogonality of these two subspaces follows from the fact that,  $\forall |\psi\rangle, |\phi\rangle \in \mathcal{H}_k$ ,

$$\langle\psi|E|\phi\rangle = \langle\psi|E S_i |\phi\rangle = \langle\psi|(-S_i E) |\phi\rangle = -\langle\psi|E|\phi\rangle, \quad (5.9)$$

and thus that  $\langle\psi|E|\phi\rangle = 0$ . A measurement of  $S_i$  now projects onto either of the subspaces, and the measurement outcome indicates which one. However, judging by the Knill-LaFlamme criteria for QECCs, this is not a strong enough statement. If two errors  $E$  and  $F$  both have a non-trivial error syndrome, then, if they need to be corrected for differently, they also need to map the codewords to

different orthogonal subspaces. There is a distinction to make between  $E$  and  $F$  having the same or a different error syndrome.

If  $E$  and  $F$  have a different error syndrome, then  $\exists S_i \in \mathcal{S}$  for which  $[E, S_i] = \{F, S_i\} = 0^3$ . Then,  $\forall |\psi\rangle \in \mathcal{H}_k$ ,

$$\langle \psi | EF | \phi \rangle = \langle \psi | EF S_i | \phi \rangle = \langle \psi | E(-S_i F) | \phi \rangle = -\langle \psi | S_i EF | \phi \rangle = -\langle \psi | EF | \phi \rangle, \quad (5.10)$$

which can only hold if  $\langle \psi | EF | \phi \rangle = 0$ , or in other words,  $E$  and  $F$  map the codespace to different orthogonal subspaces. Furthermore, Eq. (5.1) holds, with  $C(E, F) = 0$ .

If  $E$  and  $F$  have the same error syndrome, they map the codespace to the same orthogonal subspace, and the error correction has to be the same for both. We thus have  $[EF, S_i] = 0$  which holds  $\forall S_i \in \mathcal{S}$ . In other words,  $EF \in N(\mathcal{S})$ , and  $E$  and  $F$  are said to be in the same *coset* of  $N(\mathcal{S})$ . Then:

- If also  $EF \in \mathcal{S}$ , they're also in the same coset of  $\mathcal{S}$ , and they effectively cancel each other. Correcting for either one also corrects the other, so we choose the easiest one to implement, and the error is corrected. We have  $\langle \phi | EF | \psi \rangle = \langle \phi | \psi \rangle$ , so Eq. (5.1) holds with  $C(E, F) = 1$ .
- If  $EF \notin \mathcal{S}$ ,  $E$  and  $F$  have the same error syndrome, but they act differently on the codewords. They both map the codespace to the same orthogonal subspace, so  $EF$  maps the codespace into itself. However,  $EF \notin \mathcal{S}$ , so  $\exists |\psi\rangle, |\phi\rangle \in \mathcal{H}_k$  for which

$$EF |\psi\rangle = |\phi\rangle \neq |\psi\rangle. \quad (5.11)$$

$E$  and  $F$  can not be told apart based on the error syndrome, and Eq. (5.11) shows that correcting  $F$  with  $E$  (or vice versa) introduces a *logical error*. The code can not correct both errors, so only one can be in  $\mathcal{E}$ . Furthermore, we also have

$$\begin{aligned} \langle \psi | EF | \phi \rangle &= \langle \psi | \psi \rangle \neq \langle \psi | \phi \rangle, \\ \langle \psi | EF | \psi \rangle &= \langle \psi | \phi \rangle \neq \langle \psi | \psi \rangle, \end{aligned} \quad (5.12)$$

meaning there is no  $C(E, F)$  that does not depend on the codewords for Eq. (5.1) to hold.

### The set for which $E \in N(\mathcal{S}) \setminus \mathcal{S}$

If  $E \in N(\mathcal{S}) \setminus \mathcal{S}$ , then  $\forall |\psi\rangle \in \mathcal{H}_k$  we have  $E |\psi\rangle \in \mathcal{H}_k$ , but  $\exists |\psi\rangle \in \mathcal{H}_k$  such that  $E |\psi\rangle = |\phi\rangle \neq |\psi\rangle$  because  $E$  is not in  $\mathcal{S}$ . In other words,  $E$  is a logical operator for the code. The code can not detect these errors, because there is no non-trivial error syndrome. Furthermore, by the same argument as in Eq. (5.12), with  $F = I$ , there is no  $C(E, F)$  that does not depend on the codewords for Eq. (5.1) to hold.

### 5.2.2 The set of correctable errors $\mathcal{E}$ for a stabilizer code

The total number of correctable errors can now be determined. Every  $P \in \mathcal{P}_n$  can be written as a product of three different elements; let  $C$  be the representative of the coset of  $N(\mathcal{S}) \setminus \mathcal{S}$  that  $P$  is in,  $L$  be the representative of which logical operation  $P$  corresponds to, and  $S$  be any element of the stabilizer. Then we can write

$$P = C \times L \times S. \quad (5.13)$$

For every coset of  $N(\mathcal{S}) \setminus \mathcal{S}$  represented by  $C$  there is one element, up to the stabilizers, that can be corrected, corresponding to a specific  $L$ . There are  $2^r$  different  $C$  for a number of generators  $r$ , corresponding to the different orthogonal subspaces determined by the  $2^r$  different error syndromes. Of course, one of those cosets is the stabilizer itself, corresponding to the trivial error syndrome; for this coset we necessarily choose its  $L$  equal to  $I$ . For every other  $C$  we also choose an  $L$  (normally this is also  $I$ ), and then for every combination  $C \times L$  there is the choice of every different  $S$  that give the same result. There are  $2^r$  elements in the stabilizer  $\mathcal{S}$ , so there are in total  $2^r \times 2^r = 4^r$  correctable

<sup>3</sup>Or for which  $[F, S_i] = \{E, S_i\} = 0$ , but then  $E$  and  $F$  in Eq. (5.10) can be exchanged, or the minus sign there can be introduced one equality later.

errors. Many of these errors have a high weight, because a stabilizer can have a high weight and thus a low-weight  $C \times L$  times this stabilizer then has a high weight as well. These errors are usually very unlikely to occur, and the code is normally not designed with these errors in mind.

Every coset determined by the error syndrome has in total  $4^k = 4^{n-r}$  logical operations (up to the stabilizers), so, counting for the  $2^r$  different cosets and  $2^r$  different stabilizers, there are  $(4^{n-r} - 1) \times 2^r \times 2^r = 4^n - 4^r$  different Paulis that can't be corrected. The union of the correctable and non-correctable errors has of course  $4^n$  elements, and every element of  $\mathcal{P}_n$  is thus accounted for.

### 5.3 Examples of stabilizer codes

The 5-qubit perfect code  $[[5, 1, 3]]$  is the stabilizer code defined by the generators shown in table 5.1a. There are 5 physical qubits and 4 generators, so there is  $5 - 4 = 1$  logical qubit. The set of correctable errors  $\mathcal{E}$  for this code is every single-weight Pauli (and the product of those single-weight Paulis with every stabilizer), so the distance of the code is 3. The code is called perfect because it saturates the quantum Hamming bound [18] and the quantum Singleton bound [18], meaning that there are no distance-3 codes with smaller  $n$ ; there are 15 non-trivial error syndromes, and there are 15 single-qubit errors (A Pauli  $X$ ,  $Y$  and  $Z$  for each of the 5 qubits). The logical  $\bar{X}$  and  $\bar{Z}$  can be implemented by the bottom two Paulis given in table 5.1a; the logical  $\bar{Y}$  is then necessarily equal to  $\bar{X}\bar{Z}$ , which can be implemented by a  $Y$  on every qubit. Note that  $\bar{X}$  and  $\bar{Z}$  do commute with every generator, but not with each other. Furthermore, the choice of  $\bar{X}$  (and thus  $\bar{Z}$ ) is arbitrary:  $\bar{X}$  can be defined to be  $Z \otimes Z \otimes Z \otimes Z \otimes Z$  as well, thereby fixing  $\bar{Z}$  as well.

		$S_1$	$X$	$Z$	$Z$	$X$	$I$			$S_1$	$Z$	$Z$	$Z$	$Z$	$I$	$I$	$I$
		$S_2$	$I$	$X$	$Z$	$Z$	$X$			$S_2$	$Z$	$Z$	$I$	$I$	$Z$	$Z$	$I$
		$S_3$	$X$	$I$	$X$	$Z$	$Z$			$S_3$	$Z$	$I$	$Z$	$I$	$Z$	$I$	$Z$
		$S_4$	$Z$	$X$	$I$	$X$	$Z$			$S_4$	$X$	$X$	$X$	$X$	$I$	$I$	$I$
		$\bar{X}$	$X$	$X$	$X$	$X$	$X$			$S_5$	$X$	$X$	$I$	$I$	$X$	$X$	$I$
		$\bar{Z}$	$Z$	$Z$	$Z$	$Z$	$Z$			$S_6$	$X$	$I$	$X$	$I$	$X$	$I$	$X$
										$\bar{X}$	$X$						
										$\bar{Z}$	$Z$						

(a)

(b)

**Table 5.1:** (a) Generators for the 5-qubit perfect stabilizer code  $[[5, 1, 3]]$ . There are also shown representatives for the logical  $\bar{X}$  and  $\bar{Z}$  operators. (b) Generators for the 7-qubit Steane code  $[[7, 1, 3]]$ . There are also shown representatives for the logical  $\bar{X}$  and  $\bar{Z}$  operators.

The 7-qubit Steane code is  $[[7, 1, 3]]$  code, for which the 6 generators are shown in table 5.1b. It is an example of a CSS code [20], an important class of stabilizer codes. CSS codes, named for its inventors *Calderbank*, *Shor* and *Steane*, are constructed using two classical codes with certain properties. They are characterized by a set of generators that only consist of  $Z$  Paulis ( $S_1$ ,  $S_2$  and  $S_3$  for the Steane code), and a set of generators that only consist of  $X$  Paulis ( $S_4$ ,  $S_5$  and  $S_6$  for the Steane code). Since they are constructed from classical codes, many of their properties are known from the classical codes.

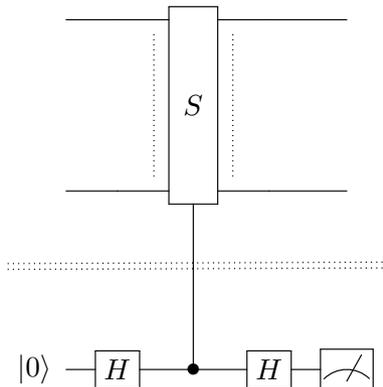
### 5.4 Syndrome measurement

To obtain the error syndrome, all generators have to be measured. The measurement of a stabilizer  $S = P_1^1 \otimes P_2^1 \dots \otimes P_n^1$  means that, for a erroneous codeword  $E|\psi\rangle$ , it must be checked if  $E$  commutes or anticommutes with  $S$ . This has to be done without altering the codeword, or obtaining any information of that codeword (because then the encoded information would be destroyed). Therefore, simply measuring every qubit in the basis dictated by the Paulis  $P_i^1$  is too much: not only is the state of the qubits projected upon either the  $+1$  or  $-1$  eigenspace of  $S$ , but onto all eigenspaces of all  $P_i^1$ s. To project only onto the eigenspace of  $S$  itself, an extra qubit is needed. This qubit is generally called an *ancilla* qubit; qubits on which data is encoded are called *data* qubits. The ancilla qubit is first prepared in the  $|+\rangle$  state, after which the stabilizer  $S$  is applied to  $E|\psi\rangle$ , conditioned on the ancilla

qubit being in the  $|1\rangle$  state. Such a controlled- $S$  gate can be performed by applying a controlled- $P_i^1$  gate to the  $i^{\text{th}}$  qubit, with the ancilla as the control qubit. After rotating the ancilla back to the computational basis, this results in the entangled state

$$\frac{1}{\sqrt{2}}E|\psi\rangle \otimes |+\rangle + \frac{1}{\sqrt{2}}SE|\psi\rangle \otimes |-\rangle = \frac{1}{2}(E|\psi\rangle + SE|\psi\rangle) \otimes |0\rangle + \frac{1}{2}(E|\psi\rangle - SE|\psi\rangle) \otimes |1\rangle. \quad (5.14)$$

Now, since both  $S$  and  $E$  are Paulis, they either commute or anti-commute. If  $[S, E] = 0$ ,  $(E|\psi\rangle - SE|\psi\rangle)$  vanishes, whereas if  $\{S, E\} = 0$ ,  $(E|\psi\rangle + SE|\psi\rangle)$  vanishes. Therefore, after measuring the ancilla qubit in the computational basis, the outcome indicates whether  $[S, E] = 0$  or  $\{S, E\} = 0$ , with the resulting state after the measurement being  $E|\psi\rangle$ , thus leaving the codeword itself intact. See also figure 5.1. This gives a way of measuring the syndrome of the stabilizer  $S$  without affecting the codeword.



**Figure 5.1:** Measurement of stabilizer  $S$  on a set of physical qubits with error  $E$ . An ancilla qubit is prepared in the  $|+\rangle$  state and then a controlled- $S$  operation is performed with the ancilla the control qubit. Just before measurement the (entangled) total state is  $\frac{1}{2}(E|\psi\rangle + SE|\psi\rangle) \otimes |0\rangle + \frac{1}{2}(E|\psi\rangle - SE|\psi\rangle) \otimes |1\rangle$ . Therefore, upon measurement of the ancilla (separated by the dashed horizontal line) the state either collapses to  $(E|\psi\rangle + SE|\psi\rangle) \otimes |0\rangle$  or  $(E|\psi\rangle - SE|\psi\rangle) \otimes |1\rangle$ . If  $[S, E] = 0$ , the latter half vanishes, and if  $\{S, E\} = 0$ , the first half vanishes, so the measurement outcome indicates whether the syndrome is +1 or -1.

The ability to perform syndrome measurements, together with a well designed code, enough and adequately connected qubits, to ability to perform Pauli operations on those qubits, and some various other ingredients, now allows for encoding of any qubit state, which is then protected to any error  $E \in \mathcal{E}$  that occurs on the codewords. It is also possible to perform logical Pauli operations, but we do not know how to perform other logical operations yet. Moreover, if any of these logical operations, or the syndrome measurements themselves, also introduce an error during the operation, there is no guarantee whatsoever that the codespace will stay unaffected. These problems can be very hard to solve, and they are addressed in more detail (and most of them are solved) in chapter 6.

## 5.5 Clifford operators

The set of Clifford operators  $\mathcal{C}_n$  is the subset of  $SU(2^n)$  that is the normalizer of  $\mathcal{P}_n$ :

$$\mathcal{C}_n : \{C \in SU(2^n) | CP_nC^\dagger \in \mathcal{P}_n \forall P_n \in \mathcal{P}_n\}. \quad (5.15)$$

Thus, just as the normalizer of  $\mathcal{S}$  maps codewords to codewords, the Clifford set maps Paulis to Paulis. Furthermore, the Clifford set is a group [18], so it is also known as the *Clifford group*. The Clifford group is important in many different areas of quantum information, including QECCs. For CSS codes, they are easily implementable, and therefore favourable over non-Clifford operations. Furthermore, Clifford operations are exactly the set of operations that are classically efficiently simulatable, by making us of *Gottesman-Knill* or GK-simulation [4], named after its inventors. GK-simulation can simulate highly entangled states, so it shows that the power of a quantum computer is more elaborate than just entanglement. I have implemented GK-simulation during my thesis, but it is outside the scope of this report to discuss it any further.



## Chapter 6

# Fault-tolerant computation

Quantum error-correcting codes offer a way of tracking errors and handling them, without destruction of the encoded quantum information. However, actually implementing a QECC is far from trivial. They can correct errors (that are within the correctable set  $\mathcal{E}$ ) that arise on an encoded qubit not during the actual error correction itself. However, errors can arise both during the syndrome measurements of the error correction scheme, and also when performing logical gates. These errors are influenced by these operations, possibly drastically increasing the error weight through *error propagation*. After only a small number of gates, an error can change into an error with a much higher weight, rendering the encoded data useless. To combat these problems, circuits have to be designed which limit the error propagation, so that QECCs can still work; these circuits are called *fault-tolerant*. In section 6.1 the problem of error propagation is introduced, and section 6.2 discusses the methods devised to combat this problem. Finally, in section 6.3 a small introduction is given to one of the remaining problems, that also introduces the need for the fault-tolerant SWAP operation, which is introduced, experimentally implemented and characterized in chapter 7 and 8.

### 6.1 Error propagation

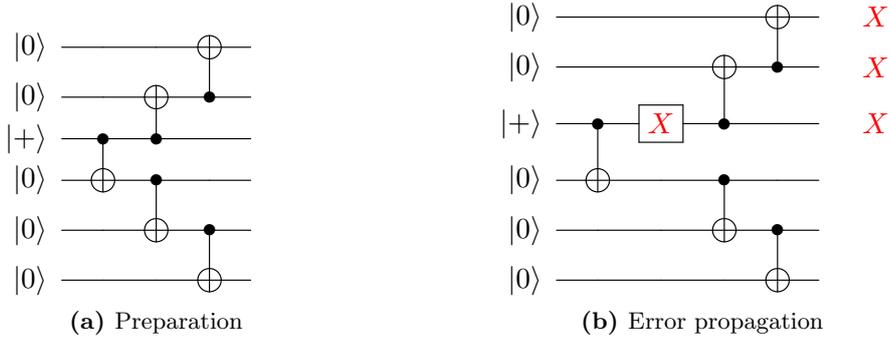
If a codeword  $|\psi\rangle$  has an error  $E \in \mathcal{E}$  on it, there is ostensibly no problem. The error can be determined, and then be corrected for. However, if a unitary operation  $U$  is applied to this erroneous state  $E|\psi\rangle$ , problems may arise. We can write this state as  $UE|\psi\rangle = (UEU^\dagger)U|\psi\rangle$ ; the error  $E$  propagates through the unitary operation and becomes  $UEU^\dagger$ . There are now three different situations:

- $E$  and  $U$  commute. Then  $UEU^\dagger = EUU^\dagger = E$  and the unitary operation leaves the error unaffected. So,  $EU|\psi\rangle = E|\phi\rangle$  is a (different) codestate with the same error  $E$ . Now, if  $E \in \mathcal{E}$  to begin with, this error is still correctable.
- $E$  and  $U$  do not commute, but  $UE = E^*U$  with  $\text{wt}(E^*) \leq \text{wt}(E)$ . Then, after applying  $U$  to the erroneous state  $E|\psi\rangle$ , the state  $E^*|\phi\rangle$  is obtained. For most QECCs, if  $E \in \mathcal{E}$ , then any other error with the same weight will also be in the set of correctable errors; the new error  $E^*$  is therefore still correctable.
- $E$  and  $U$  do not commute, and  $UE = E^*U$  with  $\text{wt}(E^*) > \text{wt}(E)$ . Now after applying  $U$  to  $E|\psi\rangle$  the state  $E^*U|\psi\rangle$  is obtained. There is no guarantee that  $E^* \in \mathcal{E}$ ; new error is now not correctable any more.

The third case only occurs for multi-qubit gates. This is easily seen by considering an operation that consists of tensor products  $U \otimes I$  and noting how it commutes through an error ( $E \otimes I$ ) or ( $I \otimes E$ ). Let  $U$  and  $E$  both be single-qubit operations. Then:

$$\begin{aligned} (U \otimes I)(E \otimes I) &= (UE \otimes I) = (E^*U \otimes I) = (E^* \otimes I)(U \otimes I) \\ (U \otimes I)(I \otimes E) &= (U \otimes E) = (I \otimes E)(U \otimes I), \end{aligned} \tag{6.1}$$

with  $E^*$  necessarily also a single-qubit operation. Clearly, the weight of the propagated errors ( $E^* \otimes I$ ) and ( $I \otimes E$ ) will always have the same weight as the initial errors ( $E \otimes I$ ) and ( $I \otimes E$ ).



