Advanced Methods for Quasiprobabilistic Quantum Error Mitigation

ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Advanced Methods for Quasiprobabilistic Quantum Error Mitigation

Master Thesis

Christophe Piveteau

September 28, 2020

Advisors: Dr. D. Sutter, Dr. S. Woerner, Prof. Dr. R. Renner
Institute of Theoretical Physics, ETH Zürich

Abstract

Current quantum computers are plagued by prohibitive amounts of noise, which complicates the experimental realization of useful quantum algorithms that outperform classical computers. Quantum error mitigation techniques could constitute a potential avenue to demonstrate this feat without the need for fault-tolerant quantum error correction. One method in this family of mitigation techniques is the quasiprobability method introduced by Temme *et al.* [1]. It simulates a noise-free quantum computer with a noisy one, with the caveat of only producing the correct expected values of measurement observables. The cost of a quasiprobability simulation manifests as a sampling overhead which scales exponentially in the number of error-mitigated gates in the circuit. In this thesis we aim to reduce the exponential basis of that overhead, which in turn allows the application of the quasiprobability method to deeper quantum circuits. A central result is the introduction of a novel scheme, which we call Stinespring algorithm, that aims to choose the quasiprobability decomposition in a noise-aware manner. Along the way, we introduce a generalization of the quasiprobability method, which we denote approximate quasiprobability method, that allows for a tradeoff between an approximation error and the sampling overhead. This method is already interesting on its own and we present a few potential applications. Finally, we present some ideas how quantum error correction can benefit from the quasiprobability method. This final part is separate to the other topics of the master thesis and gives a rough overview of a new research project that started towards the end of the thesis and that we will continue to pursue in the future.

Contents

Chapter 1

Introduction

Quantum computing holds the promise for significant advancements in many fields such as quantum chemistry, material science, optimization and machine learning. Recent decades have seen a surge in experimental and theoretical efforts towards the realization of practical quantum computers, and while many obstacles have already been overcome, there is still a long way to go before quantum computers might outperform classical computers for useful problems. The major obstacle towards large-scale fault tolerant quantum computing is the experimental difficulty of coherently storing and manipulating quantum information. In contrast to their transistor-based classical counterparts, quantum computers suffer from a significant amount of noise, which physically stems from uncontrolled interactions between the qubit systems and their environment. This noise accumulates over the execution of a quantum algorithm and thus limits experimenters to the execution of restrictively small quantum circuits.

A major breakthrough was the realization that unwanted noise could be suppressed by the use of quantum error correction[2]. This family of schemes encodes the quantum information of a qubit into multiple physical qubits in order to protect it from local noise. Quantum error correction requires a polylogarithmic overhead in terms of number of qubits and quantum gates, and furthermore it only works if the hardware operations exhibit fidelities above a certain threshold. This polylogarithmic overhead is tame when speaking of asymptotics, but currently available quantum hardware does not fulfill the requirements for quantum error correction [3] and it is estimated that it might take decades to achieve this goal.

Considering this problem, it is not a surprise that recent research on alternative methods to mitigate noise on quantum hardware has gained traction in the last few years. Multiple schemes have been proposed [4, 5, 1, 6] that aim to reduce the effect of noise while also being significantly easier to implement than quantum error correction. All of these methods have some kind

of drawback that prohibits them from achieving large-scale fault tolerant quantum computation. The hope is rather to enable current or near-term quantum hardware to demonstrate a speed-up on useful tasks compared to classical computers. The term *Quantum Error Mitigation* is often used to group these methods.

One specific method in the family of quantum error mitigation techniques is the *quasiprobability method* (sometimes also known as *probabilistic error cancellation*). The central idea is to decompose the ideal (noise-free) quantum circuit into a quasiprobabilistic mixture of noisy circuits that one can efficiently implement on a given hardware. We will call such mixtures *quasiprobability decompositions*. The origin of the method lies in the domain of classical simulators for quantum circuits [7, 8]. The main insight of Temme *et al.* [1] was that the method could easily be lifted to a quantum error mitigation setting: Instead of simulating a noise-free quantum computer with a classical computer, one rather simulates a noise-free quantum computer with a noisy quantum computer. For the rest of this thesis we will restrict ourselves to quasiprobability simulations corresponding to the latter setting.

1.1 Previous Works

The quasiprobability method exhibits a simulation overhead that manifests as an additional sampling cost. More precisely, the number of shots required to execute a circuits scales as $\mathcal{O}(C^{2\cdot\text{number of gates}})$ where $C \geq 1$ is the so-called C-factor of the quasiprobability decomposition. This quantity encapsulates how strongly the method has to compensate for the noise in the quantum system. Notably, C goes to 1 in the limit where the noise vanishes, making the simulation overhead disappear. This exponential cost restricts the quasiprobability method to shallow quantum circuits.

By the above reasoning it is evident that one wants to find quasiprobability decompositions that exhibit the smallest possible C-factor. The arguably most difficult part of finding a suitable quasiprobability decomposition is how to choose the noisy quantum circuits into which we decompose the ideal quantum circuit. We denote this set of circuits the *decomposition set*.

Temme *et al.* [1] realized, under the assumption that the decomposition set is already fixed, that the optimal quasiprobability decomposition could be expressed as a linear program. However, they did not specify a procedure to choose a decomposition set that would suffice to decompose an arbitrary circuit under the constraint of some arbitrary noise. Endo *et al.* [9] introduced an explicit decomposition set that fulfills this requirement. However, this decomposition set is not adapted to the circuit to be decomposed, nor to the hardware noise present to the hardware. Ideally one would optimize over the decomposition set in order to achieve a lower C-factor.

In this thesis, we introduce a novel method, called the *Stinespring algorithm*, that aims to solve this problem. We make extensive use of mathematical optimization techniques for convex and non-convex problems to obtain a decomposition set that is adapted to the hardware noise. Simulation results indicate that our method significantly reduces the C-factor of the quasiprobability decomposition. On the way towards the Stinespring algorithm, we also introduce a new technique which we call the *approximate quasiprobability decomposition*, which is interesting on its own. Instead of perfectly simulating a certain circuit, we allow for a small approximation error. This enables us to make a tradeoff between the approximation quality and the sampling overhead of the quasiprobability method. We will illustrate our results with simulations and hardware demonstrations whenever possible.

1.2 Outline of Thesis

In Chapter 2 we review the quasiprobability method in detail and investigate the associated simulation overhead and the difficulty of finding a decomposition set. We also introduce the channel difference decomposition, which serves as a theoretical bound on how small the C-factor of a quasiprobability decomposition can be. In Chapter 3 we introduce the approximate QPD method and demonstrate some direct applications of it. In Chapter 4 we present the Stinespring algorithm as well as the necessary building blocks required to realize it.

Chapter 5 introduces a separate topic of the master thesis that is not directly related to Chapters 3 and 4. The aim of this section is to investigate how the quasiprobability method could be used to facilitate the implementation of quantum error correction on near-term quantum hardware. This is still an ongoing research effort, and this part of the manuscript aims to document some first steps towards the described goal.

1.3 Overview of Contributions

We summarize the novel contributions from this thesis:

- Many details in the presentation of Chapter 2 have not been stated in such precision in previous literature. However, we estimate that many of these contributions are evident for someone knowledgeable in the domain.
- The channel difference decomposition (Section 2.6).
- The approximate QPD and its applications (Chapter 3).
- The Stinespring algorithm (Chapter 4).

- A few results on the application of the quasiprobability method to quantum error correction (Chapter 5).

1.4 Notation

In the following section we introduce some recurring notation used throughout this thesis.

Hilbert spaces: We denote Hilbert spaces with alphabetic subscripts $\mathcal{H}_A, \mathcal{H}_B, \ldots$ and interchangeably reference them by $A, B, \ldots$. The tensor products of two spaces $\mathcal{H}_A \otimes \mathcal{H}_B$ is also written as AB. All Hilbert spaces in this thesis will be finite-dimensional.

Density matrices: We denote the set of density matrices on the system A as

$$\mathcal{S}(A) := \{\rho \in \mathrm{End}(\mathcal{H}_A) | \rho \geq 0, \mathrm{Tr}[\rho] = 1\} \quad .$$

Linear operators: We denote linear maps from a system A to a system B as

$$\begin{aligned} \mathrm{L}(A, B) &:= \mathrm{Hom}(\mathcal{H}_A, \mathcal{H}_B) \\ \mathrm{L}(A) &:= \mathrm{L}(A, A) \quad . \end{aligned}$$

Unitary operators: We denote the unitary maps on a system A as

$$\mathrm{U}(A) := \{U \in \mathrm{L}(A) | U^\dagger = U\} \quad .$$

Quantum channels:

$$\begin{aligned} \mathrm{TP}(A, B) &:= \{\mathcal{E} \in L(L(A), L(B)) | \mathcal{E} \text{ is trace-preserving } \} \\ \mathrm{CP}(A, B) &:= \{\mathcal{E} \in L(L(A), L(B)) | \mathcal{E} \text{ is completely positive } \} \\ \mathrm{TPCP}(A, B) &:= \mathrm{TP}(A, B) \cap \mathrm{CP}(A, B) \end{aligned}$$

Induced quantum operation: For a linear map $O \in \mathrm{L}(A, B)$ we define the induced quantum operation $[O] \in \mathrm{L}(\mathrm{L}(A), \mathrm{L}(B))$ as

$$[O](\rho) := O\rho O^\dagger \quad .$$

Chapter 2

Quasiprobability Method

Consider n-qubit and m-qubit Hilbert spaces $\mathcal{H}_A, \mathcal{H}_B$ and a linear operator $\mathcal{F} \in \mathrm{L}(\mathrm{L}(A), \mathrm{L}(B))$. We would like to execute $\mathcal{F}$ on n qubits of our quantum computer which are initially in the state ρ_{in}. This task is obviously nonsensical if $\mathcal{F}$ is a non-physical operation[1]. Even if $\mathcal{F}$ is physical, or even unitary, the quantum computer will at best only be able to approximate its execution, as current quantum hardware suffers from significant amounts of noise.

The quasiprobability method allows us to simulate a noiseless execution of $\mathcal{F}$ using noisy operations. In contrary to quantum error correction, we don't encode our quantum information in a larger space, so a logical qubit still corresponds to a physical qubit. However, the quasiprobability simulation only works *on average*, i.e. one doesn't actually obtain access to the (possibly non-physical) state $\mathcal{F}(\rho_{in})$ itself, but rather to a random physical state which is sampled independently in every run of the experiment. This random state exhibits the same expectation values as $\mathcal{F}(\rho_{in})$ for the outcomes of any measurement, provided that we perform a small amount of classical post-processing.

A typical application would be that $\mathcal{F}$ is a unitary operation representing some quantum gate. If the the individual gates composing a quantum algorithm suffer from too much noise, then the measurement statistics at the end of the circuit will differ so strongly from the ideal ones, that it becomes impossible to retrieve any useful information out of them. The quasiprobability method allows us to circumvent that noise, as long as we are only interested in expectation values of observables at the end of the circuit. This setting is interesting for many types of quantum algorithms, such as variational quantum eigensolvers [10, 11], iterative quantum phase estimation [12, 13], recursive QAOA [14] and generally all quantum circuits that exhibit a deterministic measurement outcome when executed without noise. Interestingly,

[1] We use the term physical operation interchangeably with TPCP.

we will also encounter settings where it might be interesting to simulate the execution of a non-physical map $\mathcal{F}$.

2.1 Quasiprobability Sampling

Definition 2.1. A quasiprobability decomposition (QPD) of the linear operator $\mathcal{F} \in \mathrm{L}(\mathrm{L}(A), \mathrm{L}(B))$ is a finite set of tuples $\{(a_i, \mathcal{E}_i)\}_i$ with $a_i \in \mathbb{R}$ and quantum channels $\mathcal{E}_i \in TPCP(A, B)$ such that

$$\mathcal{F}(\rho) = \sum_i a_i \mathcal{E}_i(\rho)\,, \quad \forall \rho \in \mathrm{L}(A)\,. \tag{2.1}$$

We call $\{\mathcal{E}_i\}_i$ the *decomposition set* of the QPD.

The quasiprobability method requires a QPD $\{(a_i, \mathcal{E}_i)\}_i$ where the maps $\mathcal{E}_i$ correspond to operations that can be implemented on the quantum hardware, e.g. they might correspond to the channel of a noisy quantum gate. In practice, these could be obtained by doing tomography and/or by using prior knowledge of the experimental noise. Suppose now that we are interested in the expectation value of a projective measurement described by a Hermitian operator O, which would be performed after the operation $\mathcal{F}$, i.e. we would like to obtain $\mathrm{Tr}[O\mathcal{F}(\rho)]$, given a certain input state ρ. The linearity of the trace implies

$$\mathrm{Tr}[O\mathcal{F}(\rho)] = C \sum_i \frac{|a_i|}{C} \mathrm{sgn}(a_i) \mathrm{Tr}[O\mathcal{E}_i(\rho)]\,, \tag{2.2}$$

where we have introduced the C-factor $C := \sum_i |a_i|$. The right-hand side of Equation (2.2) naturally gives us a way to construct an unbiased[2] estimator for $\mathrm{Tr}[O\mathcal{F}(\rho)]$, while only having access to the operations $\mathcal{E}_i$ of the noisy hardware, as seen in Figure 2.1. We choose a random number i with probability $|a_i|/C$ and then perform the operation $\mathcal{E}_i$ followed by the measurement O. The output of the estimator is the outcome of the measurement weighted by the coefficient $C\mathrm{sgn}(a_i)$. By sampling this estimator many times and considering the average value, we can obtain an arbitrarily precise estimate of the true expectation value $\mathrm{Tr}[O\mathcal{F}(\rho)]$.

Although generally the Hilbert spaces A and B can have different dimensionality, we will mostly restrict ourselves to the case where $\mathcal{H}_A = \mathcal{H}_B$ as this is the relevant setting for error mitigated quantum computing.

Remark 2.2 (Limitations of Tomography)**.** In practice the $\mathcal{E}_i$ are imperfect estimates of the true underlying quantum channels, produced by tomography. Unfortunately, tomography fundamentally cannot give us an arbitrarily

[2] An estimator is called unbiased if its expected value is exactly the true value.

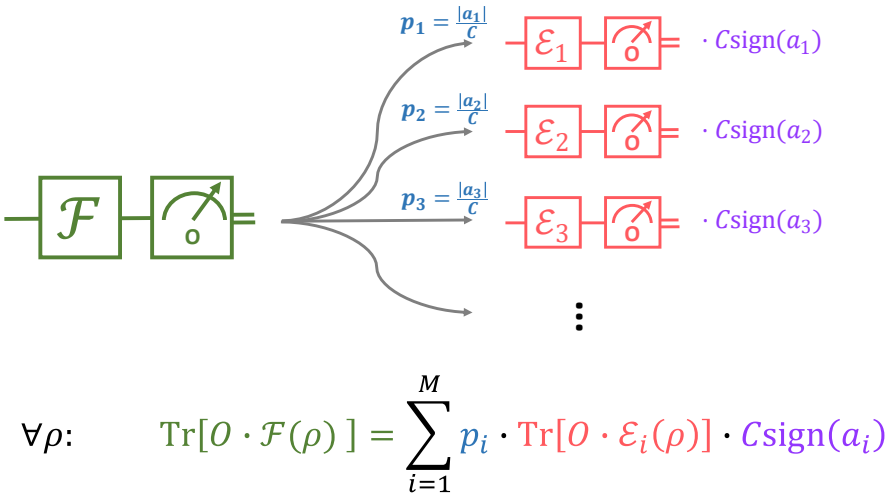


Figure 2.1: Graphical representation of the quasiprobability method. The ideal quantum gate U is randomly replaced by a quantum channel $\mathcal{E}_i$ that our hardware can implement. The output of the measurement O must be weighted according to the sampled operation.

precise estimate of the true channels, due to state preparation and measurement (SPAM) errors. When we have an erroneous knowledge of the $\mathcal{E}_i$, our QPD will have erroneous coefficients a_i, which again translate into an error in our estimator of the expectation value. Endo *et al.* have shown [9] that this problem can be circumvented by using gate set tomography (GST), a specific type of tomography, which gives self-consistent estimates of the SPAM errors. While the estimates of the $\mathcal{E}_i$ are also erroneous with GST, the authors show that the estimator of the quasiprobability method remains unbiased. For more information on SPAM errors and GST we refer the reader to [15]. Recent works have also explored the possibility of learning the QPD coefficients ab-initio without the need of explicit tomography [16], though these methods come with their own share of technical difficulties and shortcomings. An alternative solution could also be the utilization of novel techniques that have recently been proposed, which allow the characterization of quantum channels without being affected by the error induced by SPAM errors [17].

For small-scale experiments on current quantum hardware the issue of erroneous tomography is not too problematic , since the single-qubit gate errors occurring during SPAM are significantly lower than the two-qubit gate errors that can be corrected using the quasiprobability method. Therefore one can still expect a significant improvement in accuracy by using the quasiprobability method.

In light of the above discussion we will ignore the issue of tomography errors for the rest of the thesis, as it can be treated as a separate and independent problem to the ones we will address.

2.2 Sampling Overhead and the C-factor

The quasiprobability estimator constructed above does generally not retain the full statistics of the measurement outcome of O, it only conserves the expectation value. In fact, the variance of the estimator increases with C and one requires $\mathcal{O}(C^2)$ more shots to estimate $\mathrm{Tr}[O\mathcal{F}(\rho)]$ compared to the case where $\mathcal{F}$ is implemented perfectly.

To formulate the above statement more precisely, we must introduce some additional notation. Consider a random variable X which is distributed according to the outcomes of the measurement of O on $\mathcal{F}(\rho)$. Similarly, Y_i are random variables distributed according to the outcomes of the measurement O on $\mathcal{E}_i(\rho)$. For simplicity, we restrict ourselves to the case where O is a binary measurement with the two possible outcomes 0 and 1. This corresponds to the setting that is relevant for error-mitigated quantum computing. So the X and Y_i are binomial distributions and thus fully characterized by their expectation values, which we denote by μ and m_i. Let I be a discrete random variable taking as value one of the indices i with probability $\mathbb{P}[I = i] = |a_i|/C$. The outcome of our quasiprobability sampling estimator is thus modelled by a random variable $Z := C\mathrm{sgn}(a_I)Y_I$. The behavior of the variance of Z compared to the reference variance of X (i.e. if we could implement $\mathcal{F}$ perfectly on the hardware) is given by following lemma:

Lemma 2.3. *Consider the setup described above. Then,*

$$\mathrm{Var}[Z] \leq \mathrm{Var}[X] + C^2 \tag{2.3}$$

with equality for some choice of m_i and μ.

If we draw N samples of our quasiprobability sampling estimator, the variance of the averaged result will be reduced to $\mathrm{Var}[Z]/N$. This provides us the practical implication of Lemma 2.3: The number of samples required to reduce the error of our estimator to some fixed constant scales as $\mathcal{O}(C^2)$.

Proof. We can make use of the law of total variance, i.e. ,

$$\mathrm{Var}[Z] = \mathbb{E}_I[\mathrm{Var}[Z|I]] + \mathrm{Var}_I[\mathbb{E}[Z|I]] \tag{2.4}$$

$$= \sum_i \mathbb{P}[I = i]\mathrm{Var}[Z|I = i] + \sum_i \mathbb{P}[I = i](\mathbb{E}[Z|I = i] - \mathbb{E}[Z])^2 . \tag{2.5}$$

By inserting the respective definitions of I, Z and Y_i we find

$$\mathrm{Var}[Z] = \sum_i |a_i| \left(Cm_i(1 - m_i) + \frac{(C\mathrm{sgn}(a_i)m_i - \mu)^2}{C} \right) , \tag{2.6}$$

which can be rewritten as

$$\mathrm{Var}[Z] = \frac{1}{C}\sum_i |a_i|[C^2 m_i - 2C\,\mathrm{sgn}(a_i) m_i \mu + \mu^2] = C\sum_i |a_i| m_i - 2\mu^2 + \mu^2 \tag{2.7}$$

where we used $\sum_i a_i m_i = \mu$. Therefore we get

$$\mathrm{Var}[Z] = \mathrm{Var}[X] + C\sum_i |a_i| m_i - \mu\,. \tag{2.8}$$

The second term on the right hand side reaches it maximum of C^2 when all $m_i = 1$ and the third term vanishes when $\mu = 0$, so the desired inequality follows directly. We can easily construct an example of random variables Y, X_i that saturates these bounds, while still fulfilling $Y = \sum_i a_i X_i$ by choosing $m_i = 1, a_i = \pm 1$ and $\mu = 0$. □

The above considerations motivate us to choose the C-factor as a measure of quality for a QPD. A desired operation $\mathcal{F}$ will typically have infinitely many QPDs into operations that our hardware can execute. Hence, it is important to find a QPD with the lowest possible C-factor. One way this can be done will be presented in Section 2.5.

Technically, the C-factor is a worst-case metric for the increase in variance of the quasiprobability sampling, as there is no guarantee that the inequality in Lemma 2.3 is close to being saturated. It is impossible to get a better bound in practice, as we do not have access to the m_i during the computation of the optimal QPD. This is because typically our corrected gate will be part of a much larger circuit where many other gates, occurring before and after the gate in question, will also be corrected by the quasiprobability method. Therefore, one has to make do with using C as a proxy estimate for the sampling overhead.

2.3 Quasiprobability Decomposition of Multiple Operations

The above introduction to the quasiprobability method considers the simplest case where one simulates only a single operation. In this section, we seek to answer how this technique can be applied to large quantum circuits consisting of many different gates. One might be tempted to try to directly find a QPD of the quantum circuit as a whole. However, this turns out to be unfeasible in practice, as finding a QPD of a many-qubit operation is an optimization problem that requires an amount of resources exponential in the number of qubits (the details of that statement will be discussed in Section 2.5). So the more useful approach is to find a QPD of each quantum gate individually, and then combine them together into one large QPD of

the whole circuit. As we will see further below, this local approach is computationally tractable, but we have to pay a cost: We have no guarantee that our QPD has the lowest possible C-factor, since the optimal total QPD is unlikely to be separable into QPDs of the individual gates.

In the following we describe how the QPDs of two separate operations naturally lead to a QPD of the concatenation of both. Note that showing this for the concatenation also directly implies that it also works for the tensor product of the two operations. Using this argument iteratively, one can construct a QPD of the total circuit from the QPDs of the individual gates.

Consider two linear operators $\mathcal{F}_1 \in \mathrm{L}(\mathrm{L}(A), \mathrm{L}(A'))$, $\mathcal{F}_2 \in \mathrm{L}(\mathrm{L}(A'), \mathrm{L}(A''))$ applied in succession on some quantum state $\rho \in \mathcal{S}(A)$. Suppose we are given a QPD of each, which we call $\{(a_i, \mathcal{E}_i)\}$ and $\{(b_j, \mathcal{G}_j)\}$. Together they naturally lead to a QPD of $\mathcal{F}_2 \circ \mathcal{F}_1$:

$$(\mathcal{F}_2 \circ \mathcal{F}_1)(\rho) = C_1 C_2 \sum_{i,j} \frac{|a_i \cdot b_j|}{C_1 C_2} \mathrm{sgn}(a_i \cdot b_j) (\mathcal{G}_j \circ \mathcal{E}_i)(\rho) \,, \quad \forall \rho \in \mathrm{L}(A) \,. \tag{2.9}$$

We see that the combined C-factor scales in a multiplicative way. Therefore, when one has a circuit consisting of N gates corrected by the quasiprobability method, the C-factor of the total circuit is given by the product of the individual C-factors, i.e., $C_{\mathrm{tot}} = \prod_{i=1}^{N} C_i$. This exponential cost in the number of gates is a heavy price to pay, it is maybe not too surprising. If there were no such drawback, the quasiprobability method would allow for fault-tolerant quantum computation without the harsh resource requirements found in quantum error correction (high gate fidelities and large number of physical qubits).