**Figure 6.1:** (a) Preparation of the 6-qubit cat state  $|\psi\rangle = \frac{1}{\sqrt{2}}(|000000\rangle + |111111\rangle)$  by making use of multiple CX gates. (b) A single-weight error  $E_1 = IIXIII$  on the third qubit right after the first CX gate (indicated by the red X gate) results in a wt(3) error  $F = XXXIII$  on the output state.

The most notable example of a unitary  $U$  that can increase the error weight after propagation is the CX gate:

$$\begin{aligned} CX(X \otimes I) &= (X \otimes X)CX \\ CX(I \otimes Z) &= (Z \otimes Z)CX, \end{aligned} \tag{6.2}$$

with the CZ gate showing similar behaviour. Note that for a Z error, the extra phase propagates from the target qubit to the control qubit; this is known as phase kickback and is also the basic principle for the stabilizer measurement circuits from section 5.4.

This has strong implications for many different circuits. For instance, in figure 6.1a a circuit can be found that prepares a so called *cat state*<sup>1</sup>. A single error on the third qubit can lead to a much higher weight error on the output qubits, by propagating through the CX gates (figure 6.1b), in this case resulting in a weight 3 error; the resulting state is  $\frac{1}{\sqrt{2}}(|111000\rangle + |000111\rangle)$ .

Another example is the syndrome measurement circuit in section 5.4; if one of the 2-qubit gates that (e.g. the 2-qubit gates that make the controlled- $S$  gate) in the circuit fails, it can invoke an error  $E$  on the qubits it has support on, including the ancilla. Even if the part of the error that has support on the data qubit is in  $\mathcal{E}$ , the error can still propagate via the ancilla through the rest of the circuit and will very often result in a higher weight error on the data qubits, that may not be in  $\mathcal{E}$ . Not only will the error syndrome measurements be wrong, performing these measurements will increase the error weight, rendering the state useless.

This behaviour is not unique to cat state preparation or syndrome measurements; any time a (faulty) multi-qubit gate is applied to two data qubits in the same codeblock, it can introduce an error that might have a weight that is too high for a distance 3 code to handle. Furthermore, error weight can drastically increase after application of multi-qubit gates within the codeblock via propagation as in Eq. (6.2). These problems will arise in any circuit that has these interactions.

## 6.2 Fault-tolerance

Circumventing these problems can be done by carefully designing the circuits. The normal implementations of all elements of the circuit are replaced by sub-circuits that perform the same task, but are designed to limit error propagation. To be able to perform any general quantum computation, implementation of four different components needs to be possible (These components (also called locations) are closely related to the 5 *DiVincenzo criteria* [21]):

- **Preparation of qubits.** Qubits need to be reliably initialized into a state. One state is sufficient, because any other state can be prepared from this state by applying a gate. Normally, the  $|0\rangle$  is prepared. However, it is very helpful to be able to prepare a range of states, for instance the  $|1\rangle$ ,  $|+\rangle$  and  $|-\rangle$  in addition to the  $|0\rangle$  state.

<sup>1</sup>A cat state is a state of the form  $\frac{1}{\sqrt{2}}(|000\dots0\rangle + |111\dots1\rangle)$ , and is also called a generalized or  $n$ -qubit GHZ state.

- **Measurement of qubits.** Qubits need to be reliably measured. Just as with preparation, measurement in only a single basis is sufficient (see section 7.2), but additional bases might be desirable.
- **Logical quantum gates.** A universal set of fault-tolerant quantum gates needs to be possible at the logical level. These gates are fault-tolerant in the sense that they do not give rise to the problems discussed in the previous section.
- **Syndrome measurement and error correction.** To be able to detect and correct errors on the data qubits, the syndromes need to be measured reliably. Moreover, if the syndrome measurement is faulty, the probability that these faults introduce an uncorrectable error needs to be low.

Loosely speaking, whenever a circuit performs all these tasks without increasing error weights, it is said to be *fault-tolerant*. Any sub-circuit that performs any of these tasks is then also fault-tolerant, and it is called a *fault-tolerant gadget*. This is not a strict definition, but more rigorous definitions of fault-tolerance exist [22][19][23][16][18], which are based on the idea that every gadget should do two things [19]: when not too many errors occur on the physical qubits and during the physical gates, and if the input state to the gadget was not too faulty to begin with, then the output state should also not have too many errors, and it should perform the correct logical operation. Quantifying ‘not too many’ and ‘not too faulty’ can be done [19][18], but normally the limit is set to weight-1 errors both on the input state and during the gadget, and the failure of at most one physical gate. This is also how we treat fault tolerance in this thesis:

*If at most one gate in the circuit fails, including 2-qubit gates, thereby introducing a weight  $\leq 2$  error anywhere within the circuit, that error must result in an error of weight  $\leq 1$  on the data qubits. Then we call the circuit fault-tolerant.*

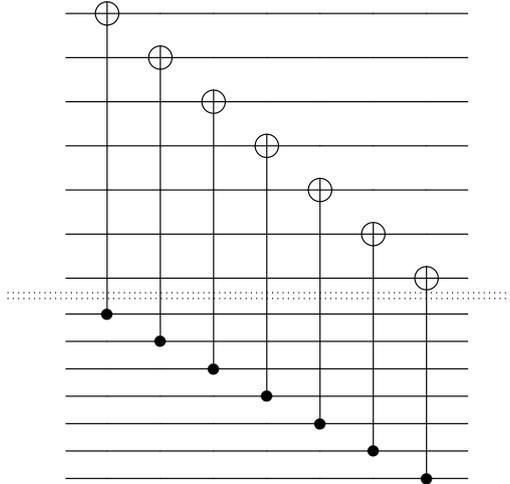
It follows that any gadget consisting of only single-qubit operations is then immediately fault-tolerant. Only whenever a gadget contains at least one multi-qubit operation, the gadget can become non fault-tolerant; it generally will be when no extra care is taken. Furthermore, if any gate fails in a circuit with probability  $p$ , then the multi-qubit error rate must scale with  $p^2$ , essentially suppressing the multi-qubit error rate. This is a direct consequence of the fact that the circuit is designed to limit a single gate failure to weight-1 errors. Higher weight errors are therefore only possible from the failure of at least 2 gates. Furthermore, the definition above does not take into account errors on the input of the circuit; it is assumed to have no errors on the input. This can be realized by the idea of *extended rectangles* [18]; each extended rectangle is a fault-tolerant gadget preceded and followed by fault-tolerant error correction.

Fault-tolerant gadgets for all the different elements in the list at the start of this section are known. These gadgets then can be combined [19][18] into a circuit. That this combination works is proven by the *threshold theorem* [2], which guarantees that if the error rates of the physical elements of a quantum computer are below a certain threshold, fault-tolerant quantum computation will be possible. Below is a short introduction of each of the fault-tolerant implementations of the different locations described in the above list.

**Logical quantum gates** Stabilizer codes have the property that every logical Pauli operation is always implemented by applying a certain Pauli  $P_n \in \mathcal{P}_n$  to the register. Such a  $P_n$  necessarily consists of only single-qubit physical gates, namely single-qubit Paulis. This renders any logical Pauli immediately fault-tolerant. There are many other gates that can be implemented as a tensor product of only single-qubit operations for certain codes, for instance the Hadamard transform and  $S$  gate for the 7-qubit Steane code [19].

However, universal quantum computation can not be achieved with these gates, based on the simple observation that any logical multi-qubit operation necessarily needs to be performed by using at least one physical multi-qubit gate.

If a logical multi-qubit gate has no physical multi-qubit gates with support on multiple qubits within the same codeblock, it is almost fault-tolerant. If furthermore every physical qubit from one of the codeblocks is only connected to one other physical qubit of the other codeblock, and if this behaviour goes both ways (see figure 6.2), then that multi-qubit gate, called a *transversal gate*, is fault-tolerant [19]. This is best seen by the fact that no single failure of a multi-qubit gate can introduce a weight  $> 1$  error within a single codeblock. Note that the Paulis and other tensor products are also transversal gates. Transversal gates are linked to the Clifford gates, for CSS codes the transversal gates are always the Cliffords [18]. Note that the Pauli, Hadamard and  $S$  gates described at the start of this section are also transversal. These transversal gates offer a clean way of performing certain non-Pauli and multi-qubit gates fault-tolerantly, but it has been shown [24] that no universal set of gates can be reached this way. Therefore, there necessarily also must be non-transversal gates.



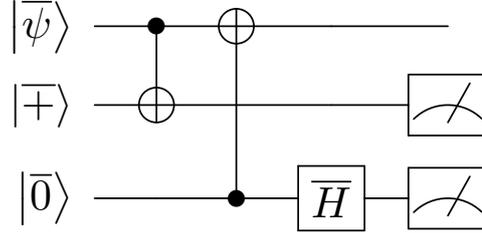
**Figure 6.2:** The logical CX operation between two codeblocks (separated by the dashed lines) both encoding the 7-qubit Steane code. The operation is implemented by applying 7 CX-gates from the  $i$ th physical qubit in the codeblock of the control logical qubit to the  $i$ th physical qubit in the codeblock of the target logical qubit. Because no single physical qubit is connected to more than one other physical qubit, and because no two physical qubits from the same codeblock are connected, this gate is transversal, and therefore fault-tolerant.

Non-transversal gates can be implemented fault-tolerantly using a method known as *gate teleportation* [25]. In gate teleportation, so called *magic states* [26] are used to perform the gates not on the data qubits themselves. By teleporting the data qubits to a different logical qubit, the gate that needs to be implemented is transformed into a different gate that has to be applied to this new logical qubit. By careful design, this different gate can be easier to implement; generally they can be implemented as transversal gates. Gates that are transversal to begin with are easier to implement than gate teleportation gadgets, so it is favourable to have as few non-transversal logical gates in a code as possible. The Solovay-Kitaev theorem [5] guarantees that for many codes one non-transversal gate is enough [19]. Note that the magic state needs to be prepared fault-tolerantly for gate-teleportation to be fault-tolerant.

**Error correction** As mentioned above, the method of syndrome measurement described in section 5.4 is not fault-tolerant. However, by making use of cat states as the ancilla, every data qubit has a single corresponding ancilla, and the readout can be done transversally. Controlled gates from the non fault-tolerant method from section 5.4 are distributed over all the ancilla qubits transversally. Then, by measuring every ancilla in the Hadamard basis, the outcome of the measurement is encoded in the parity of the resulting bitstring of the measurement. This method is not fault-tolerant on its own, but it is the core of a fault-tolerant protocol known as Shor error correction [27][19]. Shor error correction is very costly, and improvements exist for certain codes that make use of properties of those codes. Most notable are Steane error correction [19] and Knill error correction [19].

Steane error correction works only for CSS codes, by making use of the transversality of the CX gate. Two CX gates are used, one through which bit-flips ( $X$ ) errors on the data qubits can propagate

to a logical ancilla prepared in the  $|\overline{+}\rangle$  state, and one through which phase-flips ( $Z$ ) on the data qubits can propagate to a logical ancilla prepared in the  $|\overline{0}\rangle$  state; see figure 6.3. Non-trivial outcomes of transversal measurements of the logical ancillae then indicate the presence of  $X$  and  $Z$  errors; the error syndrome tells which one. This method needs fault-tolerant preparation of the logical  $|\overline{0}\rangle$  and  $|\overline{+}\rangle$  states on the ancillae, so that phase- and bit-flip errors don't propagate back to the data qubits, through the first and second logical CX gates, respectively.



**Figure 6.3:** Steane error correction. Every line represents a full code block. The first (transversal) CX gate allows for bit-flip errors on the data block to propagate to the first ancilla block; the second CX gates allows for phase-flip errors on the data block to propagate to the second ancilla block. Non-trivial answers of both transversal measurements indicate the syndrome of the error on the data qubit. The preparation of the logical ancillae need to be performed fault-tolerantly, so that  $Z$  and  $X$  errors can't propagate to the data qubits via the first and second CX gates, respectively.

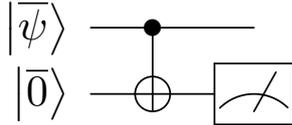
Apart from that there are *Flagged circuits* [28][29] that make use of normal syndrome measurements with an extra ancilla. This extra ancilla, called the flag, is entangled with the read-out ancilla, and measured together with that qubit. A non-trivial outcome of the flag measurement (a raised flag) indicates the failure of a multi-qubit gate in the syndrome measurement circuit. Furthermore, a raised flag together with the resulting syndrome indicates the added error on the data qubits, which then can be corrected for. These flags can also be used in other circuits, creating fault-tolerant gadgets for certain codes [30].

**Measurement** Measuring a logical qubit in the  $|\overline{0}\rangle, |\overline{1}\rangle$  basis is performing a measurement of the observable  $\overline{Z}$ , the logical  $Z$  operator. Any logical operator of a stabilizer code is a Pauli operator, and by using the syndrome measurement schemes of the previous section, these operators can be measured fault-tolerantly; it is effectively adding one extra stabilizer on top of the initial stabilizer group of the code. By first performing error correction and then using the readout method with cat states from the error correction schemes, the added stabilizer can be measured fault-tolerantly; first performing error correction is needed to fulfil one of the fault-tolerant criteria [18].

For CSS codes, a transversal physical measurement of every qubit in the  $Z$  basis also measures the logical qubit in the logical  $\overline{Z}$  basis. However, this collapses the data qubits to a separable state (which is not a codeword of the CSS code), thereby performing a *destructive* measurement; this measurement doesn't project upon either the logical  $|\overline{0}\rangle$  or  $|\overline{1}\rangle$ , but essentially destroys the qubit. It is, however, a fault-tolerant measurement [18].

By use of the transversal CX gate, the data qubits can be entangled with an ancilla register prepared in the  $|\overline{0}\rangle$  state (encoded in the same CSS code as the data), creating the state  $\alpha |\overline{00}\rangle + \beta |\overline{11}\rangle$ . By measuring the logical ancilla qubit transversally (and thus destructively), the logical data qubit is projected onto the  $\overline{Z}$  basis, with the outcome of the transversal ancilla measurement indicating the logical measurement outcome. Just as with Steane error correction, this technique needs fault-tolerant preparation of the logical  $|\overline{0}\rangle$  state.

**Preparation of qubits** Preparation of logical data- and ancilla qubits can be done with different methods. The transversal CX gates of CSS codes allow to perform a transversal check for errors on a non fault-tolerant preparation of the logical  $|\overline{0}\rangle$  state [18]. This transversal check does not tell what error has happened, but by repetition of the (possibly faulty) preparation and then performing the transversal check, a fault-tolerant state can be prepared.



**Figure 6.4:** Steane measurement of the state  $|\bar{\psi}\rangle = \alpha|\bar{0}\rangle + \beta|\bar{1}\rangle$ ; every line represents a full code block. The (transversal) CX gate creates the state  $\alpha|\bar{00}\rangle + \beta|\bar{11}\rangle$ ; transversal destructive measurement of the logical ancilla qubit then projects the logical data qubit onto the  $\bar{Z}$  basis.

Any logical state can in principle be prepared by adding a logical as an extra generator for the stabilizer group and measuring that stabilizer. However, fault-tolerant stabilizer measurements generally require fault-tolerant preparation of some specific ancilla state, creating a vicious circle. There are ways to circumvent this, for example making use of Shor error correction and then preparing the cat states by verifying them with only a bare ancilla. However, these methods can become increasingly costly and hard to implement.

Finally, to prepare magic states fault-tolerantly, one can make use of a technique called *magic state distillation* [26], where from multiple faulty magic states one less faulty magic state can be prepared. By iterative application of the technique, a magic state with arbitrary small error can be obtained, but the method described in [26] makes use of 5 faulty states per iteration, so the total number of states quickly increases as the number of iterations becomes bigger.

### 6.3 Connectivity between qubits

In the above introduction of fault-tolerance, we assume that any pair of qubits can be subjected to a 2-qubit gate. When this is possible for a pair, the two qubits are said to be connected. In a physical realization of a quantum computer, connection between all pairs of qubits will generally not be possible; to perform a multi-qubit gate on an arbitrary pair of qubits, they first need to be physically connected. This can be achieved by swapping qubit states with their (connected) neighbours, and doing this consecutively [18] until the qubits are indeed connected. The (necessarily multi-qubit) SWAP gate is not fault-tolerant, but a fault-tolerant SWAP gate that makes use of a single ancilla exists [16][18]. This gate will be introduced in chapter 7, and this gate is characterized and experimentally realized in this thesis, of which the results can be found in chapter 8.

## Chapter 7

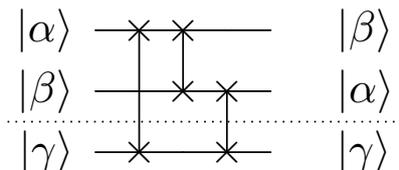
# The fault-tolerant SWAP circuit

As explained in the last section of chapter 6, in any realistic computer architecture qubits need to be swapped reliably to perform computations on them. A SWAP operation is of course a 2-qubit gate, but when two qubits in the same codeblock are swapped, this implementation is not fault-tolerant: the 2-qubit operation is able to introduce a weight-2 error on the two data qubits. To SWAP qubits reliably the a fault-tolerant SWAP gadget is needed. This chapter introduces a fault-tolerant SWAP circuit, which is a fault-tolerant gadget that performs the SWAP operation. It is this circuit for which tomography experiments are performed to analyse its fault-tolerant properties.

The circuit is introduced and explained in section 7.1. When implementing the tomography experiments on an actual chip, some considerations need to be made, which are discussed in section 7.2. To better understand the circuit, and to get an indication of what the actual experiments will give as results, section 7.3 contains simulations on the fault-tolerant and normal circuit, specifically of the tomography experiments and the error maps that will be introduced by faulty gates. These simulations give insight in the tomography experiments and how the estimation methods for  $\chi$  influence the fidelity  $F_p$  and the error ratio  $r$ , including the CP subroutine from section 4.3.2.

### 7.1 The fault tolerant SWAP operation

When the SWAP operation is applied to two qubits, it can induce a weight-2 error on these qubits. Generally speaking, a distance-3 code will not be able to correct this error, rendering the SWAP operation a non-fault tolerant gate, if it is applied to two qubits in the same code block. This problem can be solved by introducing an ancilla qubit that acts as an intermediary qubit for the SWAP operation; the circuit for this can be found in figure 7.1. The two data qubit states are never swapped directly, meaning that there is never a single 2-qubit gate that is applied to both states directly. Therefore, no single faulty 2-qubit gate can introduce a weight-2 error on the data qubits. Furthermore, a perfect SWAP operation never increases the error weight of an incoming error: the operation exchanges the qubit states, including potential errors. If then at most 1 gate is faulty, thereby introducing an error with weight 2 or below, the use of the ancilla guarantees a weight  $\leq 1$  error on the data qubits. This renders this implementation fault-tolerant.



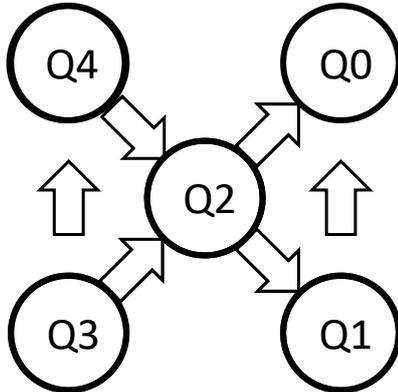
**Figure 7.1:** The fault-tolerant SWAP operation. By using a 3rd qubit as an ancilla (separated by the dashed line), any single weight-2 gate error will result in at most a weight-1 error on the data qubits. This is a direct consequence of the fact that no single weight-2 operation is applied to both data qubits directly.

To compare the fault-tolerant implementation with the normal SWAP gate and investigate the error maps induced by both implementations, the circuits are characterized as channels using quantum process tomography. The tomography is only performed on the data qubits, because the state of the

ancillary qubit is not of interest. Note that the ancillary qubit is always prepared in the  $|0\rangle$  state and that it is never measured. From the obtained representations of the error maps, by taking the Pauli twirl, the error weights can then be estimated, enabling a comparison of the fault-tolerant and normal SWAP operation. How to implement the operations on a physical chip is discussed in the next section.

## 7.2 Implementation on a quantum chip

The tomography experiments are implemented on the 5-qubit ibmqx4 device ‘Tenerife’ [3] using the IBM Q Experience and qiskit [31]. The device layout can be found in fig. 7.2. The five qubits are represented by the 5 circles, and each arrow in between a pair indicates that a CX operation can be performed on those two qubits, with the arrow pointing from the control- to the target-qubit. Note that there is triangular connectivity: there are two sets of three qubits which are all directly connected to each other; this connectivity is required for the fault-tolerant SWAP gate. The device characteristics are shown in table 7.1. In table 7.1a the relaxation time  $T_1$ , dephasing time  $T_2$ , average gate error  $E_g$  and average readout error  $E_r$  of all 5 qubits are shown. In table 7.1b the average errors of the 2-qubit CX gates are shown.



**Figure 7.2:** Layout of the ibmqx4 5-qubit chip on which the tomography experiments are run. Each circle represents a qubit, and each arrow indicates that a CX operation can be performed, with the arrow pointing from the control to the target qubit.

	Q0	Q1	Q2	Q3	Q4
$T_1$ ( $\mu s$ )	42.6	44.1	33.5	44.9	51.6
$T_2$ ( $\mu s$ )	36.7	15.6	21.0	14.9	11.2
$E_g$ ( $10^{-3}$ )	0.77	6.35	1.16	2.06	1.12
$E_r$ ( $10^{-2}$ )	7.0	8.3	3.5	2.3	5.3

(a)

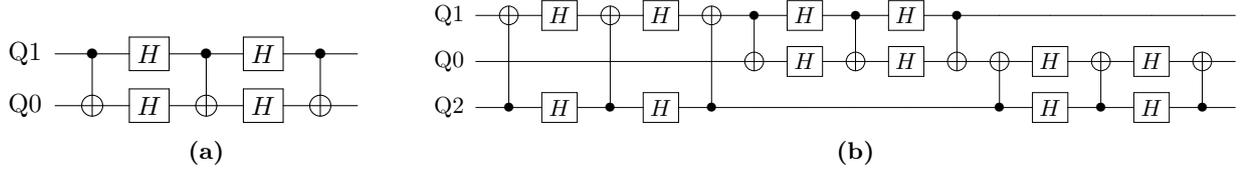
	Q0	Q1	Q2	Q3	Q4
Q0	x	x	x	x	x
Q1	0.034	x	x	x	x
Q2	0.030	0.045	x	x	x
Q3	x	x	0.059	x	0.042
Q4	x	0.056	x	x	x

(b)

**Table 7.1:** (a) Qubit characteristics for the ibmqx4 5-qubit device provided by the IBM Q Experience. The relaxation time  $T_1$ , the dephasing time  $T_2$ , the average gate error rate  $E_g$  and the average readout error rate  $E_r$  are shown. (b) Multi-qubit error rates for the ibmqx4 5-qubit device provided by the IBM Q Experience. Every row is a control qubit (C), every column is a target qubit (T).

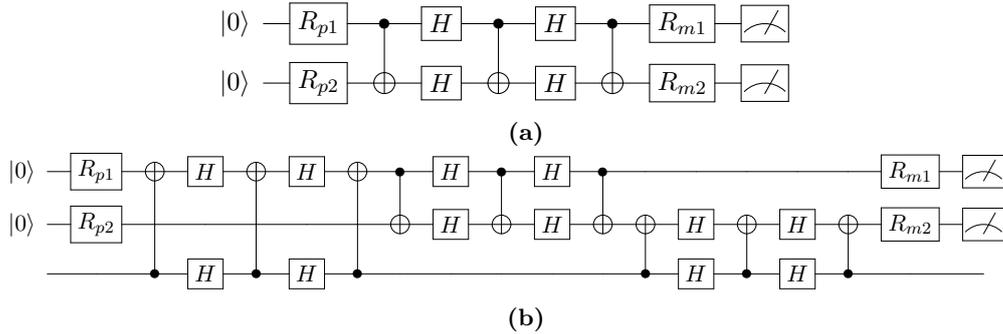
As can be seen in fig. 7.2, any pair of connected qubits has only one direction in which the CX gate can be implemented natively. However, the action of the CX gate on two qubits is reversed if both

qubits are rotated to the Hadamard basis. This can be easily verified by calculating the composite gate as a matrix product of the individual gates. The SWAP gate can then be compiled as three CX gates with four Hadamard gates, as shown in fig. 7.3.



**Figure 7.3:** (a) The normal, non fault-tolerant SWAP and (b) fault-tolerant SWAP gate as implemented on the ibmqx4 device. The SWAP operation can not be implemented natively, so it is decomposed to three consecutive CX operations with the control qubit flipped for the 2nd CX gate. However, the CX operation can only be implemented in one direction on the IBM device, so the 2nd CX gate is flipped by changing to the Hadamard basis for both qubits. For the normal SWAP operation, the top qubit is implemented on Q1, with the bottom on Q0. For the fault-tolerant SWAP, the ancilla at the bottom is implemented on Q2, so that the two data qubits (and thus tomography qubits) are the same for both experiments.

Furthermore, measurements on the ibmqx4 device are always in the computational basis. To perform a measurement in another basis  $\{|\alpha\rangle, |\beta\rangle\}$  (that is, a measurement of an observable  $M = |\alpha\rangle\langle\alpha| - |\beta\rangle\langle\beta|$ ), the qubit is first rotated from that basis to the computational basis by applying a unitary  $R_m = |0\rangle\langle\alpha| + |1\rangle\langle\beta|$ . The measurement outcome  $+1$  and  $-1$  then correspond to the  $|\alpha\rangle$  and  $|\beta\rangle$  states, respectively. Likewise, qubits are always initialized in the  $|0\rangle$  state. To prepare a state  $|\alpha'\rangle$  on a qubit, it is rotated from the computational basis by applying a unitary  $R_p = |\alpha'\rangle\langle 0| + |\beta'\rangle\langle 1|$ , with  $|\beta'\rangle$  orthogonal to  $|\alpha'\rangle$ . A schematic of the full tomography experiment can be found in fig. 7.4.



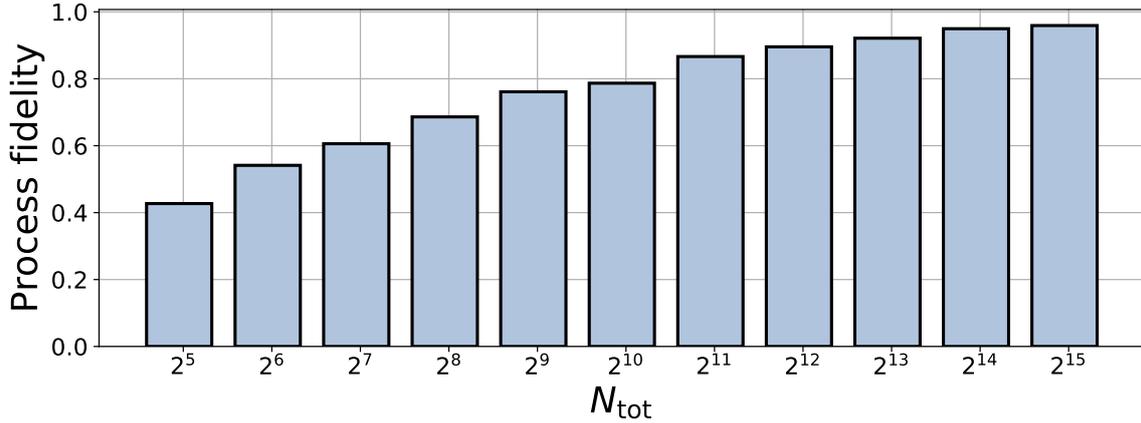
**Figure 7.4:** The full tomography experiment of (a) the normal SWAP gate and (b) the fault-tolerant SWAP gate as implemented on the ibmqx4 chip. The two data qubits are initialized in the  $|0\rangle$  states and then rotated to the various Pauli eigenstates, indicated by the  $R_p$  gates. After the circuit, the two data qubits are rotated again to perform a measurement in the various Pauli bases; this rotation is done using the  $R_m$  gates.

## 7.3 Simulations of the experiments and circuits

To understand the tomography experiments and the two circuits before implementing them on the actual chip, we perform two different simulations. The tomography experiments are simulated as a whole; this allows for understanding of the influence of statistical errors and the CP subroutine from section 4.3.2 on the outcome of the tomography. Apart from that, the complete maps as open quantum system channels of both the fault-tolerant and normal SWAP operation are calculated, with the perfect gates replaced by faulty implementations. This allows for an estimation of the actual experiments and can act as a check of the outcomes of those experiments.

### 7.3.1 Modelling the tomography experiment without errors

The simulations of the tomography experiments are performed in qiskit [31], using the off-site ibmq simulator backend. Errors and faulty gates are not included in the simulations, so that the only



**Figure 7.5:** Process fidelity  $F_p$  of the estimated process matrix  $\chi$  from the simulations. The total number of shots  $N_{\text{tot}}$  is varied from  $2^5 - 2^{15}$ . The infidelity is caused by the CP subroutine, that maps the estimated  $\chi$  to the space of CP maps.

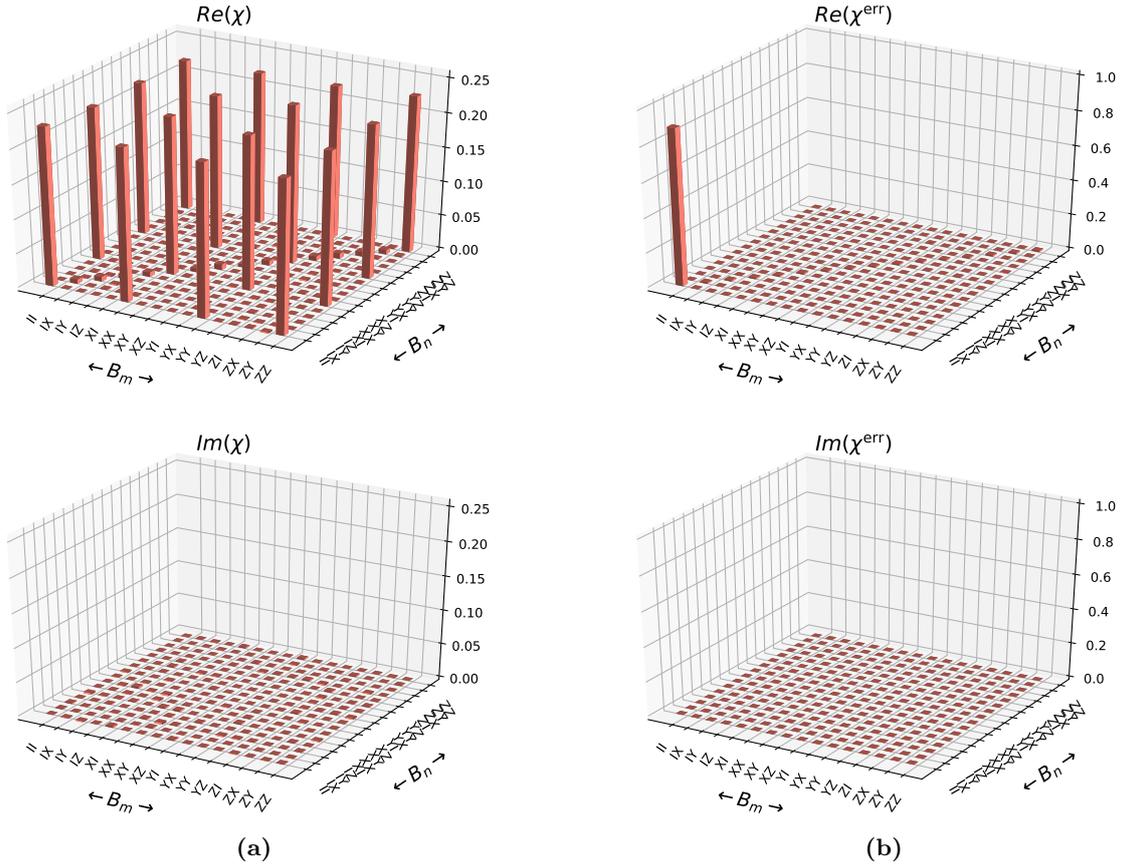
infidelity on the estimated  $\chi$  is a result of finite sampling and its consequences, most notably the CP subroutine. This also means that the normal and fault-tolerant SWAP circuits are exactly the same for these simulations; therefore the simulations are only performed for one of the two experiments. The fidelity of the estimated process is calculated using the  $\chi^{\text{err}}$  method from section 4.4. In figure 7.5 the fidelities of the simulations can be found with varying total number of shots  $N_{\text{tot}}$ . From  $N_{\text{tot}} = 2^{14} = 16384$ , the minimum eigenvalue of the estimated  $\chi$  is positive, discarding the use for the CP subroutine. The maximum by the IBM Q Experience allowed  $N_{\text{tot}}$  for a single experiment is  $2^{13} = 8192$ . The estimate of  $\chi$  from simulations with that  $N_{\text{tot}}$  can be found in figure 7.6.

The process fidelity of the estimated  $\chi$  is  $F_p = 0.92718$ . Note that the diagonal clearly indicates the use of the CP subroutine from section 4.3.2. The elements on the diagonal indicate that the initial estimated  $\chi$  had a minimum eigenvalue of  $\lambda_{\min} = -5.26 \times 10^{-3}$  and a maximum eigenvalue of  $\lambda_{\max} = 1$ , resulting in a maximum eigenvalue after the CP subroutine of  $\xi_{\max} = \frac{1+5.26 \times 10^{-3}}{1+16 \times 5.26 \times 10^{-3}} = 0.92719 \approx F_p$ , which is to be expected based on the arguments in section 4.4.1 and the discussion of the CP subroutine in section 9.1. Furthermore, with error processes absent, the process fidelity should be 1 (the largest eigenvalue before the CP subroutine is  $1 - 9 \times 10^{-12}$  reflecting that). Therefore, for tomography experiments on a perfect device, the CP subroutine can bring down the estimated fidelity by more than 7% for a total number of shots  $N_{\text{tot}} = 8192$ .

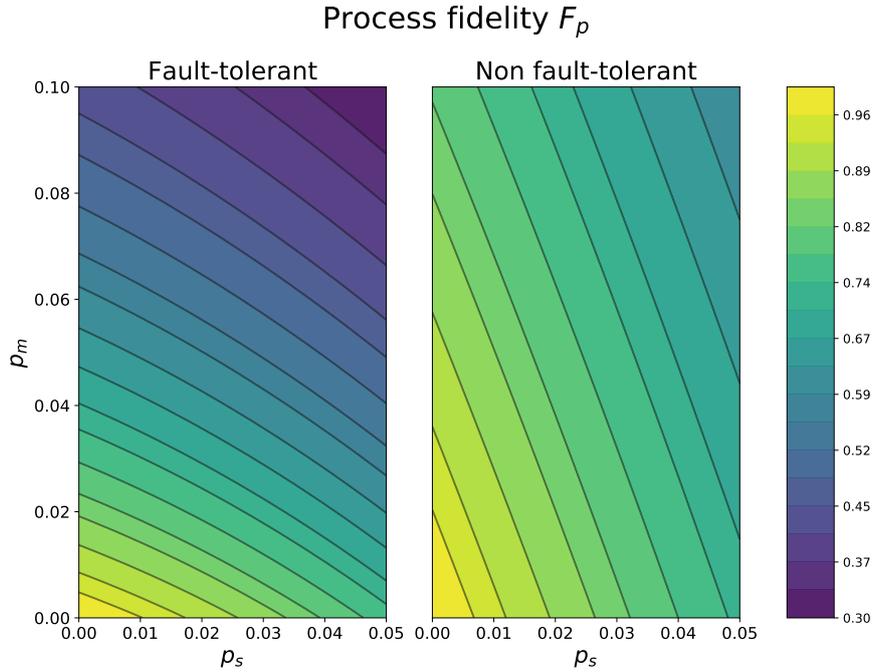
### 7.3.2 Modelling the circuit with faulty gates

The calculations of the circuits as linear maps are performed using numpy [32]. We represent each faulty gate as a perfect unitary operation followed by a depolarizing channel; for the 2-qubit gates this is also the 2-qubit depolarizing channel, see section 3.5.3. An identity or wait location is also modelled with a depolarizing channel. Every single-qubit gate is modelled to have the same error rate  $p_s$ , and every multi-qubit gate is modelled to have the same error rate  $p_m$ , meaning that single-weight and multi-weight errors occur with the same probability  $\frac{p_m}{15}$ . Note that these simulations are not of the total tomography experiment, but directly calculate  $\Lambda$  itself as the composition of the maps corresponding to the individual gates. Therefore, any error within the final map is a direct indication of the errors from the faulty gates, and there are no statistical errors present.

The superoperator representation (see section 3.2.4) is used to perform these calculations. For the fault tolerant circuit, the partial trace (see 3.4) of the final 3-qubit map is taken to calculate the map on the 2 data qubits, which will also be obtained by the actual tomography experiments. The process matrix  $\chi$  of the complete circuit is then calculated from the final composite superoperator. Subsequently, we calculate the process fidelity of the two circuits when compared to the perfect SWAP operation from the  $\chi$  matrix using the  $\chi^{\text{err}}$  method (see section 4.4.1). Furthermore, from the  $\chi$  matrices we calculate the ratio  $r = \frac{s_{2\pm}}{s_1}$  of the multi- and single-weight errors by first taking the Pauli twirl (see section 3.6.1).



**Figure 7.6:** Results of tomography simulations of the fault-tolerant SWAP circuit with  $N_{\text{tot}} = 8192$ . **(a)** City bar plot of the process matrix  $\chi$  that is reconstructed from the tomography data. There are no error processes or faulty gates simulated, the errors in this estimate are therefore completely due to statistical noise. The fidelity of this process compared to the perfect SWAP operation is  $F_p = 0.927$ . Note that there is a constant diagonal term in the plot of  $\chi$ , indicating that the initial estimated process matrix did not reflect a CP map and that the CP subroutine has been activated, which is also the source of the considerable infidelity. Since there are no errors simulated whatsoever, simulations of the non-fault-tolerant SWAP circuit would give identical results. **(b)** City bar plot of the error matrix  $\chi^{\text{err}}$ . The only error in the process is a consequence of the CP subroutine, that shows as the same constant diagonal as in the process matrix (The  $z$ -axis scaling is different in plot (a) and (b) though.).



**Figure 7.7:** Contour plot of the fidelities  $F_p$  of simulations of a faulty circuit for both the fault-tolerant (left) and normal (right) SWAP circuit, without any sampling noise. Every gate in the circuit is replaced by a faulty implementation. Each faulty gate is modelled as a channel which first implements the perfect unitary operation, and then a depolarizing channel. The 1- and 2-qubit gates have a 1- and 2-qubit depolarizing channel, respectively. The single-qubit gates are all modelled with an error rate  $p_s$ , and the multi-qubit gates are modelled with an error rate  $p_m$ . Fidelities of both maps when compared to the perfect SWAP operation are shown. These calculations are performed for a  $35 \times 35$  grid of  $p_s \in [0, 0.05]$ ,  $p_m \in [0, 0.1]$ , resulting in a 2D-landscape of fidelities.