For practical usage it is not tractable to store the QPD of the full circuit itself, as that would require storing an amount of quasiprobability amplitudes that is exponential in the number of gates. Instead we store the QPDs of the individual quantum gates and due to the product form in Equation (2.9), one can still implement the quasiprobability sampling estimator efficiently. Consider a sequence of m operation $\mathcal{F}_m \circ \mathcal{F}_{m-1} \circ \cdots \circ \mathcal{F}_1$ followed by a measurement described by the observable O. For each operation $\mathcal{F}_k$ a QPD $\{(a_i^{(k)}, \mathcal{E}_i^{(k)})\}$ with C-factor C_k is given. Our estimator starts by sampling a random number i_1 according to the probabilities $\{|a_i^{(1)}|/C_1$ and executing the operation $\mathcal{E}_{i_1}^{(1)}$. In a second step we sample a random number i_2 according to the probabilities $|a_i^{(2)}|/C_2$ and execute the operation $\mathcal{E}_{i_2}^{(2)}$. This procedure is repeated for all m operations while keeping track of all indices $i_1, \ldots, i_m$ sampled along the way. At the very end we measure the observable O on the system. The estimator then outputs the outcome of that measurement multiplied by $\mathrm{sgn}(\prod_{i=k}^{m} a_{i_k}^{(k)}) \prod_{i=k}^{m} C_k$.

Remark 2.4 (Assumptions on noise model)**.** The combination of QPDs of individual parts of a circuit into a QPD of the total circuit, as described

in this section, can only work under certain assumptions on the gate noise. More precisely the noise must be localized and Markovian. In looser term this means that the noise on any quantum gate must be completely uncorrelated with other noise and independent on what operations were performed previously on the circuit.

Unfortunately, real quantum hardware exhibits significant correlations between noise of operations which are spatially and temporally separated[3] [18, 19, 20]. The first works on the quasiprobability method [1, 9] completely neglected this issue, by assuming localized and Markovian noise in a first order approximation. Some more recent research has shown that cross-correlations can be tackled by using variants of the quasiprobability method which do not rely on tomography to find the optimal quasiprobability coefficients [16]. We are aware of other, as of yet unpublished, techniques which can tackle the problem. For the rest of the thesis we will ignore this issue, as it is independent to the problems that we will consider.

2.4 Existence of a QPD and the Endo Basis

In this section we would like to answer following question: For a given quantum hardware, can we be certain that there even exists a QPD of $\mathcal{F}$ into operations that the hardware can implement? This question may seem rather difficult, as its answer appears to depend on the exact details of the capabilities of the hardware, namely what kind of quantum operations it can implement and at what fidelity it does so. Endo *et al.* [9] realized that the requirements on the hardware are actually very few and always fulfilled on a realistic quantum computer that is accurate enough, assuming that $\mathcal{F}$ is Hermitian-preserving. This assumption is very natural, as the $\mathcal{E}_i$ must be Hermitian-preserving (due to the physicality constraint), and any linear combination of Hermitian-preserving maps is itself Hermitian-preserving. Note that we don't require the $\mathcal{E}_i$ (and therefore $\mathcal{F}$) to be trace-preserving though, as non-trace-preserving maps can be simulated using measurements and postselection, as discussed at a later point in this section.

We start with the simplest case where $\mathcal{F}$ is a single-qubit operation. Let's consider 16 single-qubit operations, listed in Table 2.1, which can all be realized by the successive execution of the Hadamard gate $[H]$[4], the phase gate $[S]$[5] and/or a measurement-postselection operation $[P_0]$[6]. The measurement-

[3] These non-local errors are often referred to as *crosstalk* errors.

[4] $H = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$

[5] $S = \begin{pmatrix} 1 & 0 \\ 0 & \mathrm{i} \end{pmatrix}$

[6] $P_0 := |0\rangle\langle 0| = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}$

postselection operation can be simulated in the context of the quasiprobability sampling estimator by performing a measurement in the computational basis and aborting if the outcome is 1, where aborting means that the estimator outputs 0. Therefore any quantum computer able to implement Clifford gates and measurements in the computational basis can also implement these operations, at least approximately. It can be easily shown that these 16 operations are linearly independent, e.g. by showing that their PTMs are linearly independent. Since any Hermitian-preserving single-qubit operation has a 4×4 PTM with real entries[7], we deduce that the mentioned set of 16 operations forms a basis of the space of Hermitian-preserving operations. From now on we will call this set of basis operations the *Endo Basis*.

Note that the last 6 operations in Table 2.1 are not trace-preserving, due to the measurement-postselection operator. The inclusion of non-trace-preserving operations is necessary, as elsewise we could only decompose maps $\mathcal{F}$ which are proportional to a trace-preserving map (as any linear combination of trace-preserving maps will be proportional to another trace-preserving map).

$[\mathbb{1}]$ (no operation)		
$[\sigma^X]$		$= [H][S]^2[H]$
$[\sigma^Y]$		$= [H][S]^2[H][S]^2$
$[\sigma^Z]$		$= [S]^2$
$[R_X]$	$= [\frac{\mathbb{1}+i\sigma^X}{\sqrt{2}}]$	$= [H][S]^3[H]$
$[R_Y]$	$= [\frac{\mathbb{1}+i\sigma^Y}{\sqrt{2}}]$	$= [S][H][S]^3[H][S]^3$
$[R_Z]$	$= [\frac{\mathbb{1}+i\sigma^Z}{\sqrt{2}}]$	$= [S]^3$
$[R_{YZ}]$	$= [\frac{\sigma^Y+\sigma^X}{\sqrt{2}}]$	$= [H][S]^3[H][S]^2$
$[R_{ZX}]$	$= [\frac{\sigma^Z+\sigma^Y}{\sqrt{2}}]$	$= [S]^3[H][S]^3[H][S]^3$
$[R_{XY}]$	$= [\frac{\sigma^X+\sigma^X}{\sqrt{2}}]$	$= [H][S]^2[H][S]^3$
$[\pi_X]$	$= [\frac{\mathbb{1}+\sigma^X}{2}]$	$= [S][H][S][H][P_0][H][S]^3[H][S]^3$
$[\pi_Y]$	$= [\frac{\mathbb{1}+\sigma^Y}{2}]$	$= [H][S]^3[H]P_0[H][S][H]$
$[\pi_Z]$	$= [\frac{\mathbb{1}+\sigma^Z}{2}]$	$= [P_0]$
$[\pi_{YZ}]$	$= [\frac{\sigma^Y+i\sigma^Z}{2}]$	$= [S][H][S][H][P_0][H][S][H][S]^3$
$[\pi_{ZX}]$	$= [\frac{\sigma^Z+i\sigma^X}{2}]$	$= [H][S]^3[H][P_0][H][S][H][S]^2$
$[\pi_{XY}]$	$= [\frac{\sigma^X+i\sigma^Y}{2}]$	$= [P_0][H][S]^2[H]$

Table 2.1: 16 basis operations constituting the Endo basis. All these operations can be realized using Hadamard gates H, phase gates S and measurement-postselection operations P_0.

The above argument requires our computer to be able to implement $[H], [S]$

[7] We refer the reader to [15] for an exact derivation.

and $[P_0]$ exactly. While this is not the case, a useful quantum computer will be able to approximate them reasonable well. In fact, as long as the fidelity of this approximation is not too bad, the approximate Endo operations will still remain linearly independent. This is formalized in following lemma:

Lemma 2.5 ([9]). *Denote by $\mathcal{B}_i^{(0)}$ the elements of the Endo basis and by $\mathcal{B}_i$ an approximation thereof. These operators are represented as their PTM, i.e. as real 4×4 matrices. Define*

$$\varepsilon_{max} := \max \left\{ \left\| \mathcal{B}_i - \mathcal{B}_i^{(0)} \right\|_\infty | i = 1, \ldots, 16 \right\}. \tag{2.10}$$

The $\mathcal{B}_i$ are linearly independent when $\varepsilon_{max} < \frac{13-3\sqrt{17}}{32} \approx 0.0351$.

Actually, the $\mathcal{B}_i$ are very likely to remain linearly independent even above this threshold.

The construction of a universal decomposition set can be straightforwardly translated to n-qubit operations $\mathcal{F}$ for $n > 1$. The basis is then given by the tensor products of the Endo basis elements. Therefore we can decompose any $\mathcal{F}$ in at most 16^n basis operations.

2.5 Finding the Optimal QPD for a Fixed Decomposition Set

In Section 2.4 we have shown that in practice there always exists a decomposition set into which we can find a QPD of the desired n-qubit operation $\mathcal{F}$. Let's now consider the setting where we have a fixed decomposition set $\{\mathcal{E}_i\}_i$ and we would like to find the optimal (in terms of lowest possible C-factor) quasiprobability coefficients a_i s.t. Equation (2.1) is fulfilled. If our decomposition set consists of linearly independent operations, which is e.g. the case for the Endo basis, then this problem corresponds to solving a set of linear equations. But in practice one is often confronted with the more general case where there is an infinity of possible decompositions, so we require some method to pick out the best one of them.

Temme *et al.* [1] realized that this optimization problem can be written in terms of a linear program (LP):

$$\begin{cases} \min\limits_{a_i \in \mathbb{R}} & \sum_j |a_j| \\ \text{s.t.} & \mathcal{F} = \sum\limits_j a_j \mathcal{E}_j \,. \end{cases} \tag{2.11}$$

To see that this is indeed a linear program one can introduce additional optimization variables b_i and rewrite the problem as

$$\begin{cases} \min\limits_{a_i \in \mathbb{R}, b_i \in \mathbb{R}} & \sum_j b_j \\ \text{s.t.} & \mathcal{F} = \sum\limits_j a_j \mathcal{E}_j \text{ and } -b_i \leq a_i \leq b_i \,. \end{cases} \tag{2.12}$$

To translate the constraint $\mathcal{F} = \sum_j a_j \mathcal{E}_j$ for a numerical implementation of a mathematical optimization algorithm, one has to choose an isomorphic matrix representation of the involved quantum operations, such as the Choi representation or the PTM representation.

As a final remark we note that the number of equality constraints in the LP (2.11) is 16^n[8] where n is the number of qubits involved in the operation. This exponential cost means that the LP can not be used to compute a QPD of large circuit blocks. In practice, one typically uses the LP (2.11) to find the QPD of every one- and two-qubit gate in a given circuit and then uses the construction in Section 2.3 to implement a quasiprobability sampling estimator for the whole circuit.

2.6 Channel Difference Decomposition

In this section we will expand on the work from Section 2.5 and ask following question: What is the optimal QPD of an operation $\mathcal{F}$ without having fixed the decomposition set previously? More concretely we would like to generalize the optimization problem (2.11) to not only optimize over the a_i, but also over the $\mathcal{E}_i$. The resulting QPD will not be directly useable for the quasiprobability method, as the obtained $\mathcal{E}_i$ might not be implementable by a quantum computer in practice. Rather we will assume for the moment that we have some kind of ideal quantum computer that can implement any physical map, so the only restriction on the $\mathcal{E}_i$ will be that they must be trace-preserving and completely positive. The fact that the $\mathcal{E}_i$ are trace-preserving implies that we will only be able to find QPDs for $\mathcal{F}$ that are trace-preserving, or at least related to a trace-preserving map by a scaling factor. The results from this section will turn out to be useful for the Stinespring algorithm, which is covered in Chapter 4.

For the following section, it will be convenient to represent quantum operations by their Choi matrices. We denote the Choi matrix of $\mathcal{F}$ by $\Lambda_{\mathcal{F}}$ and our goal will be to construct a finite set of Choi matrices $\{\Lambda_{\mathcal{E}_i}\}_i$ which correspond to a decomposition set $\{\mathcal{E}_i\}_i$. The Choi representation is very

[8] The exact value 16^n holds under the assumption that we represent the channels as Choi or PTM matrices. When using the Choi representation, the equations contain complex numbers, so one actually has $2 \cdot 16^n$ real equality constraints.

convenient, as it allows us to formulate the TPCP-condition on the decomposition basis in a straightforward way: $\Lambda_{\mathcal{E}_i} \geq 0$ and $\mathrm{Tr}_2[\Lambda_{\mathcal{E}_i}] = \frac{1}{2^n}\mathbb{1}$ for all i where Tr_2 stands for the partial trace over the ancillary Hilbert space of the Choi-Jamiolkowski isomorphism and n is the number of qubits involved in $\mathcal{F}$.