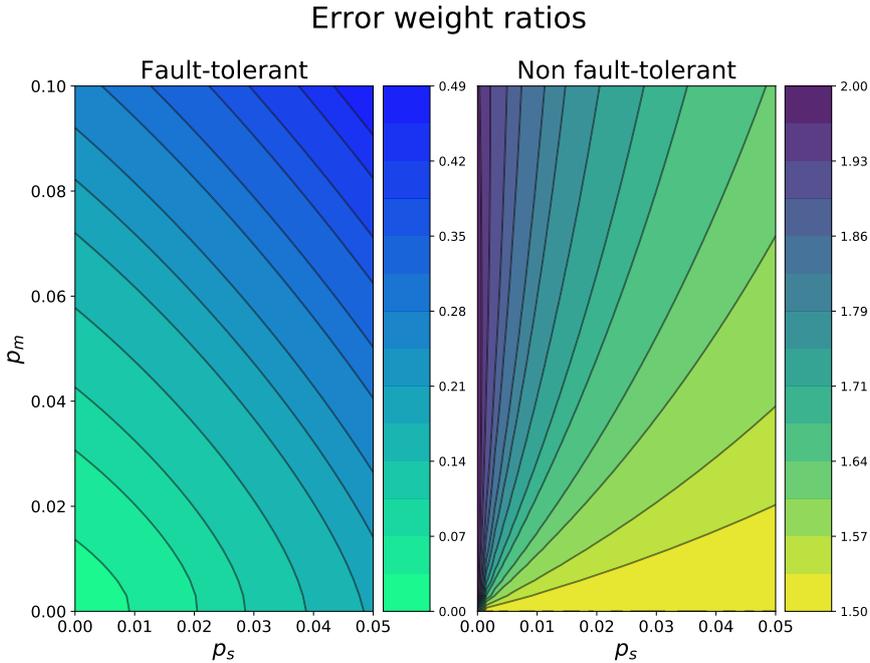
These calculations are performed for a  $35 \times 35$  grid of  $p_s \in [0, 0.05]$ ,  $p_m \in [0, 0.1]$ , resulting in a 2D-landscape of the fidelities (see figure 7.7) and ratios (see figure 7.8). The IBM Q experience gives estimates of the single- and multi-qubit average error rates (see table 7.1); using these an estimate can be made of the fidelities of the actual tomography experiments. Based on this model, with  $p_s = 6 \times 10^{-3}$  and  $p_m = 0.045$ , the highest values of the qubits and gates involved, a very crude estimate of the fidelities is:  $F_p = 0.8$  for the fault-tolerant circuit and  $F_p = 0.85$  for the non-fault tolerant circuit.

From fig. 7.7 it is evident that, for a fixed  $p_s$  and  $p_m$ , the fault-tolerant implementation has a significantly lower fidelity than its normal counterpart. This is to be expected because of the higher gate count of the fault-tolerant circuit, where every gate adds to the total error map. Furthermore, the fault tolerant implementation is more sensitive to  $p_m$  for a fixed  $p_s$  than the normal circuit. Moreover, the gradient of the fault-tolerant circuit is also much steeper; a equal change in  $p_s$  or  $p_m$  has a much stronger effect on the fidelity for the fault-tolerant circuit. By taking the fidelity of every gate as  $(1 - p)$  and accounting for various cancellations and error propagations, the fidelities of the fault-tolerant and non-fault-tolerant circuit can be approximated as:

$$\begin{aligned}
 (\text{Non-FT}) \quad F_p &\approx (1 - p_s)^4 \times (1 - p_m)^3, \\
 (\text{FT}) \quad F_p &\approx \left(1 - \frac{9}{10}p_s\right)^{15} \times \left(1 - \frac{3}{8}p_s\right)^{12} \times \left(1 - \frac{12}{45}p_m\right)^9,
 \end{aligned} \tag{7.1}$$

of which simulations can be found in appendix B.1. This approximation neglects various higher-order terms and error cancellations, so especially for larger  $p_s$  and  $p_m$  it deteriorates.

The ratio  $r$  is considerably lower for the fault-tolerant circuit. From figure 7.8 it is evident that as  $p_s$  and  $p_m$  approach zero the ratio also approaches zero; this is the property that the gadget is designed to have. The ratio steadily increases for both an increasing  $p_s$  and an increasing  $p_m$ ; there is no significant difference between the effect of a larger  $p_s$  and  $p_m$ . This reflects the fact that 2-qubit errors



**Figure 7.8:** Contour plot of the multi- to single-qubit error ratios  $r = \frac{s_2+}{s_1}$  of the Pauli twirl of the total error map calculated for both the fault-tolerant(left) and normal(right) SWAP operation, for the simulations described in section 7.3.2. Note the use of different colorbars due to difference in ratios between the fault-tolerant and normal operation; the ratio of the fault tolerant operation is significantly smaller than the normal implementation. Furthermore, note that the CP subroutine is not included in these simulations.

can become single-qubit errors after propagation and vice-versa; the 2-qubit gates and single-qubit gates can both introduce weight-1 and weight-2 errors, but never a weight-2 error on the data-qubits. The normal circuit shows significantly different behaviour regarding the ratios. For small  $p_s$  and  $p_m$ , there is only minute dependence on the other parameter. Furthermore, it has a lower limit of 1.5 and an upper limit of 2.0; the lower limit is of particular interest, because the goal of the fault-tolerant gadget is for its ratio to be lower than this value. In this lower limit, where  $p_m = 0$ , there are no direct 2-qubit errors, so every multi-qubit error is a direct results of error-weight increase by propagation.

The CP subroutine can be interpreted as adding a uniform error to the process; this will affect the error ratio. The CP subroutine adds the identity matrix to  $\chi$ , and therefore also to  $\chi^{\text{err}}$ . The ratio  $r^{\text{CP}}$  after the CP subroutine is then, for a 2-qubit process:

$$r^{\text{CP}} = \frac{r(1 - F_p) + 9|\lambda_{\min}|(r + 1)}{(1 - F_p) + 6|\lambda_{\min}|(r + 1)}, \quad (7.2)$$

with  $r$  the ratio before the CP subroutine and  $\lambda_{\min}$  and  $F_p$  the minimum eigenvalue and fidelity of  $\chi$  before the CP subroutine. This means that the ratio goes to 1.5 from both below and above, reaching that value in the limit of large  $|\lambda_{\min}|$ . Hence, the fault-tolerant circuit's ratio will go up, and the non fault-tolerant circuit's ratio down, making the difference between the two ratios less defined. Nevertheless, as long as  $|\lambda_{\min}| \ll 1$ , the ratio will not be gravely affected by the CP subroutine, which is the same as the influence on the fidelity.

These simulations and calculations reflect the overall properties that we expect from the fault-tolerant gadget compared to the normal implementation. The fidelity of the fault-tolerant gadget will be lower, but the error map will consist of considerably fewer weight-2 errors when compared to the normal SWAP operation. Furthermore, a perfect fidelity will be impossible to obtain with the estimation methods used, especially the CP subroutine. That routine will also affect the emerging ratio of multi- and single-weight errors. However, these implications can be diminished by using a larger  $N_{\text{tot}}$ , and the simulations show that the actual values can still give decisive results.



# Chapter 8

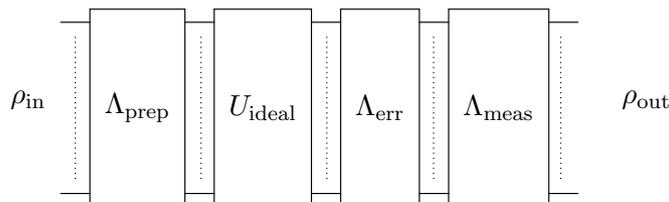
## Results of tomography experiments

The main results of the tomographic experiments are obtained from the reconstructions of  $\chi$  and the subsequent data analysis that is performed on them. The direct results after the tomographic methods give unfavourable results. As can be found in table 7.1, the measurement error rates of the ibmqx4 device are relatively high. The methods introduced in chapter 4 do not account for SPAM errors, but to obtain a better representation of the actual circuit these need to be treated. Section 8.1 introduces a method of filtering SPAM errors that assumes measurement errors are the main contributor to SPAM errors, and that preparation errors are negligible. The method is based upon characterizing the induces measurement errors of the tomography by doing full tomographic characterisation of the Identity channel and contributing all errors in this channel to measurement errors. Then, the main results of the experiments and data analysis can be found in section 8.2, along with a short discussion. The full discussion and conclusions can then be found in chapter 9.

### 8.1 Preliminary results and SPAM errors

The tomography methods described in chapter 4 and 7 perform reconstruction of  $\chi$  from the measurement data; and indicate what can be expected from the experiments. However, they do not take SPAM errors (see section ) into account. The measurement error rates provided by the IBM Q Experience (see table 7.1) [3] indicate that this results in a significantly low process fidelity for both experiments:  $F_p^{\text{FT}} = 0.504$  for the fault-tolerant SWAP and  $F_p^{\text{NFT}} = 0.730$  for the normal SWAP operation. The corresponding process- and error matrices are not shown here, but are included in appendix B.2.

To combat the SPAM errors, the whole experiment, including these errors, is modelled as a composition of maps. Assuming that the preparation errors are negligent compared to the measurement errors, the full experiment can be modelled as the composition of three maps (see figure 8.1): the perfect unitary  $U$ , the error map  $\Lambda_{\text{err}}$  induced by the SWAP circuit itself, and the error map induced by the measurements  $\Lambda_{\text{meas}}$ . Process tomography then estimates this total composition, represented by a process matrix  $\chi^{\text{total}}$ . If a representation of  $\Lambda_{\text{meas}}$  is known or estimated, it can be used to calculate (a representation of)  $U$  and  $\Lambda_{\text{err}}$  together, which is the desired  $\chi$ . This only works if the corresponding superoperator of  $\Lambda_{\text{meas}}$  is invertible.



**Figure 8.1:** Model of treating SPAM errors as an unknown error map before and after the circuit, making the full circuit a composition of maps. Furthermore, we assume that  $\Lambda_{\text{prep}} \approx I$ , so that the SPAM errors consist of only the measurement errors.

A representation of  $\Lambda_{\text{meas}}$  can be estimated by performing full process tomography on the 2-qubit

identity channel. That tomography estimates a representation of  $\Lambda_{\text{prep}} \otimes \Lambda_{\text{Id}} \otimes \Lambda_{\text{meas}}$ .  $\Lambda_{\text{prep}}$  is again assumed to be negligible, and the circuit itself is arbitrary short, so that we can assume that  $\Lambda_{\text{Id}} = \chi^I$  as well. Then, process tomography on the 2-qubit identity channel directly estimates a representation  $\chi^{\text{meas}}$  of  $\Lambda_{\text{meas}}$ . The results of this tomography experiment can be found in figure B.4.

Let  $S^{\text{total}}$ ,  $S$  and  $S^{\text{meas}}$  be the superoperators corresponding to the total experiment ( $\chi^{\text{total}}$ ), the erroneous circuit itself ( $\chi$ ) and the error map of the measurement ( $\chi^{\text{meas}}$ ), respectively. Then,  $S^{\text{total}} = S^{\text{meas}}S$  and thus

$$S = (S^{\text{meas}})^{-1}S^{\text{total}}, \quad (8.1)$$

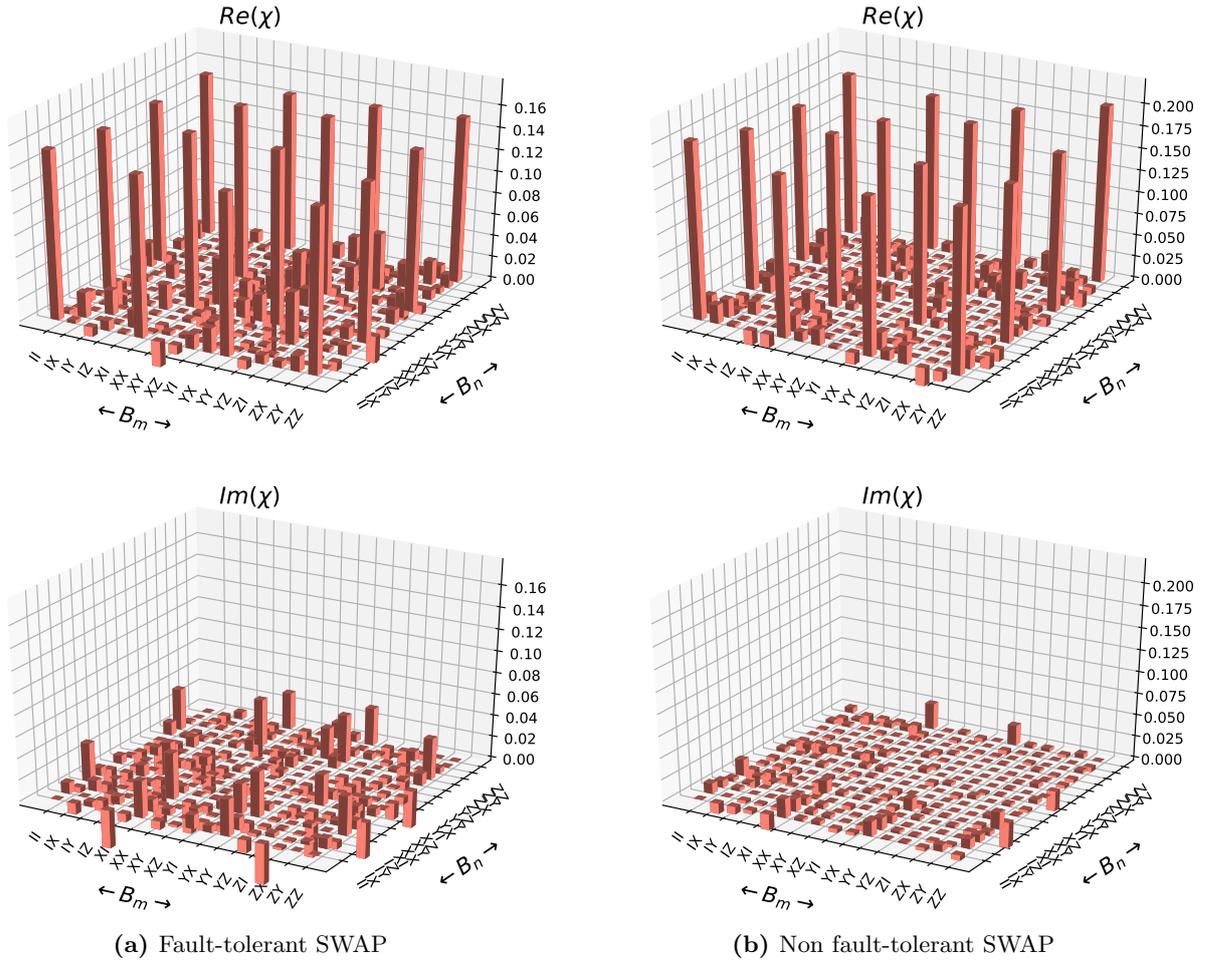
if  $S^{\text{meas}}$  is invertible. This gives a method of computing the desired  $\chi$  from  $\chi^{\text{meas}}$  and  $\chi^{\text{total}}$  via the corresponding superoperator-representations.

## 8.2 Results after filtering

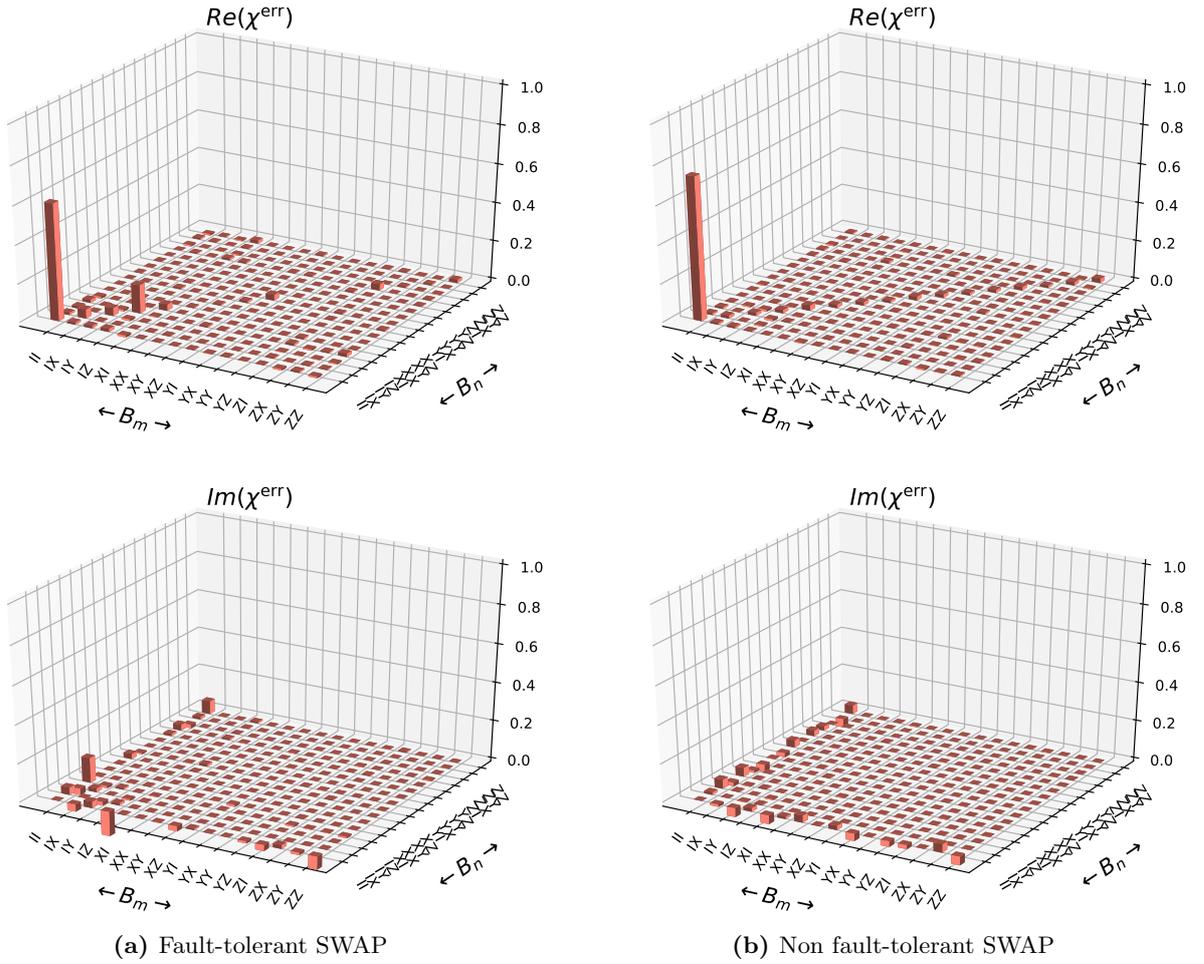
The process matrices  $\chi$  computed from the results of the tomography experiments for both implementations of the SWAP circuit can be found in figure 8.2; note the use of different  $y$ -scales due to different fidelities. Here,  $N_{\text{tot}} = 40960$ . Compare with the results of the simulation shown in figure 7.6 to separate the desired result and the noise. The process fidelities of the estimated  $\chi$  are  $F_p^{\text{FT}} = 0.608$  for the fault-tolerant SWAP and  $F_p^{\text{NFT}} = 0.742$  for the normal SWAP. The larger amount of noise on the fault-tolerant implementation reflects these fidelities; there is more error on that implementation. These graphs concern the filtered process matrix, where the SPAM errors have been filtered out. Furthermore, the CP subroutine was activated to obtain a process matrix with only positive eigenvalues. The initial estimates of  $\chi$  and  $\chi_{\text{err}}$  before the CP subroutine can be found in appendix B.3. Especially the fidelity of the non-fault-tolerant implementation is brought down.

In figure 8.3 the error matrices of the processes are shown; the hermiticity of  $\chi^{\text{err}}$  (and therefore  $\chi$ ) for both experiments is evident from this depiction. Moreover, the imaginary part of the elements in the first row and column are relatively large; this indicates a systematic unitary error (see section 3.5.1 and 4.4 or [11]). Furthermore, most other non-zero elements lie on the diagonal of the error matrix and are positive (and also necessarily real due to the hermiticity of  $\chi$ ); this invites the comparison with a Pauli error channel. By taking the Pauli twirl, off-diagonal elements are neglected. This corresponds to neglecting the systematic unitary error and focusing on the aspect of the error map  $\chi^{\text{err}}$  that randomizes the input states, creating density matrices from pure states. The error matrices after filtering before the CP subroutine can be found in appendix B.3.