With the above preparation, we can now easily write down our optimization problem as

$$\begin{cases} \min\limits_{a_i \in \mathbb{R}, \Lambda_{\mathcal{E}_i} \in \mathbb{C}^{4^n \times 4^n}} & \sum\limits_{j=1}^{N} |a_j| \\ \text{s.t.} & \Lambda_{\mathcal{F}} = \sum\limits_{j=1}^{N} a_j \Lambda_{\mathcal{E}_j} \\ & \Lambda_{\mathcal{E}_i} \geq 0 \\ & \mathrm{Tr}_2[\Lambda_{\mathcal{E}_i}] = \frac{1}{2^n}\mathbb{1}\,. \end{cases} \tag{2.13}$$

We need to be careful with how many indices i we need to sum over. If an arbitrary large N were required to reach a minimum for a general $\mathcal{F}$, then the optimization problem (2.13) would be very difficult to solve numerically. Fortunately, this is not the case:

Lemma 2.6. *If the optimization problem (2.13) reaches a minimum, then there exists an equivalent (in terms of C-factor) minimum with $N = 2$.*

Therefore we can generally choose $N = 2$.

Proof. We equivalently reformulate (2.13) by separating the positive and negative quasiprobability coefficients as

$$\Lambda_{\mathcal{F}} = \sum_{i=1}^{\tilde{N}} a_i^+ \Lambda_{\mathcal{E}_i}^+ - \sum_{i=1}^{\tilde{N}} a_i^- \Lambda_{\mathcal{E}_i}^- \tag{2.14}$$

with $a_i^\pm \geq 0$. Consider a set of variables $a_i^+, a_i^-, \mathcal{E}_i^+, \mathcal{E}_i^-$ that minimizes the optimization problem. We construct

$$a^+ := \sum_i a_i^+, \quad a^- := \sum_i a_i^-, \tag{2.15}$$

$$\mathcal{E}^+ := \frac{1}{a^+} \sum_i \mathcal{E}_i^+, \quad \mathcal{E}^- := \frac{1}{a^-} \sum_i \mathcal{E}_i^-\,. \tag{2.16}$$

These variable also fulfill the constraints: $\mathcal{F} = a^+\mathcal{E}^+ - a^-\mathcal{E}^-$ and it is easy to check that $\mathcal{E}^\pm$ are TPCP maps. The C-factor of this new solution is the same as the C-factor of the original solution, because

$$a^+ + a^- = \sum_i a_i^+ + \sum_i a_i^-\,. \tag{2.17}$$

□

Still, it is not clear yet when our problem exhibits feasible[9] solutions and whether a minimum is achieved in that case. These questions are addressed by following two lemmas:

Lemma 2.7. *If feasible solutions exist for the optimization problem (2.13), then the minimum is reached.*

Proof. This statement follows from the extreme value theorem by arguing that we are minimizing a continuous function over a compact set. So in the following we need to prove that the set of feasible solutions is compact.
The set of TPCP maps is clearly compact. For the $a_i^{\pm}$ we have to argue that we can bound their value. Take an arbitrary feasible solution of the problem that exhibits a C-factor which we denote C^*. Then we can restrict ourselves to the bounded region.

$$0 \leq a_i^{\pm} \leq \sum_j a_j^+ + \sum_i a_j^- \leq C^* \,. \tag{2.18}$$

One can easily check that adding the additional constraint $\Lambda_{\mathcal{F}} = \sum_i a_i \Lambda_{\mathcal{E}_i}$ to our compact set again results into a compact set. □

Lemma 2.8. *Consider a n-qubit quantum channel $\mathcal{F}$ for $n \leq 3$. The optimization problem (2.13) has a feasible solution if and only if $\mathcal{F}$ is Hermitian-preserving and proportional to a trace-preserving map.*

Proof. Suppose one is given a feasible solution. Then

$$\mathrm{Tr}_2[C_{\mathcal{F}}] = \sum_i a_i^+ \mathrm{Tr}_2[C_{\mathcal{E}_i}^+] - \sum_j a_j^- \mathrm{Tr}_2[C_{\mathcal{E}_j}^+] = (\sum_i a_i^+ - \sum_j a_j^-) \frac{1}{2^n} \mathbb{1} \,. \tag{2.19}$$

Similarly, $\mathcal{F}$ must be Hermitian-preserving because the the $\mathcal{E}_i$ are Hermitian-preserving.
For the reverse side, we make a dimensional argument similar to the one in Section 2.4. In the case of $n = 1$ the real vector space of Hermitian-preserving maps is 16-dimensional. Additionally asking for proportionality to a trace-preserving map is equivalent to restricting ourselves to 3 real linear constraints (to see that, consider how the TP constraint manifests in Pauli transfer matrices, see [15] for details), so the space of Hermitian-preserving trace-preserving maps is 13-dimensional. If we find 13 TPCP maps which are linearly independent, then we therefore know that they form a valid decomposition basis for any such $\mathcal{F}$. We already know 10 such maps from the Endo basis, namely $[\mathbb{1}], [\sigma^X], [\sigma^Y], [\sigma^Z], [R_X], [R_Y], [R_Z]$ (see Table 2.1). Following three maps also fulfill the required properties: $[\pi_X] + [\pi_{YZ}], [\pi_Y] + [\pi_{ZX}], [\pi_Z] + [\pi_{XY}]$. Their linear independence follows

[9] A feasible solution of a optimization problem is a solution which fulfills the constraints.

from the linear independence of the Endo basis. One can easily check that they are also TP and CP.
A similar argument can be performed in the cases $n = 2, 3$: The set of Hermitian-preserving maps forms a space of dimension 16^n and the set of Hermitian-preserving maps proportional to a TP map forms a space of dimension $16^n - (4^n - 1)$. We have to argue that there exist $16^n - 4^n + 1$ linearly independent TPCP maps. We do not explicitly write down the entire list of such TPCP map, as it would be enormous. Instead we describe how such a list can be constructed numerically. One generates $>> 16^n$ Haar-random quantum channels and checks how many of them are linearly independent.[10] We observed that we obtain exactly $16^n - (4^n - 1)$ linearly independent TPCP maps in this manner. □

Remark 2.9. The statement of Lemma 2.8 could be generalized to arbitrary n, if the following conjecture were proven:
The maximal number of linearly-independent n-qubit TPCP maps is exactly $16^n - 4^n + 1$.
Unfortunately we were not able to derive a proof for this statement for arbitrary n, so we formulate Lemma 2.8 and Theorem 2.10 only for $n = 1, 2, 3$. This will not lead to a practical limitation for the Stinespring algorithm in Chapter 4, as we will restrict ourselves to the study of 1-qubit and 2-qubit channels.

There is another obstacle before we can try to solve problem (2.13) numerically. It contains quadratic terms in the constraint $\Lambda_{\mathcal{F}} = \sum_i a_i \Lambda_{\mathcal{E}_i}$. Problems with convex objectives and quadratic constraints are generally NP-hard to solve. Luckily, using the correct substitution we can obtain a problem with a linear constraint instead. First we add a distinction between positive and negative quasiprobability coefficients, as denoted in Equation (2.14), with $a_i^{\pm} \geq 0, \tilde{N} \geq \lceil N/2 \rceil$. Then we define $\tilde{\Lambda}_i^{\pm} := a_i^{\pm} \Lambda_{\mathcal{E}_i}^{\pm}$. In this way we can reformulate the optimization problem in Equation (2.13) as

$$\begin{cases} \min\limits_{a_i^{\pm} \in \mathbb{R}^{\geq 0}, \tilde{\Lambda}_{\mathcal{E}_i}^{\pm} \in \mathbb{C}^{4^n \times 4^n}} & \sum\limits_{j=1}^{\tilde{N}} a_j^+ + a_j^- \\ \text{s.t.} & \Lambda_{\mathcal{F}} = \sum\limits_{j=1}^{\tilde{N}} \tilde{\Lambda}_{\mathcal{E}_j}^+ - \sum\limits_{j=1}^{\tilde{N}} \tilde{\Lambda}_{\mathcal{E}_j}^- \\ & \tilde{\Lambda}_{\mathcal{E}_i}^{\pm} \geq 0 \\ & \mathrm{Tr}_2[\tilde{\Lambda}_{\mathcal{E}_i}^{\pm}] = a_i^{\pm} \frac{1}{2^n} \mathbb{1} \,. \end{cases} \tag{2.20}$$

The optimization problem (2.20) is a SDP. In the mathematical optimization literature there are known algorithms that can solve such SDPs efficiently.

[10] One can numerically find the maximal number of linearly independent elements in a collection of vectors by stacking them into a matrix, computing the singular value decomposition and counting the number of nonzero singular values.

We can summarize the results of this section into following theorem:

Theorem 2.10 (Channel Difference Decomposition)**.** *Consider a n-qubit space A for $n \leq 3$ and let $\mathcal{F} \in \mathrm{L}(\mathrm{L}(A))$ be a Hermitian-preserving map proportional to a trace-preserving map. Then there exist $\mathcal{E}^+, \mathcal{E}^- \in TPCP(A, A)$ and $a^+, a^- \geq 0$ such that*

$$\mathcal{F} = a^+\mathcal{E}^+ - a^-\mathcal{E}^- , \tag{2.21}$$

where the C-factor of this QPD $a^+ + a^-$ is optimal over all possible QPDs in TPCP maps of the form of Equation (2.13). We call this decomposition the channel difference decomposition (CDD). The CDD can be expressed as a SDP.

In a certain sense one could see the C-factor $a^+ + a^-$ of the CDD as a measure of the non-physicality of a certain Hermitian-preserving TP map $\mathcal{F}$. If $\mathcal{F}$ is TPCP, then $a^+ + a^- = 1$, and the value becomes larger the more 'unphysical' the map gets. The operational interpretation of this measure is that it corresponds to the C-factor necessary to simulate a certain operation on an ideal quantum computer, which could implement any TPCP map perfectly.

2.7 Numerical Demonstration

To end this chapter, we perform a small numerical demonstration by computing the QPD of a few quantum gates using the Endo basis and some realistic assumptions on how we model the hardware noise. We denote the ideal unitary corresponding to the gate as U. When one tries to execute this gate on a noisy quantum computer, one actually implements the noisy quantum channel $\mathcal{A}$ that approximates $[U]$.

We will demonstrate the quasiprobability decomposition on three choices of U: a single-qubit *Ry* rotation with angle[11] $2 \arccos \sqrt{0.56789}$ and the two-qubit gates *CNOT* and *SWAP*. The noisy channels $\mathcal{A}$ of these three gates, as well as the operations in the noisy Endo basis, are obtained from a noise model included in Qiskit [21], which was generated from calibration measurements of the IBMQ Melbourne hardware backend. While these noise models do not reproduce the noise present on the real hardware in complete accuracy, they give us a good ballpark approximation of it. Table 2.2 depicts how well $\mathcal{A}$ approximates $[U]$ in terms of the process fidelity and the diamond norm.

In literature on QEM there are generally two variants how the correction of a gate might be performed using the quasiprobability method. The first possibility corresponds to the approach how the quasiprobability method was introduced in Section 2.1, namely by finding a QPD of the ideal gate operation $\mathcal{F} = [U]$. Endo *et al.* [9] coined the term *compensation method* for

[11] The exact angle is of no further importance.

	Ry	*CNOT*	*SWAP*
Fidelity $F([U], \mathcal{A})$	0.9973	0.9746	0.9276
Diamond norm $\|[U] - \mathcal{A}\|_\diamond$	0.0054	0.0528	0.1472

Table 2.2: Accuracy of the *Ry*, *CNOT* and *SWAP* gates from the Qiskit noise model of IBMQ Melbourne. The accuracy is given in terms of the process fidelity and the diamond norm.

this variant. Another option is to find a (generally non-physical) map $\mathcal{I}$ such that $\mathcal{I} \circ \mathcal{A} = [U]$, i.e. this map plays the role of an inverse map to the noise. The noisy gate $\mathcal{A}$ is executed as-is and the inverse map, simulated using the quasiprobability method, is implemented right afterwards. This method was coined *inverse method* by Endo *et al.*.

Table 2.3 depicts the numerical results for the compensation method and the inverse method, as well as the C-factor obtained from the CDD. The optimization problems are solved using the CVXPY package [22, 23]. MOSEK [24] is used as solver for the SDP of the CDD. Note that the compensation method works significantly better when the noisy operation $\mathcal{A}$ is included in the decomposition set. This makes sense intuitively - $\mathcal{A}$ is already a good approximation of $[U]$ and the remaining Endo basis only needs to take care of roughly $[U] - \mathcal{A}$. Because of a similar argument, the inverse method works already well with only the Endo basis.