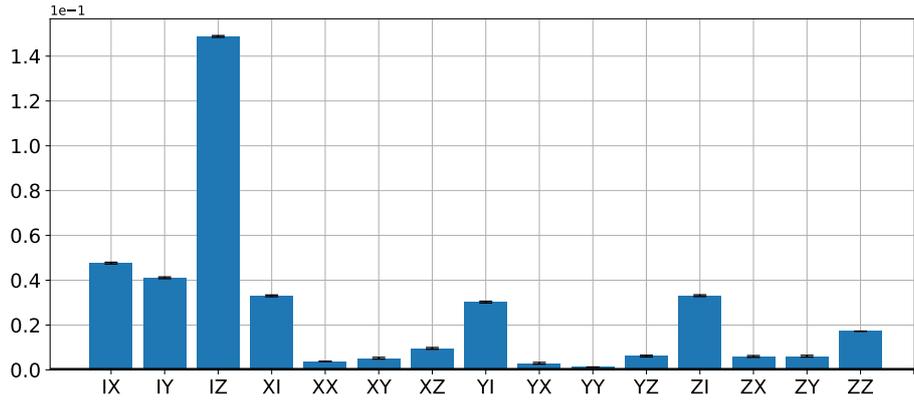
Figure 8.4 shows bar plots of the Pauli twirl of both error matrices; note the use of different  $y$ -scales here as well. As is expected from the fidelities of the processes, the error on the fault-tolerant SWAP is considerably larger. However, for the fault-tolerant circuit the single-weight errors are also much larger when compared to the multi-weight errors, whereas for the normal circuit there is no clear advantage for either of the two. This results in a ratio  $r = \frac{s_{2+}}{s_1}$ , with  $s_{2+}$  and  $s_1$  the measure for multi- and single-weight errors respectively, of  $r^{\text{FT}} = 0.170 \pm 0.0029$  for the fault-tolerant SWAP, and a ratio of  $r^{\text{NFT}} = 1.256 \pm 0.0129$  for the normal SWAP. However, the ratio for both implementations or increased by the CP subroutine after the SPAM filtering. The twirls of the error matrices after filtering before the CP subroutine can be found in appendix B.3.1.



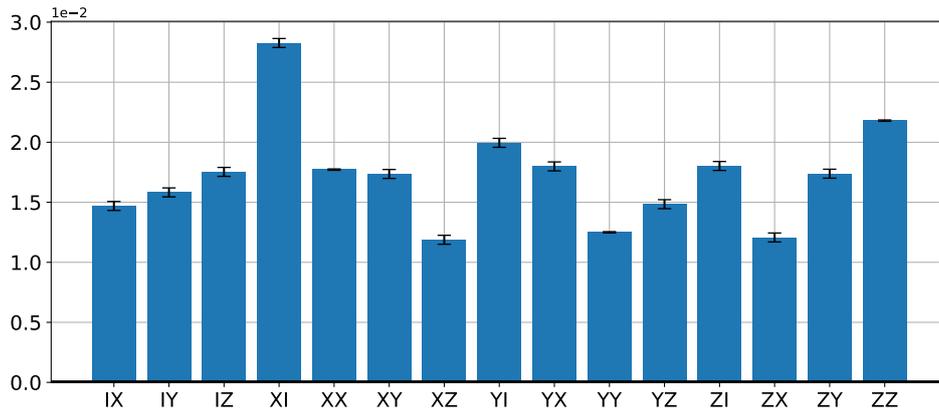
**Figure 8.2:** Results of tomography experiments of the fault-tolerant SWAP circuit (left) and the normal SWAP circuit (right);  $N_{\text{tot}} = 40960$ . Note the use of different  $y$ -scales due to different fidelities. Compare with the simulations shown in figure 7.6. The  $\chi$  matrix, with the fidelity of the process compared to the perfect SWAP operation is  $F_p^{\text{FT}} = 0.608$  for the fault-tolerant and  $F_p^{\text{NFT}} = 0.742$  for the normal implementation. It is clear that there is more noise on the fault-tolerant implementation, corresponding to the lower fidelity. For this number of shots, the initial estimate of  $\chi$  has only positive eigenvalues and is thus already CP. This drops the use of the CP subroutine in these experiments.



**Figure 8.3:** Error matrices corresponding to  $\chi$  computed from the tomographic results. The fault-tolerant implementation (a) has considerable more error than the normal implementation (b), as can be seen by the lower size of the leftmost bar in the fault-tolerant implementation. The imaginary part of the elements in the first row and column are relatively large; this indicates a systematic unitary error (see section 3.5.1 and 4.4 or [11]). Furthermore, most other non-zero elements lie on the diagonal of the error matrix and are positive (and necessarily real due to hermiticity of  $\chi$ ).



(a) Fault-tolerant SWAP



(b) Non fault-tolerant SWAP

**Figure 8.4:** Pauli twirl of  $\chi_{\text{err}}$  of the (a) fault-tolerant SWAP and (b) normal implementation. Note the use of different scales for the two graphs. The twirl of  $\chi_{\text{err}}$ , which is its diagonal, is used to calculate the ratio of multi- and single-weight Pauli errors. The fault-tolerant SWAP has considerable higher errors compared to the normal implementation, indicated by the lower fidelity of the process. However, the multi-weight Pauli errors are less likely than the single-weight errors.



# Chapter 9

## Discussion and conclusion

The previous chapter has shown the results of the tomography experiments. It allows us finally to analyse the fault-tolerant properties of the SWAP implementation. In section 9.1 I discuss the methods that are described in this report, and are used to gather the results. These results are discussed in section 9.2, and they are concluded in section 9.3. Finally, possible improvements and proposals for further research are given in section 9.4.

### 9.1 Discussion of the methods

#### The tomography experiment simulations

The simulations shown in chapter 7 reveal a few properties of the tomography procedure. Concerning the tomography experiments, a very important variable is the total number of shots  $N_{\text{tot}}$ . As can be seen in figure 7.5, the tomographic reconstruction methods can introduce considerable infidelity on the estimated process: for a  $N_{\text{tot}}$  as high as 4096 the estimated  $\chi$  will have a drop in fidelity of over 10%, even if the device on which the experiments are performed is completely devoid of errors.

#### The CP subroutine

A significant part of this decrease in estimated fidelity is due to the CP subroutine. The method of mapping back to the space of CP channels is very straightforward, but can also negatively influence the fidelity. From section 4.4.1 we know that the largest eigenvalue  $\lambda_{\text{max}}$  of  $\chi$  is approximately equal to the process fidelity  $F_p$ . If we consider the method described in section 4.3.2, we have for the new maximum eigenvalue  $\lambda_{\text{max}}^{\text{est}}$ , compared to the old maximum eigenvalue  $\lambda_{\text{max}}$ :

$$\frac{\lambda_{\text{max}}^{\text{est}}}{\lambda_{\text{max}}} = \frac{1}{\lambda_{\text{max}}} \frac{\lambda_{\text{max}} + |\lambda_{\text{min}}|}{1 + |\lambda_{\text{min}}|d^2} = \frac{1 + \frac{|\lambda_{\text{min}}|}{\lambda_{\text{max}}}}{1 + |\lambda_{\text{min}}|d^2} < 1, \quad (9.1)$$

since  $\frac{1}{\lambda_{\text{max}}} < d^2$ , and the strict inequality coming from the fact that  $|\lambda_{\text{min}}| > 0$ .

There are 2 considerations to this result:

- The ratio  $\frac{\lambda_{\text{max}}^{\text{est}}}{\lambda_{\text{max}}}$  scales as  $\frac{1}{1+2^{2n}}$ . For a large number of qubits, this decays to zero exponentially quickly.
- $\lim_{|\lambda_{\text{min}}| \rightarrow 0} \frac{\lambda_{\text{max}}^{\text{est}}}{\lambda_{\text{max}}} = 1$  and  $|\lambda_{\text{min}}|$  gets smaller for as  $N_{\text{tot}}$  increases, so increasing the number of experiments decreases the severity of the problem. This is the obvious result that more repetitions of the experiments decrease statistical noise.

It is now also evident from Eq. (9.1) that the method introduced in section 4.3.2 to obtain a CP representation impacts the fidelity of the estimated process. The fidelity of the new process  $F_p$  is: (note that the eigenvectors are the same for  $\chi^{\text{est}}$  and  $\chi$ ):

$$F_p^{CP} = \sum_i \chi_i^{\text{est}} |\langle \psi_i, u \rangle|^2 = \sum_i \frac{\lambda_i + |\lambda_{\text{min}}|}{1 + |\lambda_{\text{min}}|d^2} |\langle \psi_i, u \rangle|^2 \approx \frac{\lambda_{\text{max}} + |\lambda_{\text{min}}|}{1 + |\lambda_{\text{min}}|d^2}. \quad (9.2)$$

This means that the CP subroutine will decrease the fidelity, and that the estimated fidelity will not perfectly reflect the experiment. The same considerations for  $\chi_{\max}^{\text{est}}$  made in the above list apply to  $F_p$  after application of the CP subroutine. To circumvent this problem, a more elaborate method of mapping to the subspace of CP channels needs to be devised. Extra care has to be taken to also take TP channels into account; the method must not the TP constraint.

It is worth noting that there is an internal function of qiskit that performs a mapping to the space of CP maps based on a technique discussed in [10]. We have used this function during our testing as well; this function does indeed map a process matrix (or Choi matrix) to the subspace of CP maps, but by doing so the matrix is also mapped outside the space of TP maps, which was validated by various tests. The authors of the paper make the assumption that if the method works for state tomography, it should also work for process tomography. However, the constraint for states is solely that the trace of the matrix must equal 1, whereas the TP constraint for processes is stricter. Basically, the alteration of the Choi-Jamiolkowski isomorphism (see section 3.2.2) is not taken into account<sup>1</sup>.

## The error model

Concerning the simulations of the fault-tolerant and non-fault-tolerant circuits with faulty gates, they show qualitatively what can be expected from the experiments. Figure 7.7 shows that the fault-tolerant circuit has a much lower fidelity than the normal implementation for the same error rates, and that the fidelity is also considerably more dependent on the error rates. These two things are to be expected, because the fault-tolerant circuit has a much larger gate depth, and there are more than four times as many faulty locations in the circuit. Moreover, the contours in figure 7.7 are straight for small  $p_s$  and  $p_m$ , and only for larger error rates they start to bend. This is a direct consequence of the errors dependent on higher orders of the error rates, which are thus suppressed for small rates. Only once the rates grow in value, the higher-order terms start to be significant.

The simulations also show that the fault-tolerant circuit does what it is designed to do when errors occur uniformly and independently following each operation. The ratio  $r$  in the fault-tolerant circuit is much lower for a fixed  $p_s$  and  $p_m$ , and it goes to 0 as the error rates grow smaller. Notably, even for perfect 2-qubit gates ( $p_m = 0$ ), the ratio  $r > 0$ , meaning there are weight-2 errors for finite  $p_s$ . This is a direct consequence of error propagation, but it does not affect the fault-tolerant properties of the circuit: the definition of fault tolerance in chapter 6 does not discriminate between single and multi-qubit gates: the multi-qubit error suppression must hold for error rates of any kind. The contours of the fault-tolerant error ratios are very curved for small  $p_s$  and  $p_m$ , and they become increasingly straight for higher error rates. This is not to be expected, because for high error rates the dependence is linear, whereas for low error rates the dependence is non-linear.

The modelling of the faulty gates as depolarizing channels is a very straightforward but a crude oversimplification of the actual error maps. The simulations treat correlated (i.e. weight-2) errors on the 2-qubit gate as equally likely as single-weight errors on those gates. This results in the fact that the ratio  $r$  for the non-fault tolerant SWAP is never less than 1.5; correlated errors are always as likely as uncorrelated errors. To simulate the circuits more quantitatively would either require knowledge of the Kraus maps of every element of the circuit, or a much more elaborate simulation of the involved physics, both of which are beyond the scope of this text. However, the simulations offer a qualitative result, and can act as a ball-park estimate for the actual experiments, allowing us to check the experimental results.

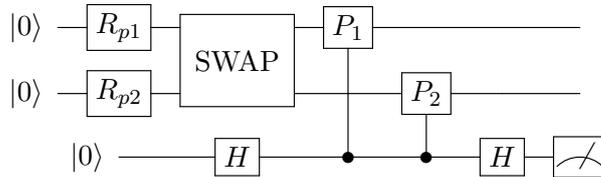
## SPAM errors and the tomography method

Tomography relies on a high number of measurements; there are  $3^n$  measurements that need to be performed per input state. The ibmqx4 device has considerable measurement error rates, this leads to infidelity and many uncorrelated errors. For tomography, the goal is to determine the expectation values of various  $n$ -qubit Paulis. In SQPT, these measurements are performed by measuring every qubit separately, resulting in  $n$  measurements. However, the expectation value can also be determined by entangling the data qubits with an ancilla using a phase kickback technique, and then only

---

<sup>1</sup>The authors of the paper and the qiskit team have been notified of this problem.

(repeatedly) measuring the ancilla; see figure 9.1. Note that this is essentially a repeated stabilizer measurement. If the single- and multi-qubit gate error rates are low compared to the measurement error rate, then the lower number of measurements can result in a better estimate of  $\chi$  and a higher fidelity. This is a technique that has not been used but is something to consider in future research.



**Figure 9.1:** Alternative method of tomography measurement of Pauli element  $P = P_1 \otimes P_2$ . Instead of measuring the qubits individually in the bases dictated by  $P_1$  and  $P_2$  respectively, the phase kickback technique is used to measure the the eigenvalue of the operator  $P$  immediately. This introduces the need for two extra multi-qubit gates and two extra single-qubit gates, but reduces the number of measurements that need to be performed.

## 9.2 Discussion of the results

There is a big gap between the fidelities of the fault-tolerant and non-fault-tolerant circuit. For the fault-tolerant SWAP circuit,  $F_p = 0.608$ , and for the normal circuit  $F_p = 0.742$ . This result is significant: there is a 30% difference between the two circuits. The unfiltered process fidelities were  $F_p^{\text{SPAM}} = 0.504$  and  $F_p^{\text{SPAM}} = 0.730$  for the fault-tolerant and normal circuit, respectively; the difference is here still more than 20%. However, note that the process fidelity of the tomography experiment on the identity channel  $F_p^{\text{meas}} = 0.818$ , and that for the fault-tolerant experiment we have

$$\frac{F_p^{\text{SPAM}}}{F_p^{\text{meas}}} = \frac{0.504}{0.818} = 0.616 \approx F_p^{\text{SPAM}}, \quad (9.3)$$

and conversely for the normal experiment we have

$$\frac{F_p^{\text{SPAM}}}{F_p^{\text{meas}}} = \frac{0.730}{0.818} = 0.819, \quad (9.4)$$

so the difference is scaled by a factor  $(F_p^{\text{meas}})^{-1}$  by filtering the SPAM errors. The latter fraction does not approximate as good as the first because of the CP subroutine; the process fidelity before the CP subroutine after filtering for the non-fault-tolerant circuit is  $F_p = 0.929$ , so the CP subroutine has a significant effect on this fidelity.

The Pauli twirls shown in figure 8.4 show that in the fault-tolerant implementation, the second qubit has a large  $Z$  error, whereas in the normal implementation, there is a large  $X$  error. Note, however, that these are not the same physical qubits. Furthermore, the correlated errors in the fault-tolerant twirl are much smaller than the single-weight errors, which is not as evident in the normal implementation. However, the total error for the fault-tolerant circuit is much larger - which reflects the process fidelities.

The unfiltered Pauli twirls, which can be found in appendix B.2.1, show less distinct behaviour. However, the twirl of the non-fault-tolerant experiment in figure B.5b and the twirl of the error matrix characterizing the SPAM errors in figure B.6 show a large correlation. This indicates that the uncorrelated errors in the non-fault-tolerant experiment are largely due to the measurement errors, which (on the assumption that they are independent for different qubits) are solely single-weight errors. This is also substantiated by the fact that the process fidelities for these two processes are close, and that after filtering the twirl of the non-fault-tolerant error matrix shows a relatively uniform distribution over the single- and multi-qubit errors.

As can be seen in the estimated error matrices, all tomography experiments shown signs of a systematic unitary error. This can be an indication of poorly calibrated gates on the ibmqx4 device. To

correct for these errors, the inverse of the systematic unitary error can be amended to the specification of the circuit on which the tomography is to be performed. This is, however, only a pseudo-solution. The systematic unitary error on the total map is a direct consequence of systematic unitaries on the individual gates that comprise the circuits, so correcting only for the total resulting error does not offer any method to correct any other circuit. By characterizing the systematic unitary errors on every individual gate in the gate set and applying an inverse to these errors (essentially recalibrating the gates), the systematic error in any circuit will be corrected.

### 9.3 Conclusion

The main result of the experiments is that the ratio of multi- to single-qubit errors is much lower for the fault-tolerant SWAP circuit compared to the normal implementation. For the normal implementation, weight-2 Pauli errors are almost as likely as single-weight Pauli errors; the ratio is  $r^{\text{NFT}} = 1.256$  with a standard deviation of  $\Delta r^{\text{NFT}} = 0.0129$ , but there are 1.5 as many weight-2 Paulis as there are weight-1 Paulis in  $P_2$ . A distance-3 code could not correct for those errors. For the fault-tolerant implementation, the ratio is  $r_{m/s}^{\text{FT}} = 0.170$  with a standard deviation of  $\Delta r^{\text{FT}} = 0.0029$ , which is about one tenth of the ratio of the normal implementation. It can thus be concluded with high certainty that the fault-tolerant circuit suppresses 2-qubit errors by a considerable amount. However, the fidelity of the fault-tolerant circuit is much lower than that of the normal implementation: 0.608 versus 0.742. The ratio might be lower for the fault-tolerant circuit, but the total error rate is much higher due to the added complexity of the fault-tolerant circuit. Therefore, with the current error rates, the fault-tolerant implementation does not offer a clear advantage over the normal circuit, and the error rates of the devices on which the computations will be performed will have to decrease still.

### 9.4 Recommendations

Although there can be made conclusions on these result, it can be expanded with further research. First and foremost, improvements in the tomographic experiments can be made. The number of necessary measurements is very large, but that amount might be reduced by using other data gathering methods, for instance by using the circuit in 9.1. The tomographic reconstruction methods can also be improved. The method of mapping to the CP subspace used in this thesis is suboptimal, but it does preserve trace-preservation. There are various methods of mapping to the CP subspace for state tomography, but these often break TP. An example of a method that does not break the TP constraint is [33]. Furthermore, a better method of SPAM filtering can give better results. The methods of SPAM filtering used in chapter 8 are relatively straightforward but also imperfect.

The simulations would also greatly benefit from a more detailed error model of the circuits. The simulations shown in section 7 can qualitatively explain the differences between the fault-tolerant and normal SWAP, but more details of the error processes need to be incorporated to get a quantitative simulation. Also, the measurement results indicate the existence of a systematic error. By performing tomography on every individual gate in the gate set these errors might be correctable.

Lastly, the fault-tolerant SWAP is only part of larger whole. Implementing different fault-tolerant circuits on the IBM devices to analyse them can further determine fault-tolerant properties (Implementing a full code is generally hard to do on the IBM devices due to connectivity restrictions.)

# Bibliography

- [1] Quantum Algorithm Zoo. <http://quantumalgorithmzoo.org/>. Accessed: 2018-01-03.
- [2] Dorit Aharonov and Michael Ben-Or. Fault-Tolerant Quantum Computation With Constant Error Rate. *STOC*, pages 1–18, jun 1999.
- [3] IBM Q Team. 'Tenerife': IBM Q5 Tenerife backend specification V1.3.0. "https://ibm.biz/qiskit-tenerife", 2018. [Online].
- [4] Daniel Gottesman. The Heisenberg Representation of Quantum Computers. *Physical Review A - Atomic, Molecular, and Optical Physics*, 1:1–20, jul 1998.
- [5] Christopher M. Dawson and Michael A. Nielsen. The Solovay-Kitaev algorithm. *Archiv*, 0(0):1–15, 2005.
- [6] M. Nielsen and I. L. Chuang. *Quantum Computation and Quantum Information: 10th Anniversary Edition*. 2010.
- [7] Man-Duen Choi. Positive Linear Maps on Complex. *October*, 290:285–290, 1975.
- [8] Min Jiang, Shunlong Luo, and Shuangshuang Fu. Channel-state duality. *Physical Review A - Atomic, Molecular, and Optical Physics*, 87(2):1–8, 2013.
- [9] John R. Taylor. *An introduction to error analysis*. 1997.
- [10] John A. Smolin, Jay M. Gambetta, and Graeme Smith. Maximum Likelihood, Minimum Effort. *Physical Review Letters*, (3), 2011.
- [11] Alexander N. Korotkov. Error matrices in quantum process tomography. *Physical Review Letters*, 92521, sep 2013.
- [12] Debbie W. Leung. Towards Robust Quantum Computation. *Archiv*, (July), 2000.
- [13] DW Leung. Choi's Proof and Quantum Process Tomography. *Arxiv preprint quant-ph/0201119*, pages 1–7, 2002.
- [14] J. B. Altepeter, D. Branning, E. Jeffrey, T. C. Wei, P. G. Kwiat, R. T. Thew, J. L. O'Brien, M. A. Nielsen, and A. G. White. Ancilla-Assisted Quantum Process Tomography. *Physical Review Letters*, 90(19):4, 2003.
- [15] M. Mohseni, A. T. Rezakhani, and D. A. Lidar. Quantum-process tomography: Resource analysis of different strategies. *Physical Review A - Atomic, Molecular, and Optical Physics*, 77(3):1–16, 2008.
- [16] Daniel Gottesman. *Stabilizer Codes and Quantum Error Correction*. PhD thesis, 1997.
- [17] Emanuel Knill and Raymond Laflamme. Theory of quantum error-correcting codes. *Physical Review A - Atomic, Molecular, and Optical Physics*, 55(2):900–911, 1997.
- [18] Daniel Gottesman. Surviving as a Quantum Computer in a Classical World. Unpublished.

- [19] Daniel Gottesman. An introduction to quantum error correction and fault-tolerant quantum computation. *American Mathematical Society*, 19(2):13–58, apr 2010.
- [20] Daniel A. Lidar and Todd A. Brun. *Quantum error correction*. 2012.
- [21] David P. DiVincenzo and IBM. The Physical Implementation of Quantum Computation. *IEEE International Conference on Fuzzy Systems*, 1:248–251, feb 2000.
- [22] John Preskill. Fault-tolerant quantum computation. In *Proceedings of 37th Conference on Foundations of Computer Science*, pages 56–65. IEEE Comput. Soc. Press, dec 1997.
- [23] John Preskill. Reliable quantum computers. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 454(1969):385–410, 1998.
- [24] Bryan Eastin and Emanuel Knill. Restrictions on transversal encoded quantum gate sets. *Physical Review Letters*, 102(11), 2009.
- [25] Daniel Gottesman and Isaac L. Chuang. Quantum Teleportation is a Universal Computational Primitive. *Nature*, 402(6760):390–393, aug 1999.
- [26] Sergey Bravyi and Alexei Kitaev. Universal quantum computation with ideal Clifford gates and noisy ancillas. *Physical Review A - Atomic, Molecular, and Optical Physics*, 2005.
- [27] David P. DiVincenzo and Peter W. Shor. Fault-tolerant error correction with efficient quantum codes. *Physical Review Letters*, 77(15):3260–3263, 1996.
- [28] Rui Chao and Ben W. Reichardt. Quantum Error Correction with only Two Extra Qubits. *Physical Review Letters*, 121(5):1–9, 2018.
- [29] Christopher Chamberland and Michael E. Beverland. Flag fault-tolerant error correction with arbitrary distance codes. *Quantum*, 2:53, feb 2018.
- [30] Rui Chao and Ben W. Reichardt. Fault-tolerant quantum computation with few qubits. *Quantum*, 15(1):34, may 2017.
- [31] IBM Q Experience. Qiskit: Quantum information science kit, 2017–. [Online; accessed 2018-01-03].
- [32] Travis Oliphant. NumPy: A guide to NumPy. USA: Trelgol Publishing, 2006–. [Online].
- [33] George C. Knee, Eliot Bolduc, Jonathan Leach, and Erik M. Gauger. Maximum-likelihood quantum process tomography via projected gradient descent. *Physical Review Letters*, pages 1–13, 2018.

# Appendix A

## Proofs for chapter 3

### A.1 Complete positivity in representations of $\Lambda$

This section aims to prove all the various equivalent conditions to complete positivity in all representations of  $\Lambda$  introduced in 3.2. I will compare a map  $\Lambda$ , its Choi matrix  $\rho_{\text{Choi}}$ , its Kraus decomposition  $\{A_k\}$  and its process matrix  $\chi$ . The following statements concerning complete positivity are equivalent:

1.  $\Lambda$  is completely positive
2.  $\rho_{\text{Choi}}$  is positive-semidefinite
3. There is a Kraus decomposition of  $\Lambda$  (i.e.  $L_k = R_k$  in 3.2.3)
4. The process matrix  $\chi$  is positive-semidefinite

The statements are shown to be equivalent using the transitive property; they are proven in the following order (with  $x \rightarrow y$  meaning statement  $x$  leads to statement  $y$ ):

- $1 \rightarrow 2$  ( $\Lambda$  being completely positive implies that the Choi matrix is positive-semidefinite)
- $2 \rightarrow 3$  (The Choi matrix being positive-semidefinite implies existence of a Kraus decomposition for  $\Lambda$ )
- $3 \rightarrow 1$  (The existence of a Kraus decomposition of  $\Lambda$  implies that  $\Lambda$  is completely positive)
- $4 \rightarrow 3$  (The process matrix  $\chi$  being positive-semidefinite implies that there exists a Kraus decomposition of  $\Lambda$ )
- $3 \rightarrow 4$  (The existence of a Kraus decomposition of  $\Lambda$  implies that the process matrix  $\chi$  is positive-semidefinite)

#### A.1.1 $\Lambda$ being completely positive implies that the Choi matrix is positive-semidefinite ( $1 \rightarrow 2$ )

By definition,  $\Lambda \otimes I$  preserves positivity if it is a CP map. Since  $|\Omega\rangle\langle\Omega|$  is rank 1 and has as its only non-zero eigenvalue  $\lambda = 1$ , positivity of  $\rho_{\text{Choi}}$  follows because the Choi matrix is  $(\Lambda \otimes I) |\Omega\rangle\langle\Omega|$ .

#### A.1.2 The Choi matrix being positive-semidefinite implies existence of a Kraus decomposition for $\Lambda$ ( $2 \rightarrow 3$ )

First, decompose  $\rho_{\text{Choi}}$  into its eigenvalues  $\lambda_k$  and eigenvectors  $|J_k\rangle$  to obtain the Schmidt decomposition:  $\rho_{\text{Choi}} = \sum_k \lambda_k |J_k\rangle\langle J_k|$ . Since  $\rho_{\text{Choi}}$  is positive-semidefinite, we have  $\lambda_k \geq 0 \forall k$  and  $\text{Re}\{\lambda_k\} = \lambda_k$ .

Furthermore, we can decompose  $|J_k\rangle$  into a tensor product of vectors in the system space and vectors in the environment space:  $\sqrt{\lambda_k}|J_k\rangle = \sum_i |a_i\rangle^k \otimes |b_i\rangle^k$ . We then get:

$$\begin{aligned}
\rho_{\text{Choi}} &= \sum_k \lambda_k |J_k\rangle \langle J_k| \\
&= \sum_k \sqrt{\lambda_k} |J_k\rangle \langle J_k| \sqrt{\lambda_k} \\
&= \sum_k \sum_i |a_i\rangle^k \otimes |b_i\rangle^k \sum_j \langle a_j|^k \otimes \langle b_j|^k \\
&= \sum_{i,j,k} |a_i\rangle^k \langle a_j|^k \otimes |b_i\rangle^k \langle b_j|^k
\end{aligned} \tag{A.1}$$

Therefore (using the relation between  $\rho_{\text{Choi}}$  and  $\Lambda$  from Eq. (??), proven in section A.3.1):

$$\begin{aligned}
\Lambda(\rho) &= d \operatorname{tr}_2 [\rho_{\text{Choi}}(I \otimes \rho^T)] \\
&= d \operatorname{tr}_2 [(\sum_{i,j,k} |a_i\rangle^k \langle a_j|^k \otimes |b_i\rangle^k \langle b_j|^k)(I \otimes \rho^T)] \\
&= d \operatorname{tr}_2 [\sum_{i,j,k} |a_i\rangle^k \langle a_j|^k \otimes |b_i\rangle^k \langle b_j|^k \rho^T] \\
&= d \sum_{i,j,k} |a_i\rangle^k \langle a_j|^k \langle b_j|^k \rho^T |b_i\rangle^k \\
&= \sum_{i,j,k} \sqrt{d} |a_i\rangle^k \langle b_i^*|^k \rho |b_j^*\rangle^k \langle a_j|^k \sqrt{d} \\
&= \sum_k A_k \rho A_k^\dagger
\end{aligned} \tag{A.2}$$

With  $A_k = \sqrt{d} \sum_i |a_i\rangle^k \langle b_i^*|^k$ . So, if  $\rho_{\text{Choi}} \geq 0$ ,  $\Lambda$  can be written using the operator-sum decomposition with  $L_k = R_k$ .

### A.1.3 The existence of a Kraus decomposition of $\Lambda$ implies that $\Lambda$ is completely positive ( $3 \rightarrow 1$ )

$\Lambda$  is CP if  $(\Lambda \otimes I)\rho_{\text{ext}} \geq 0$  when  $\rho_{\text{ext}} \geq 0$ . In other words:

$$\langle v | \Lambda \otimes I(\rho_{\text{ext}}) | v \rangle \geq 0 \quad \forall |v\rangle \tag{A.3}$$

If  $\Lambda(\rho) = \sum_k A_k \rho A_k^\dagger$ , then (with  $|v'_k\rangle = (A_k^\dagger \otimes I) |v\rangle$ )

$$\begin{aligned}
\langle v | \Lambda \otimes I(\rho_{\text{ext}}) | v \rangle &= \langle v | \sum_k (A_k \otimes I) \rho_{\text{ext}} (A_k^\dagger \otimes I) | v \rangle \\
&= \sum_k \langle v'_k | \rho_{\text{ext}} | v'_k \rangle \geq 0 \quad \forall |v\rangle
\end{aligned} \tag{A.4}$$

Where the last inequality follows since  $\rho_{\text{ext}}$  is positive-semidefinite.

**A.1.4 The process matrix  $\chi$  being positive-semidefinite implies that there exists a Kraus decomposition of  $\Lambda$  ( $4 \rightarrow 3$ )**

Since any positive matrix can be diagonalized by a unitary matrix, we have  $\chi = UDU^\dagger$ , or  $\chi_{m,n} = \sum_i U_{m,i} d_m U_{n,i}^*$  with  $d_m \geq 0$  the eigenvalues of  $\chi$ , and  $\text{Re}\{d_m\} = d_m$ . Then:

$$\begin{aligned}
\Lambda(\rho) &= \sum_{m,n} \chi_{m,n} B_m \rho B_n^\dagger \\
&= \sum_{m,n} \sum_i U_{m,i} d_m U_{n,i}^* B_m \rho B_n^\dagger \\
&= \sum_i \sum_m \sqrt{d_m} U_{m,i} B_m \rho \sum_n \sqrt{d_n} U_{n,i}^* B_n^\dagger \\
&= \sum_i A_i \rho A_i^\dagger
\end{aligned} \tag{A.5}$$

With  $A_i = \sum_m \sqrt{d_m} U_{m,i} B_m$ .

**A.1.5 The existence of a Kraus decomposition of  $\Lambda$  implies that the process matrix  $\chi$  is positive-semidefinite ( $3 \rightarrow 4$ )**

From section A.3.3 we know that  $\chi_{m,n} = \sum_k \langle B_m, A_k \rangle \langle A_k, B_n \rangle$ . Therefore,  $\chi$  is a convex linear combination of the positive operators  $|A_i\rangle \langle A_i|$ , meaning that  $\chi = \sum_k \alpha_k |A_i\rangle \langle A_i|$  with  $\alpha_k \geq 0$ ,  $\forall k$ . This means that  $\chi$  is positive itself.

## A.2 Trace-preservation in representations of $\Lambda$

This section aims to prove all the various equivalent conditions to trace-preservation in all representations of  $\Lambda$  introduced in 3.2. I will compare a map  $\Lambda$ , its Choi matrix  $\rho_{\text{Choi}}$ , its Kraus decomposition  $\{A_k\}$  and its process matrix  $\chi$ . The following statements concerning trace-preservation are equivalent:

1.  $\Lambda$  is trace-preserving
2.  $\text{tr}_1 [\rho_{\text{Choi}}] = \frac{1}{d}I$
3.  $\sum_k A_k^\dagger A_k = I$
4.  $\sum_{m,n} \chi_{m,n} B_n^\dagger B_m = I$

The statements are shown to be equivalent using the transitive property; they are proven in the following order (with  $x \rightarrow y$  meaning statement  $x$  leads to statement  $y$ ):

- $1 \rightarrow 2$  ( $\Lambda$  being trace-preserving implies that  $\text{tr}_1 [\rho_{\text{Choi}}] = \frac{1}{d}I$ )
- $2 \rightarrow 1$  ( $\text{tr}_1 [\rho_{\text{Choi}}] = \frac{1}{d}I$  implies that  $\Lambda$  is trace-preserving)
- $1 \rightarrow 3$  ( $\Lambda$  being trace-preserving implies that  $\sum_k A_k^\dagger A_k = I$  for the Kraus operators  $\{A_k\}$ )
- $3 \rightarrow 1$  (If  $\sum_k A_k^\dagger A_k = I$  for the Kraus operators  $\{A_k\}$  then  $\Lambda$  is trace-preserving)
- $3 \rightarrow 4$  (If  $\sum_k A_k^\dagger A_k = I$  for the Kraus operators  $\{A_k\}$  then  $\sum_{m,n} \chi_{m,n} B_n^\dagger B_m = I$  for the process matrix  $\chi$ )
- $4 \rightarrow 1$  (If  $\sum_{m,n} \chi_{m,n} B_n^\dagger B_m = I$  for the process matrix  $\chi$  then  $\Lambda$  is trace-preserving)

### A.2.1 $\Lambda$ being trace-preserving implies that $\text{tr}_1 [\rho_{\text{Choi}}] = \frac{1}{d}I$ ( $1 \rightarrow 2$ )

First, expand  $\rho$  into the canonical basis:  $\rho = \sum_{i,j} \rho_{ij} |i\rangle \langle j|$ . Therefore,  $\text{tr} [\rho] = \sum_{i,j} \rho_{i,i} = \sum_{i,j} \rho_{ij} \delta_{i,j}$ . Furthermore, since  $\Lambda$  is linear,  $\text{tr} [\Lambda(\rho)] = \sum_{i,j} \rho_{ij} \text{tr} [\Lambda(|i\rangle \langle j|)]$ .

The premise is that  $\Lambda$  is trace-preserving:  $\text{tr} [\Lambda(\rho)] = \text{tr} [\rho]$ . In other words, we have the identity  $\sum_{i,j} \rho_{ij} \text{tr} [\Lambda(|i\rangle \langle j|)] = \sum_{i,j} \rho_{ij} \delta_{i,j}$ . This holds  $\forall \rho$ , so  $\text{tr} [\Lambda(|i\rangle \langle j|)] = \delta_{i,j}$ .

Now:

$$\begin{aligned}
\text{tr}_1 [\rho_{\text{Choi}}] &= \text{tr}_1 [(\Lambda \otimes I) |\Omega\rangle \langle \Omega|] \\
&= \frac{1}{d} \text{tr}_1 [(\Lambda \otimes I) \sum_{i,j} (|i\rangle \otimes |i\rangle) \langle j| \otimes \langle j|)] \\
&= \frac{1}{d} \text{tr}_1 \left[ \sum_{i,j} \Lambda(|i\rangle \langle j|) \otimes |i\rangle \langle j| \right] \\
&= \frac{1}{d} \sum_{i,j} \text{tr} [\Lambda(|i\rangle \langle j|)] |i\rangle \langle j| \\
&= \frac{1}{d} \sum_{i,j} \delta_{i,j} |i\rangle \langle j| \\
&= \frac{1}{d} \sum_i |i\rangle \langle i| = \frac{1}{d}I
\end{aligned} \tag{A.6}$$

**A.2.2**  $\text{tr}_1 [\rho_{\text{Choi}}] = \frac{1}{d}I$  implies that  $\Lambda$  is trace-preserving ( $2 \rightarrow 1$ )

The premise is that  $\text{tr}_1 [\rho_{\text{Choi}}] = \frac{1}{d}I = \frac{1}{d} \sum_i |i\rangle \langle i| = \frac{1}{d} \sum_{i,j} \delta_{ij} |i\rangle \langle j|$ .

We still have (with  $\{|i\rangle\}$  and  $\{|j\rangle\}$  an orthonormal basis for the operator space):

$$\begin{aligned} \text{tr}_1 [\rho_{\text{Choi}}] &= \text{tr}_1 [(\Lambda \otimes I) \sum_{i,j} \frac{1}{d} (|i\rangle \otimes |i\rangle)(\langle j| \otimes \langle j|)] \\ &= \text{tr}_1 \left[ \sum_{i,j} \frac{1}{d} \Lambda(|i\rangle \langle j|) \otimes |i\rangle \langle j| \right] \\ &= \sum_{i,j} \frac{1}{d} \text{tr} [\Lambda(|i\rangle \langle j|)] |i\rangle \langle j| \end{aligned} \tag{A.7}$$

Following from the premise,  $\text{tr} [\Lambda(|i\rangle \langle j|)] = \delta_{i,j}$ , and thus:

$$\begin{aligned} \text{tr} [\Lambda(\rho)] &= \text{tr} \left[ \Lambda \left( \sum_{i,j} \rho_{ij} |i\rangle \langle j| \right) \right] \\ &= \sum_{i,j} \rho_{ij} \text{tr} [\Lambda(|i\rangle \langle j|)] \\ &= \sum_{i,j} \rho_{ij} \delta_{ij} = \sum_i \rho_{ii} = \text{tr} [\rho] \end{aligned} \tag{A.8}$$

**A.2.3**  $\Lambda$  being trace-preserving implies that  $\sum_k A_k^\dagger A_k = I$  for the Kraus operators  $\{A_k\}$  ( $1 \rightarrow 3$ )

The premise is that  $\text{tr} [\Lambda(\rho)] = \text{tr}(\rho)$ . Furthermore,  $\text{tr} [\Lambda(\rho)] = \text{tr} [I^\dagger \Lambda(\rho)] = \langle I, \Lambda(\rho) \rangle$  and similarly  $\text{tr} [\rho] = \langle I, \rho \rangle$ .

I denote  $\Lambda^*$  as the dual of  $\Lambda$ :  $\langle I, \Lambda(\rho) \rangle = \langle \Lambda^*(I), \rho \rangle$ . (i.e.  $\Lambda^*$  is that operator for which that equality holds). The premise is that  $\langle \Lambda^*(I), \rho \rangle = \langle I, \rho \rangle \forall \rho$ , meaning that  $\Lambda^*$  is unital:  $\Lambda^*(I) = I$ .

Furthermore, if  $\Lambda(\rho) = \sum_k A_k \rho A_k^\dagger$ , then  $\Lambda^*(\rho) = \sum_k A_k^\dagger \rho A_k$ . This can be shown by the cyclic and linear properties of the trace:

$$\langle \sigma, \Lambda(\rho) \rangle = \sum_k \langle \sigma, A_k \rho A_k^\dagger \rangle = \sum_k \text{tr} [\sigma A_k \rho A_k^\dagger] = \sum_k \text{tr} [A_k^\dagger \sigma A_k \rho] = \sum_k \langle A_k^\dagger \sigma A_k, \rho \rangle = \langle \Lambda^*(\sigma), \rho \rangle \tag{A.9}$$

So finally  $\sum_k A_k^\dagger A_k = \sum_k A_k^\dagger I A_k = \Lambda^*(I) = I$ .

**A.2.4** If  $\sum_k A_k^\dagger A_k = I$  for the Kraus operators  $\{A_k\}$  then  $\Lambda$  is trace-preserving ( $3 \rightarrow 1$ )

This is straightforward: the premise is  $\Lambda(\rho) = \sum_k A_k \rho A_k^\dagger$  with  $\sum_k A_k^\dagger A_k = I$ .