In all cases the results are far away from the best theoretically achievable C-factor, which is given by the CDD. This motivates our search for a decomposition set which could hopefully be better than the Endo set, in order to bring the C-factor closer to the theoretical best. In Chapter 4 we will present an algorithm that does exactly that.

Ry gate	
	C-factor
compensation method, Endo basis	2.9892
compensation method, Endo basis $\cup\{\mathcal{A}\}$	1.0106
compensation method, theoretical best from CDD	1.0000
inverse method, Endo basis	1.0096
inverse method, Endo basis $\cup\{\mathcal{A}\}$	1.0096
inverse method, theoretical best from CDD	1.0054
CNOT gate	
	C-factor
compensation method, Endo basis	9.0564
compensation method, Endo basis $\cup\{\mathcal{A}\}$	1.1789
compensation method, theoretical best from CDD	1.0000
inverse method, Endo basis	1.1935
inverse method, Endo basis $\cup\{\mathcal{A}\}$	1.1935
inverse method, theoretical best from CDD	1.0618
SWAP gate	
	C-factor
compensation method, Endo basis	34.1381
compensation method, Endo basis $\cup\{\mathcal{A}\}$	2.2095
compensation method, theoretical best from CDD	1.0000
inverse method, Endo basis	1.4284
inverse method, Endo basis $\cup\{\mathcal{A}\}$	1.4284
inverse method, theoretical best from CDD	1.1779

Table 2.3: Numerical results from solving the linear program (2.11) to find the optimal QPD. The experiment is performed on three different gates: a *Ry* rotation, a *CNOT* gate and a *SWAP* gate. Both the compensation method and the inverse method are implemented. It is clearly visible that the noisy operation $\mathcal{A}$ should be included in the decomposition basis when the compensation method is used. The noise channels of the quantum operations involved in the numerical computations are obtained from a noise model in Qiskit which roughly approximates the noise present on the IBMQ Melbourne backend.

Chapter 3

Approximate Quasiprobability Decomposition

The sampling overhead of the quasiprobability method presents a major hurdle in the practical realization of the technique. Due to the exponential scaling of the C-factor in the number of corrected gates, one is restricted to shallow circuits in practice.

In this chapter, we present a relaxation of the QPD into a more general form, where we allow for a certain error in the decomposition, i.e.,

$$\mathcal{F}(\rho) \approx \sum_i a_i \mathcal{E}_i(\rho)\,, \tag{3.1}$$

which motivates the term approximate QPD. While we give up the exactness of the quasiprobability method, this generalization allows us to find a better decomposition in terms of the C-factor. The hope is that one would be able to significantly reduce the C-factor while only paying with a very small error in the decomposition. The situation at hand is therefore a tradeoff between the sampling overhead and the error in the method.

3.1 SDP Relaxation using the Diamond Norm

In order to flesh out the aforementioned intuition in a rigorous manner, we first have to decide how to quantify the error in the approximate QPD. A natural candidate is to use the diamond distance, i.e., the distance induced by the diamond norm, as it has a strong operational interpretation. In fact the diamond norm fits very naturally in our mathematical optimization setting, as it has been shown to be expressible as a semidefinite program (SDP) [25]:

Theorem 3.1 (SDP for diamond norm [25]). *Let* $\mathcal{G} \in \mathrm{L}(\mathrm{L}(A), \mathrm{L}(B))$ *and denote*

its Choi matrix representation by $\Lambda_{\mathcal{G}}$. Then

$$\|\mathcal{G}\|_\diamond = \begin{cases} \max\limits_{\substack{\rho_0,\rho_1\in \mathrm{L}(A)\\ X\in \mathrm{L}(BA)}} & \frac{1}{2}\langle \Lambda_{\mathcal{G}}, X\rangle + \frac{1}{2}\langle \Lambda_{\mathcal{G}}^*, X^*\rangle \\ \text{s.t.} & \begin{pmatrix} \mathbb{1}_B \otimes \rho_0 & X \\ X^* & \mathbb{1}_B \otimes \rho_1 \end{pmatrix} \geq 0 \\ & \rho_0 \geq 0, \rho_1 \geq 0 \\ & \rho_0^\dagger = \rho_0, \rho_1^\dagger = \rho_1\,, \end{cases} \tag{3.2}$$

where $\langle \cdot, \cdot \rangle$ is the trace inner product. (3.2) is a SDP.

The dual formulation of the SDP (3.2) is given by

$$\|\mathcal{G}\|_\diamond = \begin{cases} \min\limits_{Y_0,Y_1\in \mathrm{L}(BA)} & \frac{1}{2}\left\|\mathrm{Tr}_B[Y_0]\right\|_\infty + \frac{1}{2}\left\|\mathrm{Tr}_B[Y_1]\right\|_\infty \\ \text{s.t.} & \begin{pmatrix} Y_0 & -\Lambda_{\mathcal{G}} \\ -\Lambda_{\mathcal{G}}^* & Y_1 \end{pmatrix} \geq 0 \\ & Y_0 \geq 0, Y_1 \geq 0\,, \end{cases} \tag{3.3}$$

where $\|\cdot\|_\infty$ denotes the spectral norm of a matrix.

Let's consider a setting similar to Section 2.5: We have a fixed decomposition set $\{\mathcal{E}_i\}_i$ and we would like to find the best possible approximate QPD of an operation $\mathcal{F} \in \mathrm{L}(\mathrm{L}(A), \mathrm{L}(B))$ into that decomposition set. More precisely, we give a certain limit of the C-factor, denoted C_{budget}, such that the QPD may not have a C-factor higher than this limit. The optimization problem at hand is thus

$$\begin{cases} \min\limits_{a_i\in\mathbb{R}} & \left\|\mathcal{F} - \sum_j a_j \mathcal{E}_j\right\|_\diamond \\ \text{s.t.} & \sum\limits_j |a_j| \leq C_{\text{budget}}\,. \end{cases} \tag{3.4}$$

By inserting the SDP from Equation (3.3) into Equation (3.4) the problem can be rewritten as

$$\begin{cases} \min\limits_{a_i\in\mathbb{R}, Y_0,Y_1\in \mathrm{L}(AB)} & \frac{1}{2}\left\|\mathrm{Tr}_B[Y_0]\right\|_\infty + \frac{1}{2}\left\|\mathrm{Tr}_B[Y_1]\right\|_\infty \\ \text{s.t.} & \begin{pmatrix} Y_0 & \sum_j a_j \Lambda_{\mathcal{E}_j} - \Lambda_{\mathcal{F}} \\ \sum_j a_j \Lambda_{\mathcal{E}_j}^* - \Lambda_{\mathcal{F}}^* & Y_1 \end{pmatrix} \geq 0 \\ & \sum_j |a_j| \leq C_{\text{budget}} \\ & Y_0 \geq 0, Y_1 \geq 0\,. \end{cases} \tag{3.5}$$

where $\Lambda_{\mathcal{F}}, \Lambda_{\mathcal{E}_i}$ are the Choi matrices of $\mathcal{F}, \mathcal{E}_i$. Luckily Equation (3.5) is still a SDP, which allows us to efficiently solve our optimization problem.

Remark 3.2 (Non-physicality of the approximate decomposition)**.** The approximation $\mathcal{F}_{\text{approx}} := \sum_i a_i \mathcal{E}_i$ of $\mathcal{F}$ obtained from the optimization problem in Equation (3.4) is not guaranteed to be a physical map.[1] In practice, this implies that we simulate the execution of a non-physical map using the quasiprobability method, even if $\mathcal{F}$ is itself physical. It is possible to enforce complete positivity and/or trace-preservingness into the optimization problem

$$\begin{cases} \min\limits_{a_i \in \mathbb{R}} & \left\| \mathcal{F} - \sum_j a_j \mathcal{E}_j \right\|_\diamond \\ \text{s.t.} & \sum\limits_j |a_j| \leq C_{\text{budget}} \\ & \sum\limits_j a_j \mathcal{E}_i \text{ is TPCP}\,, \end{cases} \tag{3.6}$$

which translates into additional constraints in the SDP (3.5):

$$\begin{cases} \min\limits_{a_i \in \mathbb{R}, Y_0, Y_1 \in \mathrm{L}(BA)} & \frac{1}{2} \left\| \mathrm{Tr}_B[Y_0] \right\|_\infty + \frac{1}{2} \left\| \mathrm{Tr}_B[Y_1] \right\|_\infty \\ \text{s.t.} & \begin{pmatrix} Y_0 & \sum_j a_j \Lambda_{\mathcal{E}_j} - \Lambda_{\mathcal{F}} \\ \sum_j a_j \Lambda^*_{\mathcal{E}_j} - \Lambda^*_{\mathcal{F}} & Y_1 \end{pmatrix} \geq 0 \\ & \sum_j |a_j| \leq C_{\text{budget}} \\ & Y_0 \geq 0, Y_1 \geq 0 \\ & \sum_j a_j \Lambda_{\mathcal{E}_j} \geq 0 \\ & \mathrm{Tr}_B[\sum_j a_j \Lambda_{\mathcal{E}_j}] = \frac{1}{2^n} \mathbb{1}\,. \end{cases} \tag{3.7}$$

Of course, adding an additional constraint into the problem will lead to an equal or worse C-factor. Intuitively, one would expect this drawback not to be significant, as long as the approximation is not too coarse, since a good approximation of $\mathcal{F}$ is going to be 'almost' a TPCP map. This intuition seems to hold at least for the three gates *Ry*, *CNOT* and *SWAP* discussed previously. We observed that their tradeoff curves remain almost identical under the inclusion of the CP constraint. In some situations it can be very useful to have some complete positivity guarantee on $\mathcal{F}_{\text{approx}}$, as it allows us to use known results and tools from quantum information theory. An example would be the linear propagation of errors (in terms of the diamond norm) throughout the circuit.

3.2 Tradeoff Curves

In this section, we present some numerical results of the approximate QPD by visualizing the result of the SDP (3.5). More precisely, we consider the same three quantum gates *Ry*, *CNOT* and *SWAP* as in Section 2.7 with the same noise model. We decompose the gates using the compensation method

[1] Recall that a map is physical if it is trace-preserving and completely positive.

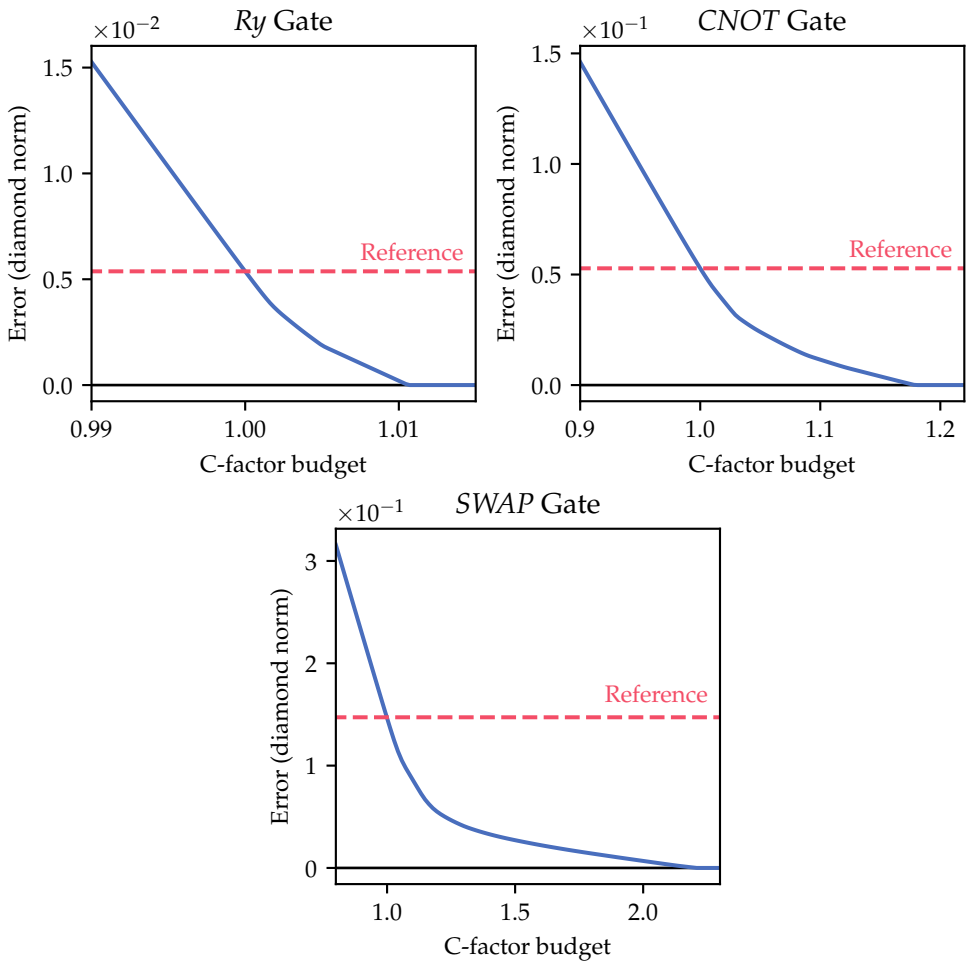


Figure 3.1: Tradeoff curves for the three quantum gates $Ry(2\arccos\sqrt{0.56789})$, *CNOT* and *SWAP* using the compensation method and a decomposition set consisting of the Endo basis and the respective noisy variant of the gate. The noisy channels of the three gates and the Endo basis are extracted from a noise model in Qiskit [21] which approximates the noise on the IBMQ Melbourne hardware backend. The red dashed line represents the error (in diamond norm) of the reference noisy channel, i.e. when the gate is implemented as-is without QEM.