Then:

$$\text{tr} [\Lambda(\rho)] = \text{tr} \left[ \sum_k A_k \rho A_k^\dagger \right] = \text{tr} \left[ \sum_k A_k^\dagger A_k \rho \right] = \text{tr} [I \rho] = \text{tr} [\rho] \tag{A.10}$$

**A.2.5** If  $\sum_k A_k^\dagger A_k = I$  for the Kraus operators  $\{A_k\}$  then  $\sum_{m,n} \chi_{m,n} B_n^\dagger B_m = I$  for the process matrix  $\chi$  (3  $\rightarrow$  4)

Again,  $\Lambda(\rho) = \sum_k A_k \rho A_k^\dagger$  with  $\sum_k A_k^\dagger A_k = I$ .

We know (see sect. A.3.3) that  $\chi_{m,n} = \sum_k \alpha_{mk} \alpha_{nk}^* = \sum_k \langle B_m, A_k \rangle \langle A_k, B_n \rangle$ . So:

$$\begin{aligned}
I &= \sum_k A_k^\dagger A_k \\
&= \sum_{m,n} \sum_k \alpha_{nk}^* B_n^\dagger \alpha_{mk} B_m \\
&= \sum_{m,n} \sum_k \alpha_{mk} \alpha_{nk}^* B_n^\dagger B_m \\
&= \sum_{m,n} \chi_{m,n} B_n^\dagger B_m
\end{aligned} \tag{A.11}$$

Furthermore:

$$\begin{aligned}
\text{tr} [\chi] &= \sum_m \chi_{m,m} \\
&= \sum_m \sum_k \langle B_m, A_k \rangle \langle A_k, B_m \rangle \\
&= \sum_k \sum_m \langle A_k, B_m \rangle \langle B_m, A_k \rangle \\
&= \sum_k \langle \langle A_k | (\sum_m |B_m\rangle) \rangle \langle \langle B_m | \rangle | A_k \rangle \rangle \\
&= \sum_k \langle \langle A_k | I | A_k \rangle \rangle \\
&= \sum_k \langle A_k, A_k \rangle \\
&= \text{tr} \left[ \sum_k A_k^\dagger A_k \right] = \text{tr} [I] = d
\end{aligned} \tag{A.12}$$

Where  $\sum_m |B_m\rangle \langle \langle B_m | = I$  follows since  $B_i$  is an orthogonal basis and the last identity follows from the premise.

**A.2.6** If  $\sum_{m,n} \chi_{m,n} B_n^\dagger B_m = I$  for the process matrix  $\chi$  then  $\Lambda$  is trace-preserving (4  $\rightarrow$  1)

If  $\sum_{m,n} \chi_{m,n} B_n^\dagger B_m = I$ , then:

$$\begin{aligned}
\text{tr} [\Lambda(\rho)] &= \text{tr} \left[ \sum_{m,n} \chi_{m,n} B_m \rho B_n^\dagger \right] \\
&= \text{tr} \left[ \sum_{m,n} \chi_{m,n} B_n^\dagger B_m \rho \right] \\
&= \text{tr} [I \rho] = \text{tr} [\rho]
\end{aligned} \tag{A.13}$$

### A.3 Relations between different representations of $\Lambda$

This section aims to prove the relations between the different representations of  $\Lambda$  as stated in section 3.3.

#### A.3.1 The Choi matrix and $\Lambda(\rho)$

$\Lambda(\rho) = d \operatorname{tr}_2 [\rho_{\text{Choi}}(I \otimes \rho^T)]$  follows from:

$$\begin{aligned}
& d \operatorname{tr}_2 [\rho_{\text{Choi}}(I \otimes \rho^T)] \\
&= d \operatorname{tr}_2 \left[ \frac{1}{d} \sum_{i,j} (\Lambda \otimes I)(|i\rangle \otimes |i\rangle)(\langle j| \otimes \langle j|)(I \otimes \rho^T) \right] \\
&= \operatorname{tr}_2 \left[ \sum_{i,j} (\Lambda(|i\rangle \langle j|) \otimes |i\rangle \langle j|)(I \otimes \rho^T) \right] \\
&= \operatorname{tr}_2 \left[ \sum_{i,j} \Lambda(|i\rangle \langle j|) \otimes |i\rangle \langle j| \rho^T \right] \\
&= \sum_{i,j} \Lambda(|i\rangle \langle j|) \operatorname{tr} [|i\rangle \langle j| \rho^T] \\
&= \sum_{i,j} \Lambda(|i\rangle \langle j|) \langle j| \rho^T |i\rangle = \sum_{i,j} \Lambda(|i\rangle \langle j|) \langle i| \rho |j\rangle \\
&= \sum_{i,j} \Lambda(\langle i| \rho |j\rangle |i\rangle \langle j|) = \sum_{i,j} \Lambda(\rho_{ij} |i\rangle \langle j|) \\
&= \Lambda(\rho)
\end{aligned} \tag{A.14}$$

#### A.3.2 The Choi matrix and $\chi$ matrix

Using the  $\chi$  decomposition  $\Lambda(\rho) = \sum_{m,n} \chi_{m,n} B_m \rho B_n^\dagger$ , with the basis  $\{|B_m\rangle\rangle$  as defined in section 3.3.2,  $\rho_{\text{Choi}}$  becomes:

$$\begin{aligned}
\rho_{\text{Choi}} &= (\Lambda \otimes I)(|\Omega\rangle \langle \Omega|) \\
&= \sum_{m,n} \chi_{m,n} (B_m \otimes I) |\Omega\rangle \langle \Omega| (B_n^\dagger \otimes I) \\
&= \sum_{m,n} \chi_{m,n} |B_m\rangle\rangle \langle\langle B_n|
\end{aligned} \tag{A.15}$$

The orthonormality of  $\{|B_i\rangle\rangle\}$  can be shown (this will be used in A.3.5):

$$\begin{aligned}
\langle B_n | B_m \rangle &= \langle \Omega | (B_n^\dagger \otimes I) (B_m \otimes I) | \Omega \rangle \\
&= \left( \frac{1}{\sqrt{d}} \sum_i \langle i | \otimes \langle i | \right) (B_n^\dagger \otimes I) (B_m \otimes I) \left( \frac{1}{\sqrt{d}} \sum_j |j\rangle \otimes |j\rangle \right) \\
&= \sum_{i,j} \frac{1}{d} \langle i | B_n^\dagger B_m | j \rangle \otimes \langle i | I I | j \rangle \\
&= \sum_{i,j} \frac{1}{d} \langle i | B_n^\dagger B_m | j \rangle \delta_{ij} \\
&= \sum_i \frac{1}{d} \langle i | B_n^\dagger B_m | i \rangle = \frac{1}{d} \operatorname{tr} [B_n^\dagger B_m] \\
&= \delta_{m,n}
\end{aligned} \tag{A.16}$$

With the last equality following from the fact that  $\{B_i\}$  is orthogonal under the standard Hilbert-Schmidt inner product  $\langle B_i, B_j \rangle = \text{tr} [B_i^\dagger B_j] = d\delta_{ij}$ .

### A.3.3 The $\chi$ matrix and the Kraus decomposition

$\Lambda(\rho) = \sum_k A_k \rho A_k^\dagger$ . Expanding  $A_k$  into an orthogonal basis  $B_m$  we get  $A_k = \sum_m \alpha_{mk} B_m$  with  $\alpha_{mk} = \langle B_m, A_k \rangle$ . Thus:

$$\begin{aligned}
\Lambda(\rho) &= \sum_k A_k \rho A_k^\dagger \\
&= \sum_k \left( \sum_m \alpha_{mk} B_m \right) \rho \left( \sum_n \alpha_{nk}^* B_n^\dagger \right) \\
&= \sum_{m,n} \sum_k \alpha_{mk} \alpha_{nk}^* B_m \rho B_n^\dagger \\
&= \sum_{m,n} \chi_{m,n} B_m \rho B_n^\dagger
\end{aligned} \tag{A.17}$$

With  $\chi_{m,n} = \sum_k \alpha_{mk} \alpha_{nk}^* = \sum_k \langle B_m, A_k \rangle \langle A_k, B_n \rangle$ . This is used in section 3.3.3 and section A.1.5.

### A.3.4 The Choi matrix and the Kraus decomposition

$\rho_{\text{Choi}} = \frac{1}{d} \sum_{i,j,k} A_k |i\rangle \langle j| A_k^\dagger \otimes |i\rangle \langle j|$  follows directly from:

$$\begin{aligned}
\rho_{\text{Choi}} &= (\Lambda \otimes I) |\Omega\rangle \langle \Omega| \\
&= \frac{1}{d} \sum_k (A_k \otimes I) |\Omega\rangle \langle \Omega| (A_k^\dagger \otimes I) \\
&= \frac{1}{d} \sum_{i,j,k} A_k |i\rangle \langle j| A_k^\dagger \otimes |i\rangle \langle j|
\end{aligned} \tag{A.18}$$

### A.3.5 The $\chi$ matrix and the superoperator

Using the identity  $|ABC\rangle\rangle = (C^{-1} \otimes A)|B\rangle\rangle$  and that for any unitary operator  $U$  it holds that  $U^\dagger = U^{-1}$  the relation of  $S$  expressed in terms of  $\chi$  is easily shown:

$$|\rho_{\text{out}}\rangle\rangle = \left| \sum_{m,n} \chi_{m,n} B_m \rho_{\text{in}} B_n^\dagger \right\rangle\rangle = \sum_{m,n} \chi_{m,n} |B_m \rho_{\text{in}} B_n^\dagger\rangle\rangle = \sum_{m,n} \chi_{m,n} (\overline{B_n} \otimes B_m) |\rho_{\text{in}}\rangle\rangle = S |\rho_{\text{in}}\rangle\rangle \tag{A.19}$$

The inverse relation is likewise trivial due to the orthogonality of  $\{|B_i\rangle\rangle\}$  (see sect. A.3.2):

$$\text{tr} [(\overline{B_n} \otimes B_m)^\dagger S] = \sum_{i,j} \chi_{i,j} \text{tr} [(\overline{B_n} \otimes B_m)^\dagger (\overline{B_j} \otimes B_i)] = \sum_{i,j} \chi_{i,j} \delta_{im} \delta_{jn} = \chi_{m,n} \tag{A.20}$$

## Appendix B

# Preliminary results on tomography experiments

B.1 Other simulation results

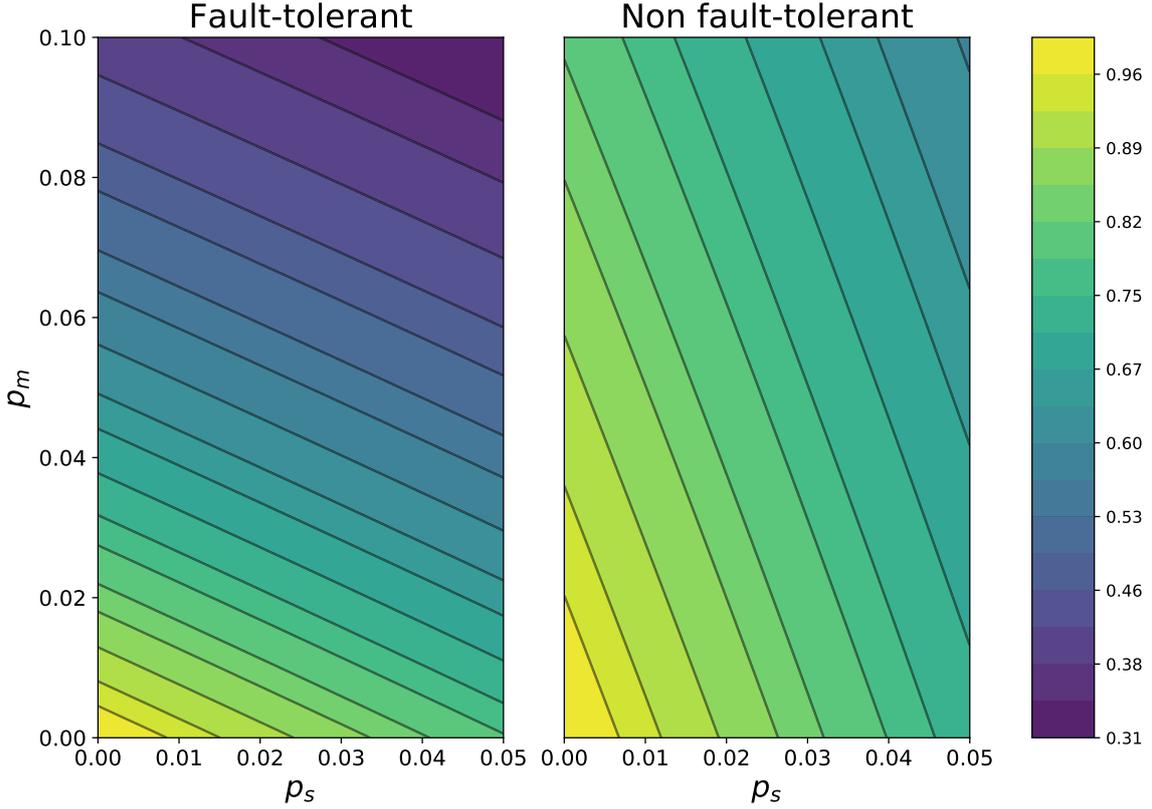
B.2 Results of tomography experiments without SPAM filtering

B.2.1 Pauli twirls of unfiltered error matrices

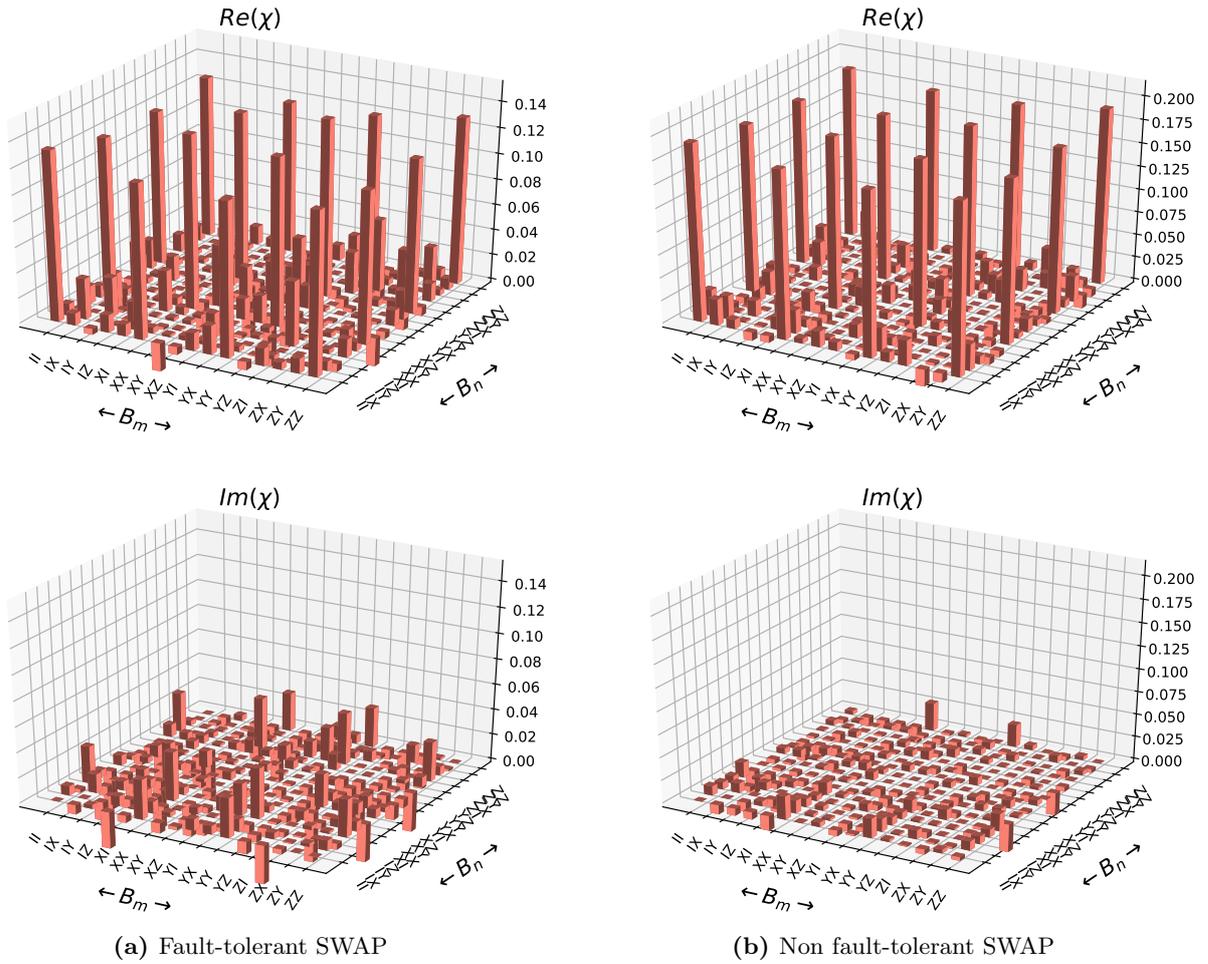
B.3 Results of experiments after filtering before CP subroutine

B.3.1 Pauli twirls of filtered error matrices before CP subroutine

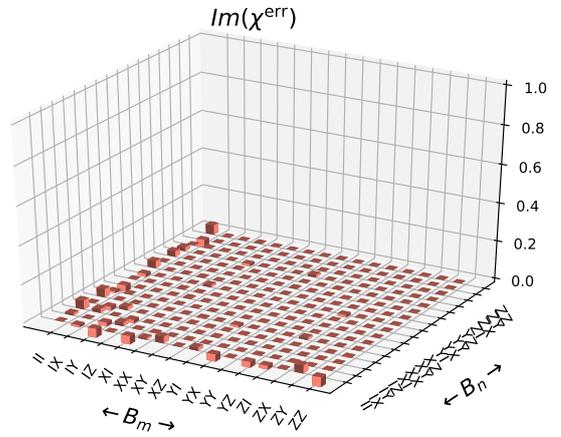
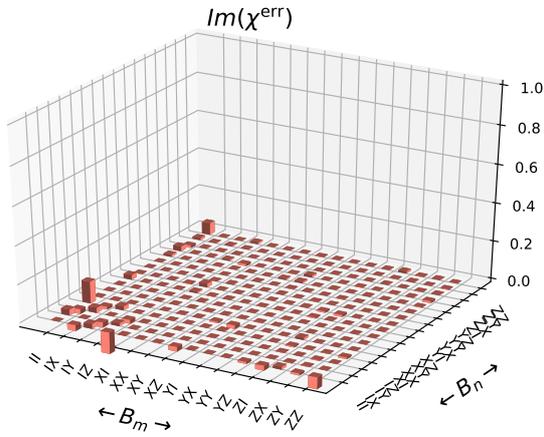
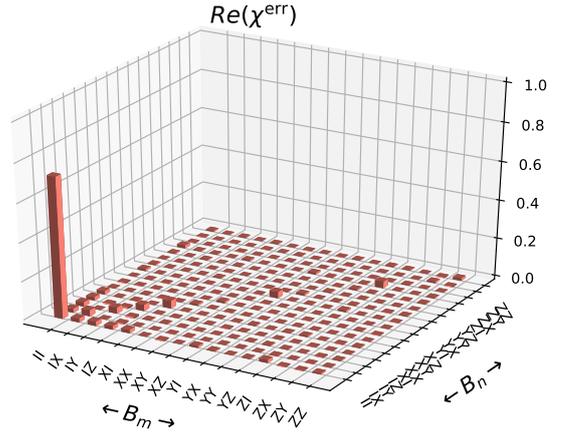
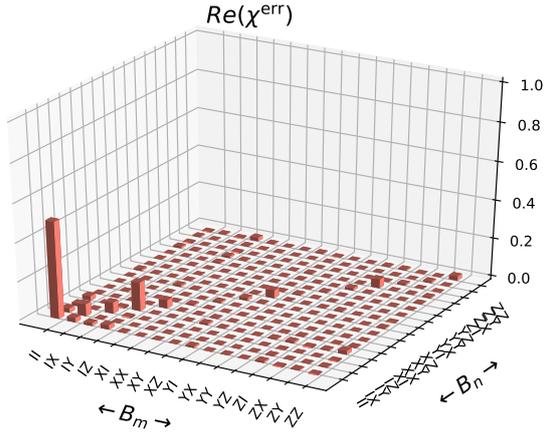
## Approximation of $F_p$



**Figure B.1:** Approximation of the simulations of the fidelities of the fault-tolerant and normal SWAP circuit, for the same error model as the simulations in section 7.3.2. The fidelity of the normal implementation is approximated as  $F_p \approx (1 - p_s)^4 \times (1 - p_m)^3$ , and the fidelity of the fault-tolerant circuit is approximated as  $F_p \approx (1 - \frac{9}{10}p_s)^{15} \times (1 - \frac{3}{8}p_s)^{12} \times (1 - \frac{12}{45}p_m)^9$ . The fidelity is approximated as the probability that the error map of every gate does not have a non-trivial action on the data qubits. For the non-fault tolerant circuit, this is  $(1 - p_s)$  for the 4 single-qubit gates and  $(1 - p_m)$  for the 3 multi-qubit gates. For the fault-tolerant circuit, this is  $(1 - \frac{9}{10}p_s)$  for the 15 weight locations,  $(1 - \frac{3}{8}p_s)$  for the 12 single-qubit gates and  $(1 - \frac{12}{45}p_m)$  for the 9 multi-qubit gates. This approximation only partly includes accounting for error cancellations. Compare with the actual simulations in figure 7.7, and note the correspondence. There is a difference between the actual simulations and the approximation, due to higher order terms, which is thus apparent for higher  $p_m$  and  $p_s$ .



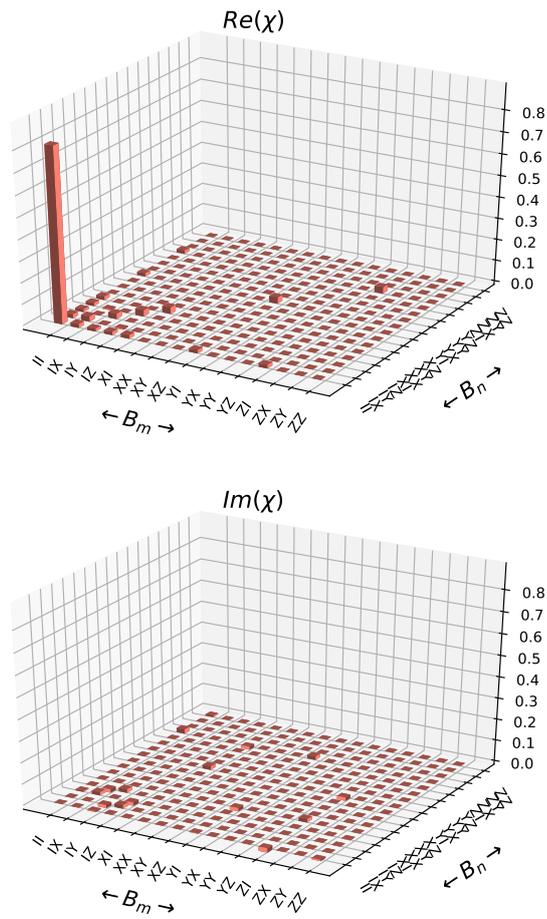
**Figure B.2:** Tomography results without SPAM filtering for the fault-tolerant (a) and normal (b) SWAP circuit. Here the process matrices  $\chi$  are shown.  $F_p^{\text{FT}} = 0.504$  for the fault-tolerant SWAP and  $F_p^{\text{NFT}} = 0.730$  for the normal SWAP.



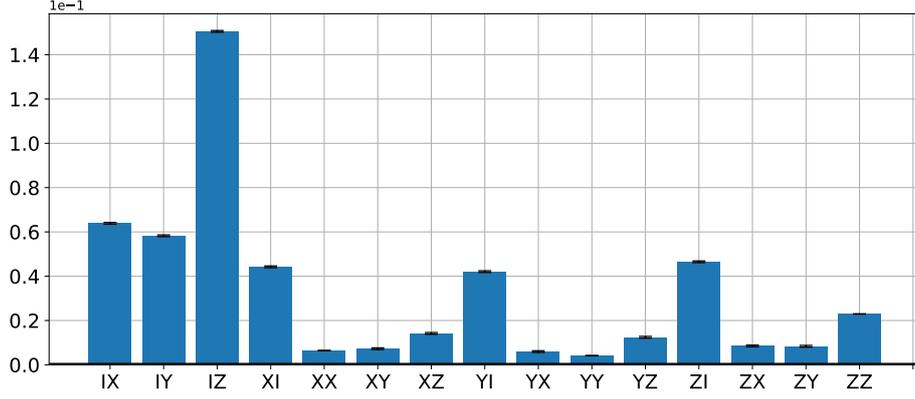
(a) Fault-tolerant SWAP

(b) Non fault-tolerant SWAP

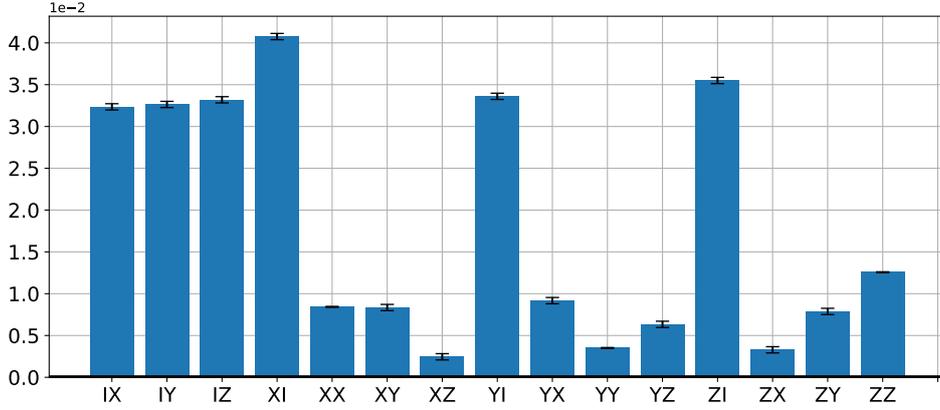
**Figure B.3:** Tomography results without SPAM filtering for the fault-tolerant (a) and normal (b) SWAP circuit. Here the error matrices  $\chi^{\text{err}}$  are shown.  $F_p^{\text{FT}} = 0.504$  for the fault-tolerant SWAP and  $F_p^{\text{NFT}} = 0.730$  for the normal SWAP.



**Figure B.4:** Tomography results for the identity channel to characterize the SPAM errors.  $F_p = 0.818$ . Because the desired operation is the identity channel, the error matrix  $\chi^{\text{err}}$  is identical to  $\chi$ .

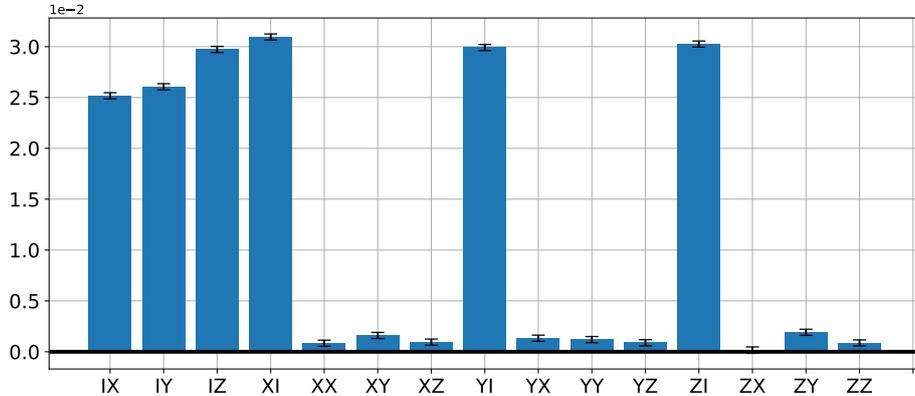


(a) Fault-tolerant SWAP

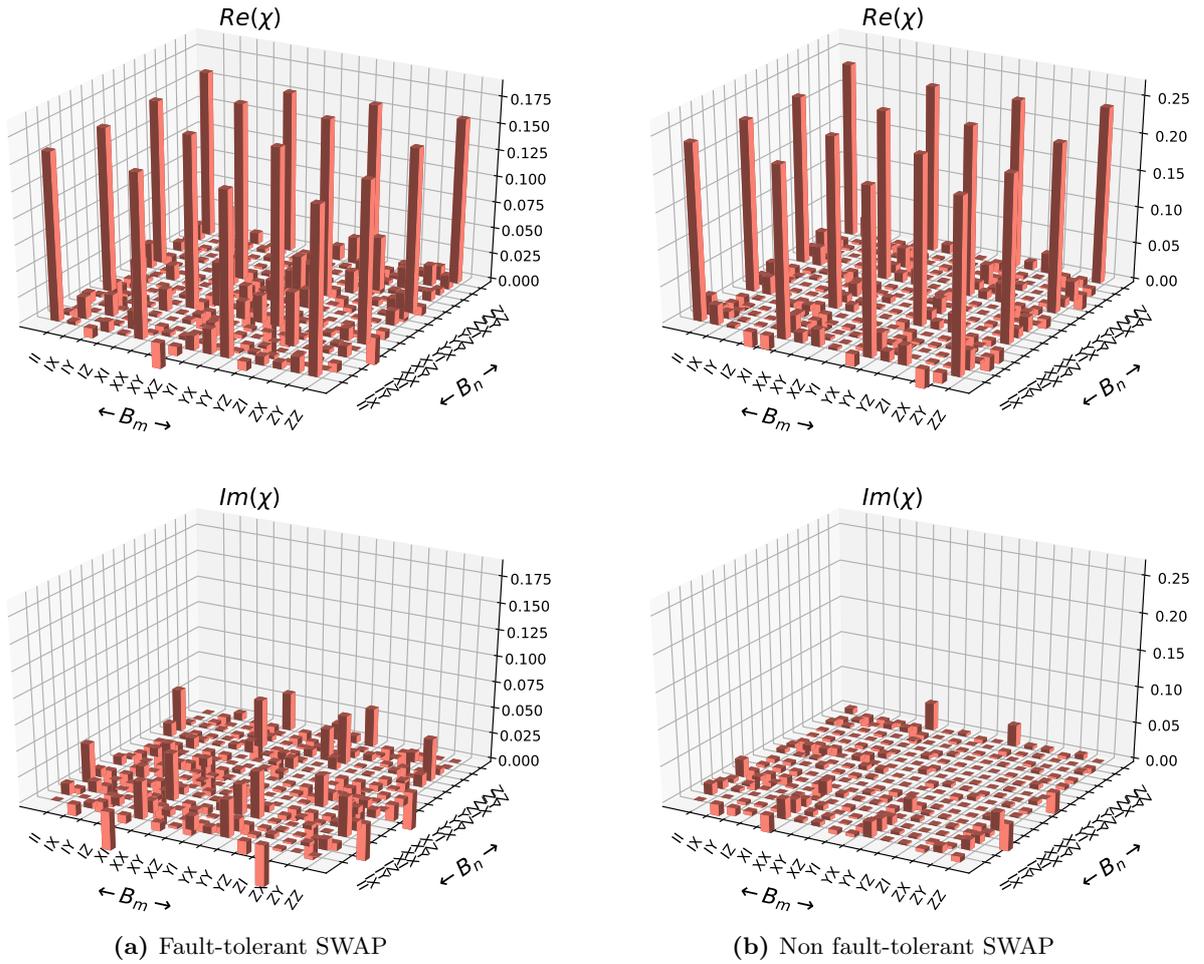


(b) Non fault-tolerant SWAP

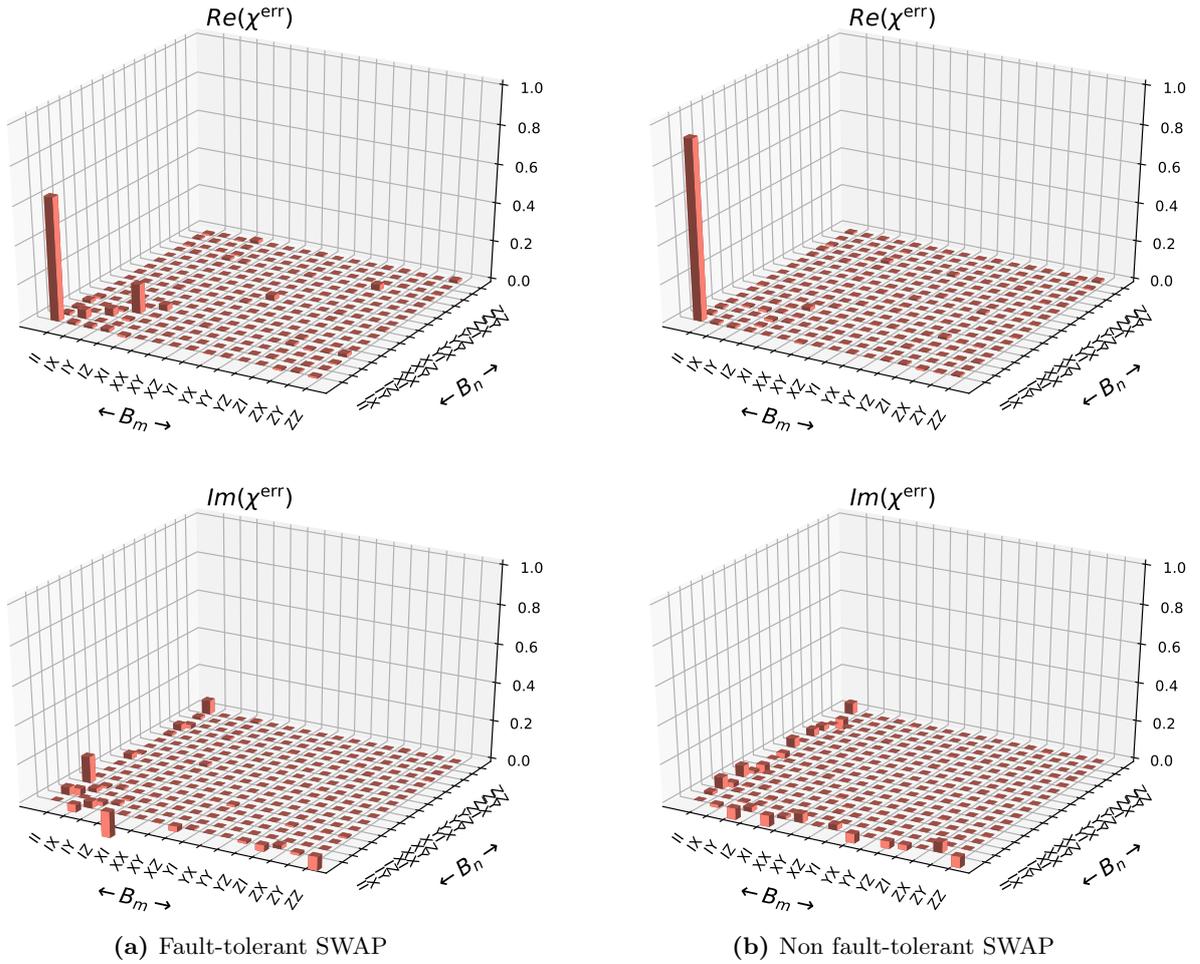
**Figure B.5:** Pauli twirl of the unfiltered error matrices  $\chi_{\text{err}}$  of the (a) fault-tolerant SWAP and (b) normal implementation. Note the use of different scales for the two graphs. The fault-tolerant SWAP has considerable higher errors compared to the normal implementation, indicated by the lower fidelity of the process. For both implementations, the uncorrelated errors (i.e. single-weight errors) are greater than the correlated errors. However, especially the normal implementation shows large correlation with the Pauli twirl of the SPAM error matrix shown in figure B.6.



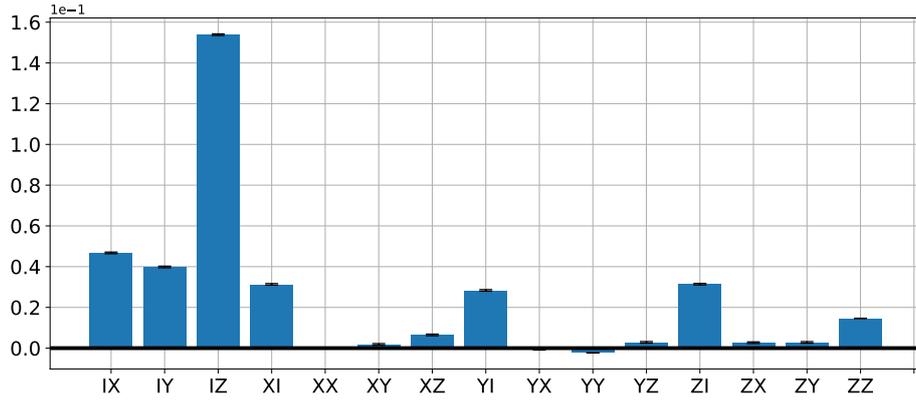
**Figure B.6:** Pauli twirl of the SPAM error matrix  $\chi_{\text{err}}$  corresponding to the map  $\Lambda_{\text{meas}}$  characterizing the SPAM errors. Note the strong correlation with figure B.5b.



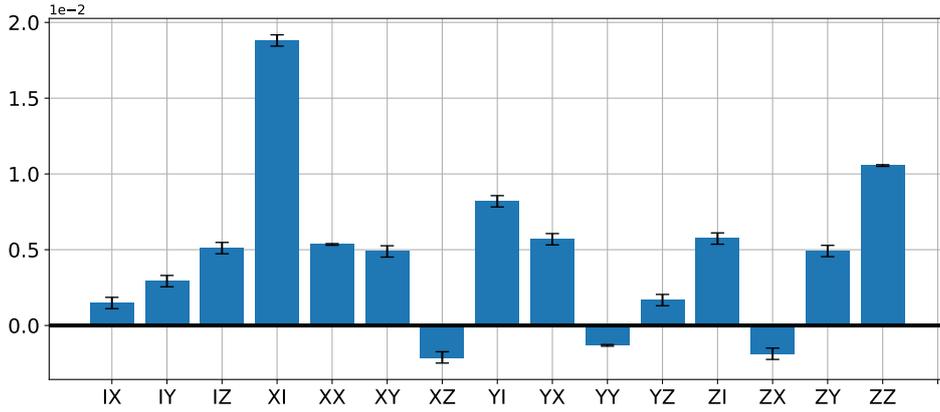
**Figure B.7:** Tomography results with SPAM filtering for the fault-tolerant (a) and normal (b) SWAP circuit. These representations do not reflect CP maps, so the CP subroutine still needs to be applied. Here the process matrices  $\chi$  are shown.  $F_p^{\text{FT}} = 0.640$  for the fault-tolerant SWAP and  $F_p^{\text{NFT}} = 0.929$  for the normal SWAP.



**Figure B.8:** Tomography results with SPAM filtering for the fault-tolerant (a) and normal (b) SWAP circuit. These representations do not reflect CP maps, so the CP subroutine still needs to be applied. Here the error matrices  $\chi^{\text{err}}$  are shown.  $F_p^{\text{FT}} = 0.640$  for the fault-tolerant SWAP and  $F_p^{\text{NFT}} = 0.929$  for the normal SWAP.



(a) Fault-tolerant SWAP



(b) Non fault-tolerant SWAP

**Figure B.9:** Pauli twirl of the filtered error matrices  $\chi_{\text{err}}$  of the (a) fault-tolerant SWAP and (b) normal implementation. Note the use of different scales for the two graphs. The CP subroutine has not been applied, resulting in negative values in the Pauli twirl. By taking the absolute values of  $p_i$  in the calculation of  $r$ , the error ratio can be approximated for these maps. The fault-tolerant implementation then has an error ratio of  $r^{\text{FT}} = 0.0957 \pm 0.0029$ , and the normal implementation then has an error ratio of  $r^{\text{NFT}} = 0.907 \pm 0.0129$ . Compare with the results in section 8.2.