into the decomposition set composed of the Endo basis and the respective noisy variant of the gate. We solve the SDP (3.5) for different values of C_{budget} to obtain a relation $\epsilon(C_{\text{budget}})$ between the diamond distance error and the C-factor. This function, which we call *tradeoff curve*, encapsulates the tradeoff between the approximation error and the sampling overhead. The tradeoff curves for *Ry*, *CNOT* and *SWAP* were numerically obtained using the SDP solver in the MOSEK [24] software package through the CVXPY [22, 23] modelling language. The results are depicted in Figure 3.1.

As expected, if the C-factor budget is larger than the optimal C-factor of the

non-approximate QPD (which can be found in Table 2.3), the error becomes zero. Similarly, when the C-factor budget is exactly 1, then one doesn't gain any advantage over implementing the gate as-is without QEM. The more interesting regime is in between these two values of C_{budget}: One clearly sees that we can still significantly reduce the error of a gate, without having to pay the full C-factor necessary for a non-approximate QPD. For the *SWAP* gate, the exact QPD requires a C-factor of 2.21 to completely correct the gate. However, if we only pay a C-factor of 1.21, we can still reduce the error by 64%. The saved costs due to the lowered C-factor requirement are substantial, since the number of shots scales as $C^{2 \cdot n_{gates}}$, as seen in Section 2.2.

Interestingly the tradeoff curves seem to resemble pairwise linear functions, i.e. $\epsilon(C_{\text{budget}})$ seems to transition between regimes where the function is roughly linear. This is especially visible for the *Ry* and *CNOT* gates. We would like to emphasize that this is not a result of using a low amount of sampling points, but rather an inherent feature of the curves themselves.

3.3 Application: Optimal Resource Distribution

If one is to apply the approximate quasiprobability method to a circuit with multiple gates, a new degree of freedom emerges, which is not present in the original formulation of the quasiprobability method: How much C-factor budget do we give to every individual gate? Lets assume we have a budget C_{total} for the whole circuit, how do we distribute that budget optimally across the whole circuit? More concretely, given N gates we have to find individual budgets $C_{\text{budget},i} \geq 0$ for $i = 1, \ldots, N$ such that $\prod_{i=1}^{N} C_{\text{budget},i} = C_{\text{total}}$.

One might expect that splitting the budget equally (i.e. $C_{\text{budget},i} = C_{\text{total}}^{1/N}$) is the optimal solution. As we will see in an example at the end of the section, this is in fact not the case. The optimal distribution seems to be a non-trivial optimization problem that has to be solved numerically.

Before we continue, we have to clarify what objective function we seek to optimize when we talk about an 'optimal' budget distribution. The overall goal is to minimize the error (in terms of the diamond norm) of the total circuit. Let's consider a setting where we execute N quantum gates $U_1, \ldots, U_N$ acting on n qubits each. We denote by $\mathcal{F}_{\text{approx},i}$ the map produced by the approximate QPD of the i-th gate $[U_i]$. The objective function is therefore

$$\left\| \mathcal{F}_{\text{approx},N} \circ \cdots \circ \mathcal{F}_{\text{approx},2} \circ \mathcal{F}_{\text{approx},1} - [U_N] \circ \cdots \circ [U_2] \circ [U_1] \right\|_{\diamond} . \tag{3.8}$$

The computation of this quantity is intractable, as it would require to simulate the complete noisy circuit. Therefore, we need to find a proxy that is computable from local quantities.

From the triangle inequality and the fact that the diamond norm is a contraction under completely positive maps, one immediately obtains

$$\begin{aligned}\left\|\mathcal{E}\circ\mathcal{F}-\mathcal{E}'\circ\mathcal{F}'\right\|_\diamond &\leq \left\|\mathcal{E}\circ\mathcal{F}-\mathcal{E}'\circ\mathcal{F}\right\|_\diamond + \left\|\mathcal{E}'\circ\mathcal{F}-\mathcal{E}'\circ\mathcal{F}'\right\|_\diamond \\ &\leq \left\|\mathcal{E}-\mathcal{E}'\right\|_\diamond + \left\|\mathcal{F}-\mathcal{F}'\right\|_\diamond ,\end{aligned}$$

for any completely positive maps $\mathcal{E},\mathcal{F},\mathcal{E}',\mathcal{F}'$ where $\mathcal{E},\mathcal{E}'$ are trace-preserving. If we wish to apply this bound to our setting in Equation (3.8), we have to guarantee that the $\mathcal{F}_{\text{approx},i}$ are TPCP. This can be achieved by inserting an additional constraint into the SDP of the approximate QPD, as described in Remark 3.2. Using the notation $\epsilon_i(C)$ to denote the tradeoff curve of the i-th gate, we thus obtain the optimization problem

$$\begin{cases} \min\limits_{C_{\text{budget},i}\in\mathbb{R}} & \sum_j \epsilon_j(C_{\text{budget},j}) \\ \text{s.t.} & C_{\text{budget},i}\geq 0\,\forall i \\ & \prod_j C_{\text{budget},j} = C_{\text{global}}\,. \end{cases} \tag{3.9}$$

The problem (3.9) is generally non-convex, as we have not enough guarantees on the shape of the tradeoff curves. Still in practice we observe that the objective is convex in a broad region around the optimum, so we expect numerical heuristics based on gradient descent to work well.

We finish this section with a small demonstration on a simple circuit that consists of a *Ry* and a *CNOT* gate. The tradeoff curves are computed exactly as in Section 3.2, with the sole difference that we now include a TPCP-constraint for the approximate QPD. We evaluate the tradeoff curves at discrete points and use linear interpolation in order to extrapolate this data to a full function that can be used in the optimization routine. We solve the optimization problem (3.9) for different values of C_{total} using a black box solver based on the BFGS algorithm implemented in the SciPy [26] software package. The results are depicted in Figure 3.2. Interestingly, the optimal strategy for budget distribution is nontrivial. In the regime with a small budget $1 \leq C_{\text{total}} \lessapprox 1.02$, it is optimal to give most of the C-factor budget solely to the CNOT gate. In a transition regime $1.02 \lessapprox C_{\text{total}} \lessapprox 1.04$ both gates obtain a significant amount of C-factor. In the upper regime $1.04 \lessapprox C_{\text{total}}$ the *Ry* gate is perfectly decomposed and the remaining budget is then given to the *CNOT* gate. So whether one should prioritize the *CNOT* or the *Ry* gate in the budgeting strongly depends on the value of C_{total}.

3.4 Application: Unitary-Only Decomposition

Implementing the quasiprobability method on currently available quantum hardware is difficult for various reasons. One major roadblock is that realising a basis set able to decompose any unitary operation, such as the Endo

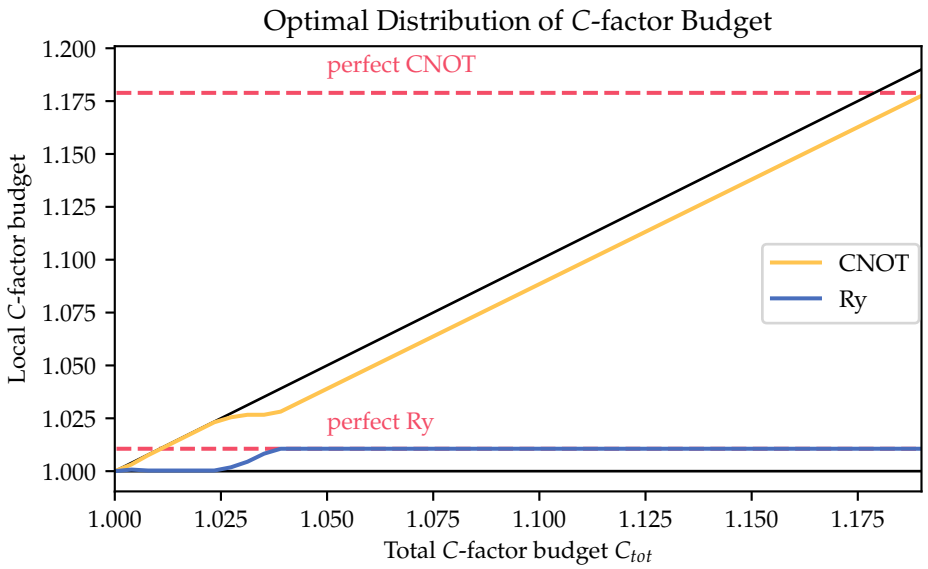


Figure 3.2: This curve shows the optimal C-factor budget distribution across a circuit consisting of a single *Ry* and *CNOT* gate. The total C-factor budget C_{total} is varied between 1 and 1.191, which is the minimal value necessary to obtain perfect QPD for both gates at the same time. The red dashed lines denote the necessary C-factor to decompose the respective gate perfectly. The black line denotes C_{total} and therefore corresponds to the multiplication of the blue and yellow curves.

basis, requires some operations with measurements and/or resets. For example, out of the 16 operations in the Endo basis (see Table 2.1) only 10 can be implemented by using purely unitary operations. Measurements at intermediate locations inside a circuit (i.e. not at the very end) are rarely implemented on current hardware platforms due to technical limitations. Furthermore, current measurements are typically plagued by prohibitively high noise. They also require orders of magnitude more time than unitary gates, causing significant decoherence in the surrounding qubits which have to wait for the measurement to finish. Therefore it is a natural question to ask, whether some form of the quasiprobability method could be implemented on a computer only having access to unitary operations.

With the approximate QPD we can perform precisely this task. Instead of decomposing a target operation $\mathcal{F}$ into the full Endo basis, we try to find the best possible decomposition of $\mathcal{F}$ into a restricted decomposition set that only contains unitary operations. The consequence of this approach is that we cannot reach a perfect QPD, even with an infinitely high C-factor budget.

In Figure 3.3 the tradeoff curves for the three previously considered gates *Ry*, *CNOT* and *SWAP* (see Section 3.2) are depicted. The decomposition set consists of the 10 unitary operations in the Endo basis (or tensor products

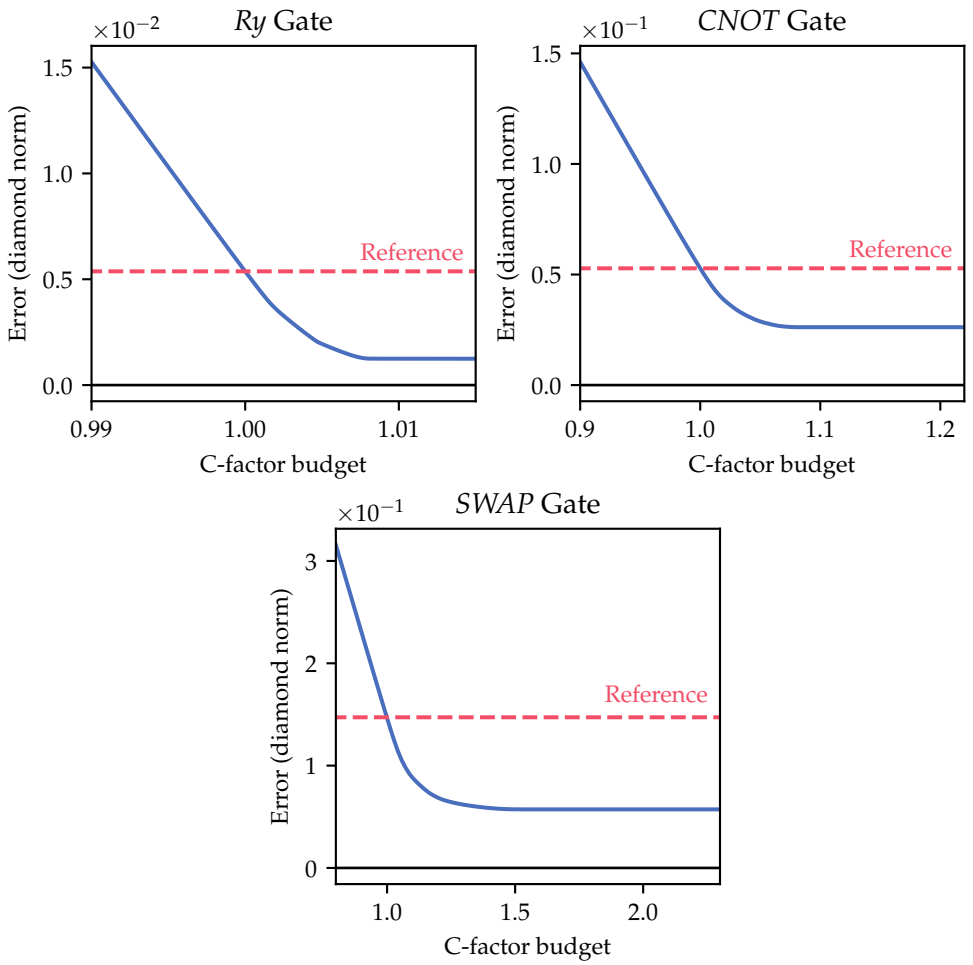
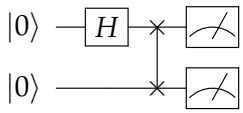


Figure 3.3: Tradeoff curves for the three quantum gates $Ry(2\arccos\sqrt{0.56789})$, *CNOT* and *SWAP* using a unitary-only quasiprobability decomposition. The only difference to Figure 3.1 is that the decomposition set is reduced to only containing the unitary operations in the Endo basis and the respective noisy variant of the gate.

thereof) as well as the noisy variant of the respective gate. The compensation method is used to correct the gates. It can be seen that a significant reduction of the error is still achievable, though the magnitude of this improvements depends on the specific noise present in the hardware.

In the last part of this section we present experimental results from an implementation of the unitary-only approximate quasiprobability method on the IBMQ Singapore quantum chip using Qiskit [21]. The task of the experiment is to correct following simple quantum circuit:



Since the noise of two-qubit gates is significantly stronger compared to single-qubit gates, we will onlly correct the *SWAP* gate with the quasiprobability method.

We wish to perform an approximate QPD as in Equation (3.1) where the 101-element decomposition set $\{\mathcal{E}_i\}$ consists of $10 \cdot 10$ unitary Endo operations and the noisy SWAP gate $\mathcal{A}$. Since the Endo operations consist of tensor products of single-qubit unitary gates, we can assume them to be ideal in a first order approximation. Therefore we only need to perform tomography of the channel $\mathcal{A}$. This is done using process tomography, which entails the execution of 144 different quantum circuits (see [15] for more details) which are each executed with 8192 shots. In a second step we can perform the approximate QPD $[U] \approx \sum_i a_i \mathcal{E}_i$ with infinite C-factor budget. Next we generate 101 circuits, each one corresponding to the original circuit with the *SWAP* gate replaced by an element $\mathcal{E}_i$ of the decomposition set. Each of these 101 circuits is executed on the hardware with 8192 shots, resulting in an estimate $\hat{p}_i$ of the distribution of the measurement outcomes of the respective circuit. Denote by p the distribution of the measurement outcome of the ideal circuit which contains an ideal SWAP gate. To evaluate the improvement caused by the quasiprobability method, we consider the L_1-norm between p and $\sum_i a_i \hat{p}_i$ (note that the latter is generally not a real distribution, since $\sum_i a_i \mathcal{E}_i$ doesn't have to be TP or CP. So we have to think of this distance as a L_1-norm between arbitrary vectors). This experimental procedure is repeated 5 times and the results are reported in Table 3.1. The qubits used on the hardware platform were the first two (labelled by indices 0 and 1). On average the error of the statistics is reduced by 23%. We also repeated the same experiment a second time on the two qubits that had the worst 2-qubit gate fidelities, with the results depicted in Table 3.2. In that case we obtained an average improvement of 34%.

	L_1 error without mitigation	L_1 error with mitigation
Run 1	0.053	0.034
Run 2	0.056	0.043
Run 3	0.058	0.049
Run 4	0.061	0.043
Run 5	0.066	0.056
Mean	0.058	0.045
Stdev	0.005	0.008

Table 3.1: Improvements of the unitary-only quasiprobability method on the IBMQ Singapore hardware backend. A single SWAP gate on the qubits 0 and 1 is corrected using the compensation method.

	L_1 error without mitigation	L_1 error with mitigation
Run 1	0.261	0.167
Run 2	0.298	0.205
Mean	0.279	0.186
Stdev	0.026	0.026

Table 3.2: Improvements of the unitary-only quasiprobability method on the IBMQ Singapore hardware backend. A single SWAP gate on the qubits 12 and 13 is corrected using the compensation method.

Chapter 4

Stinespring Algorithm

Up to now, we have introduced two variants how to obtain an optimal QPD: If the decomposition set is fixed in advance, it can be found by a simple linear program (see Section 2.5) and if we also optimize over the decomposition set, it is given by the CDD (see Section 2.6). Unfortunately, the result of the CDD is not directly applicable to the quasiprobability method, as we cannot implement an arbitrary quantum channel on our quantum hardware. Therefore there is a need for a novel method that is able to find a decomposition set which can perform better than the Endo basis, but is implementable on the quantum computer at hand. This algorithm must take into account the noise of the specific quantum quantum computer, such that it can adapt the decomposition set accordingly. This task is hard for two reasons:

1. It is difficult to optimize over the space of all possible operations that a given quantum hardware can perform. In general this space is large[1] and might vary significantly from one machine to another.

2. Performing tomography to characterize the operations $\mathcal{E}_i$ is expensive. Hence, we want to limit the number of required uses of tomography.

In this chapter, we will introduce a novel method, called the *Stinespring algorithm*, that is able to deal with both of these issues. The core idea, which addresses the first issue, is to construct our decomposition set from a limited class of operations of a specific form, which can be implemented on any reasonable quantum computer. More precisely, we will consider noisy operations which approximate an ideal quantum channel using a Stinespring dilation.

[1] The space of all possible operations doesn't just consist of individual gates. It also comprises operations that use multiple gates, introduce ancilla qubits, perform measurements, trace out a subset of the qubits, etc.... Basically any quantum circuit that starts and ends with the correct amount of qubits can be seen as a possible operation to be used in the decomposition set.

The Stinespring algorithm is able to produce a decomposition set which exhibits a significantly reduced C-factor compared to the Endo basis. Furthermore, the Stinespring algorithm comes at an additional advantage as it doesn't require measurements on the computational qubits — it only requires resets on the ancilla qubits used for the dilation. However, the need of additional ancilla qubits is a drawback that is not present when using the Endo basis.

Before presenting the full Stinespring algorithm in section 4.3, we will first introduce some essential building blocks which will prove useful later.

4.1 Stinespring Dilation on a Quantum Computer

The Stinespring dilation theorem is a fundamental result in quantum information theory. It states that any quantum channel can be expressed as a unitary evolution on some extended Hilbert space, where we don't have access to the quantum information stored in the extension.

Theorem 4.1 (Stinespring dilation)**.** *Consider $\mathcal{E} \in TPCP(A, A)$. There exists a Hilbert space $\mathcal{H}_R$ and an isometry $V_{St} \in \mathrm{L}(A, AR)$ such that*

$$\forall \rho \in \mathcal{S}(A) : \mathcal{E}(\rho) = \mathrm{Tr}_R[V_{St} \rho V_{St}^\dagger] \,. \tag{4.1}$$

Furthermore, there exists such a Stinespring dilation with $dim(R) \leq r$ where r is the rank of the quantum channel defined by $r := rank(\Lambda_\mathcal{E})$ for $\Lambda_\mathcal{E}$ the Choi matrix of $\mathcal{E}$.

Any isometry V_{St} can be extended (generally non-uniquely) to a unitary $U_{St} \in \mathrm{U}(AR, AR)$ such that U_{St} acts equivalently to V_{St} on the space of states of the form $\rho_A \otimes |0\rangle\langle 0|_R$ for some arbitrarily chosen state $|0\rangle_R$. Numerically it is not difficult to obtain the isometry V_{St} for an arbitrary channel. For the exact details we refer the reader to [27].

We can straightforwardly use the Stinespring dilation to approximate an arbitrary quantum channel on a quantum computer by making use of clean ancilla qubits, performing a circuit corresponding to the unitary U_{St} on the extended space, and finally discarding the ancilla qubits. However, we cannot expect a noisy quantum computer to implement U_{St} accurately, so the resulting channel will only be an approximation of the target channel that we want to achieve.

Current quantum hardware especially struggles to produce high-fidelity two-qubit gates, so our approximation will be poor if the circuit implementation of U_{St} requires too many of them. For simplicity we assume that the *CNOT* gate is the only multi-qubit gate that our hardware platform can implement, and that all other multi-qubit gates must be decomposed into

CNOT gates and single-qubit gates. So the number of *CNOT* gates is often a good indicator for the amount of noise accumulated by a quantum circuit. The number of required *CNOT* gates to implement U_{St} is generally related to the number of qubits involved in the unitary, so we want to keep the rank r as small as possible ($r = 1$ requires no ancilla qubits, $r = 2$ requires one ancilla qubit, $r = 3, 4$ requires two ancilla qubits, etc...).

In practice we will consider 1-qubit and 2-qubit channels of rank $r \leq 2$, such that the dilation can be implemented with at most a 3-qubit unitary. In our experience this is the upper limit where the channel still gets reasonably approximated with current quantum hardware.

In the following subsection we introduce a method which allows us to further improve the channel approximation via a Stinespring dilation.

4.1.1 Variational Unitary Approximation

To optimize the approximation of a quantum channel through the Stinespring dilation, we can make use of two central observations:

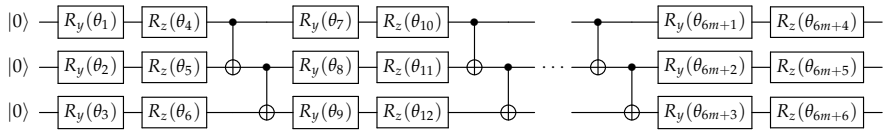
1. There is some additional degree of freedom that we can make use of to further optimize our approximation, because the choice of the unitary U_{St} extending the isometry V_{St} is not unique. Hence one could try to choose the U_{St} which requires the least amount of *CNOT* gates to implement.

2. Since the hardware is noisy, it may not be advantageous to implement a circuit that represents U_{St} exactly. It might make more sense to implement a circuit that approximates U_{St} with a finite error that requires less *CNOT* gates and thus suffers less from noise on the hardware. There is a tradeoff between the approximation error and the error stemming from hardware noise.

In this section, we propose a method that makes use of both of these optimizations, which we call *variational unitary approximation*. We use a variational circuit to implement the dilation and fit its parameters in order to approximate V_{St} as well as possible. Denote by θ the tuple of all variational parameters and $U_{Var}(\theta)$ the unitary represented by the variational form. Furthermore we denote by $V(U_{Var}(\theta))$ the submatrix of $U_{Var}(\theta)$ restricted on the subspace where the input ancilla qubits are fixed to the zero state. We want to choose our parameters θ such that we minimize the difference between $V(U_{Var}(\theta))$ and V_{St}.

The variational unitary approximation allows us to freely choose the expressiveness of our variational circuit. More concretely, we can tune how many CNOT gates it shall contain and therefore influence the tradeoff between the approximation error and the hardware noise error.

In the following we are going to look at a concrete example to illustrate the gains from the variational unitary approximation. Consider an arbitrary 2-qubit quantum channel with rank 2 that we want to approximate with a Stinespring dilation. The exact representation of a general 3-qubit unitary as a quantum circuit typically requires around 100 CNOT gates (assuming linear connectivity), which is significantly more than what current hardware can reasonably handle. We replace this lengthy circuit with a *RYRZ* variational circuit



where m denotes the depth of the variational form, the total number of parameters is $6(m+1)$ and the the total number of CNOT gates is $2m$. We use the gradient-based BFGS algorithm implemented in SciPy [26] to minimize the objective $\|V_{St} - V(U_{Var}(\theta))\|_2$.[2] As initial guess for the parameters we use uniformly random numbers. The objective is non-convex and in practice the optimization algorithm finds a different local minimum depending on the provided initial guess. To obtain a good result in practice, we observed that it is enough to repeat the optimization 5 times for different initial values and then take best result.

The above procedure is performed for different depths of the variational form on a Haar-random 3-qubit unitary U_{St}. By using a Qiskit [21] noise model extracted from the IBMQ Melbourne hardware backend, we can estimate how well our method allows us to approximate the 2-qubit quantum channel $\rho \mapsto \mathrm{Tr}_3[U_{St}\rho U_{St}^\dagger]$ where Tr_3 stands for tracing out the third qubit. More precisely, we can compute the diamond norm error between the obtained 2-qubit quantum channel and the ideal 2-qubit quantum channel. This error encapsulates the approximation error of the variational form combined with the hardware noise. The results can be seen in Figure 4.1. The variational unitary approximation technique allows us to significantly reduce the error to roughly one quarter of its reference value. One can also clearly see that there is a sweet spot in the variational circuit depth. This makes sense intuitively when considering the tradeoff mentioned previously: if the depth is too short, then U_{St} is not well approximated and if the depth is too long, then the hardware noise takes over and we start to mostly sample noise. The optimum is reached at a variational circuit depth of 6. Furthermore one observes that the approximation error goes to zero as soon as the depth is at least 7.

[2] In order to avoid having to estimate the gradient with the finite differences method, we make use of the Autograd [28] package for automatic differentiation.

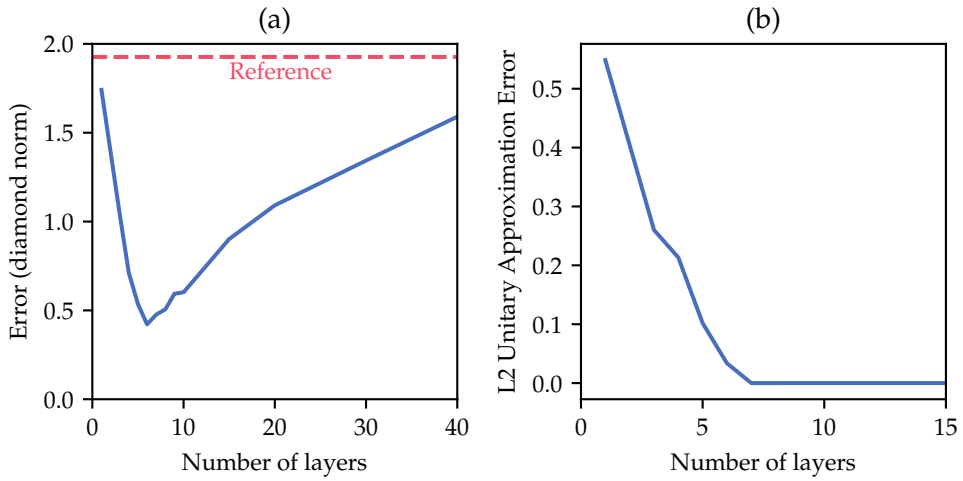


Figure 4.1: Numerical results from the variational unitary approximation of a Haar-random 3-qubit unitary using a *RYRZ* variational form. (a) depicts the diamond norm error of the induced 2-qubit quantum channel for different depths of the variational form. The dashed red line corresponds to the error obtained if U_{St} were naively decomposed in a quantum circuit without the variational unitary approximation. (b) depicts the approximation error $\|V_{St} - V(U_{Var}(\theta))\|_2$ and can thus be regarded as the analogue of (a) without the effect of hardware noise.

4.2 Rank-Constrained Channel Decomposition

Consider a map $\mathcal{F} \in TP(A, A)$. Remember that the CDD (see Section 2.6) asserts that we can find $\mathcal{E}^+, \mathcal{E}^- \in TPCP(A, A)$, $a^+, a^- \geq 0$ such that $\mathcal{F} = a^+\mathcal{E}^+ - a^-\mathcal{E}^-$ with optimal C-factor $a^+ + a^-$. For the sake of the Stinespring algorithm, we would like to approximate the $\mathcal{E}^\pm$ using a Stinespring dilation, as explained in Section 4.1. This will only work reasonably well when the number of required ancilla qubits is not too large. We can enforce this by adding an additional constraint $\text{rank}(\Lambda_{\mathcal{E}_i}^\pm) \leq r$ for some $r \in \mathbb{N}^{\geq 1}$ into Equation (2.20):

$$\begin{cases} \min\limits_{a_i^\pm \in \mathbb{R}^+, \tilde{\Lambda}_{\mathcal{E}_i}^\pm \in \mathbb{C}^{4^n \times 4^n}} & \sum\limits_{j=1}^{n_{pos}} a_j^+ + \sum\limits_{j=1}^{n_{neg}} a_j^- \\ \text{s.t.} & \Lambda_{\mathcal{F}} = \sum\limits_{j=1}^{n_{pos}} \tilde{\Lambda}_{\mathcal{E}_j}^+ - \sum\limits_{j=1}^{n_{neg}} \tilde{\Lambda}_{\mathcal{E}_j}^- \\ & \tilde{\Lambda}_{\mathcal{E}_i}^\pm \geq 0 \\ & \text{Tr}_2[\tilde{\Lambda}_{\mathcal{E}_i}^\pm] = a_i^\pm \frac{1}{2^n} \mathbb{1} \\ & \text{rank}(\tilde{\Lambda}_{\mathcal{E}_i}^\pm) \leq r\,. \end{cases} \tag{4.2}$$

Note that with this constraint we don't have the guarantee anymore that we can decompose $\mathcal{F}$ into just two channels. We denote by n_{pos}, n_{neg} the number of channels into which we perform the decomposition.

The introduction of such a rank constraint generally turns a SDP into a NP-hard problem, so one has to resort to heuristics to find a solution. One com-

mon heuristic for rank constrained SDPs is to minimize the nuclear norm instead of bounding the rank, as it constitutes the convex envelope of the rank. Unfortunately this doesn't work in our case, as the trace of the Choi matrices is already fixed. We instead opted to use the Burer-Monteiro approach for low-rank solution of SDPs [29, 30]. Suppose that we want to solve a given SDP with a rank constraint $\text{rank}(C) \leq r$ for some positive-semidefinite $n \times n$ matrix C. The main idea is to parametrize $C = X^\dagger \cdot X$ for some $r \times n$ complex matrix X and then optimize over the matrix elements of X. The positive-semidefiniteness and rank constraint of C are automatically enforced from the construction. Unfortunately, the objective and the constraints generally contain quadratic terms of X, so the problem becomes a quadratically-constrained quadratic problem. Still, recent research has demonstrated that the objective landscape of these problem tends to behave nicely and that using local optimization methods can provably lead to the global optimum under some assumptions [31, 32].

In our case the $\Lambda^{\pm}_{\mathcal{E}_i}$ don't actually show up in the objective function, so we actually have a quadratically-constrained linear problem, i.e. a problem of the form

$$\begin{cases} \min\limits_{x} & f(x) \\ \text{s.t.} & g(x) = 0\,, \end{cases} \tag{4.3}$$

where we grouped all variables into a large vector x, f is a linear function and g is a quadratic map that encapsulates all constraints. Furthermore, we know the optimum f^* of the objective function without the rank constraint. Let's make an assumption for the moment being:

Assumption 4.2. The rank-constrained optimization problem (4.2) reaches the same minimal C-factor as the original SDP in (2.20).

This assumption does obviously not hold for $r = 1$, as it would imply that we could decompose any $\mathcal{F} \in TP(A, A)$ into unitary operations. We are only concerned with the case $r \geq 2$ in practice, so let's assume for the moment that the assumption is correct. This allows us to reformulate Equation (4.3) as

$$\begin{cases} \min\limits_{x} & \|g(x)\|_2 \\ \text{s.t.} & f(x) = f^*\,. \end{cases} \tag{4.4}$$

We say that our optimization only succeeds if it finds a solution of problem (4.4) which minimizes $\|g(x)\|_2$ to zero. The reason why we do this switching around of objective and constraint, is because linear constraints tend to be easier to deal with numerically.

To numerically solve problem (4.4) we made use of the trust-region algorithm for constrained optimization which is included in SciPy [26]. We

fixed $r = 2$, which corresponds to allowing at most one ancilla qubit in the Stinespring dilation. In the single-qubit case[3] we could consistently find a good solution, whereas in the two-qubit case we were often confronted with convergence problems. It is not exactly clear whether these problem stemmed from the non-existence of a low-rank solution (i.e. Assumption 4.2 being wrong) or from numerical issues with the Burer-Monteiro method. In any case, we observed that the convergence could be significantly improved by allowing some error in the linear constraint, i.e.,

$$\begin{cases} \min\limits_{x} & ||g(x)||_2 \\ \text{s.t.} & f^* \leq f(x) \leq (1+\varepsilon) f^* \,, \end{cases} \tag{4.5}$$

where we used a value of $\varepsilon = 0.2$. In other words, we allow for a sub-optimality in the C-factor by at most 20%. In principle one could try to find the smallest admissible ε by usage of bisection. We used $n_{pos} = n_{neg} = 2$ for single-qubit quantum channels and $n_{pos} = n_{neg} = 8$ for two-qubit quantum channels. We also observed a good initialization of x for problem (4.4) to be of major importance. We defer the details how this initialisation procedure to Appendix A.

4.3 Overview of the Stinespring Algorithm

The result in Section 4.2 allows us to decompose any $\mathcal{F} \in TP(A, A)$ into rank r quantum channels. Using the result from Section 4.1 these channels can be approximated by using $\lceil \log_2 r \rceil$ ancilla qubits. By choosing r small enough, we can make sure that this approximation is not too bad.

Still, there is an important step missing before we can practically use this result. The quasiprobability method requires a QPD where the elements of the decomposition set $\mathcal{E}_i$ correspond to channels describing noisy operations which the actual quantum hardware can execute. However, the Stinespring dilation only yields us an approximation of such a QPD. Somehow we have to take into account the inaccuracy of the Stinespring dilation when constructing our QPD. This will be achieved by the use of an iterative algorithm.

Assume that we have access to a noise oracle $\mathcal{E} \mapsto \mathcal{N}(\mathcal{E})$ which tells us how well we can approximate the channel $\mathcal{E}$ using the Stinespring dilation. In general this oracle could be implemented with some form of tomography, so we have to assume that it is very expensive to call this oracle. Assume that we found a decomposition

$$\mathcal{F} = \sum_{i=1}^{n_{pos}} a_i^+ \mathcal{E}_i^+ - \sum_{i=1}^{n_{neg}} a_i^- \mathcal{E}_i^- \tag{4.6}$$

[3] Here we refer to the number of qubits n on which $\mathcal{F}$ acts.

using the rank-constrained optimization. If we were to implement $\mathcal{F}$ with the quasiprobability method by using the Stinespring approximation of the involved channels $\mathcal{E}_i^{\pm}$, an error δ would occur where

$$\delta := \mathcal{F} - \sum_{i=1}^{n_{pos}} a_i^+ \mathcal{N}(\mathcal{E}_i^+) - \sum_{i=1}^{n_{neg}} a_i^- \mathcal{N}(\mathcal{E}_i^-)\,. \tag{4.7}$$

We can iteratively repeat our procedure, but this time decomposing δ instead of $\mathcal{F}$. During each point of the iteration we store the $\mathcal{N}(\mathcal{E}_i^{\pm})$ into our decomposition set. At some point our decomposition set will be large enough such that the resulting approximation error is smaller than a desired threshold. A conceptual visualization of the procedure is depicted in Figure 4.2 and the exact algorithm can be found in Algorithm 1.

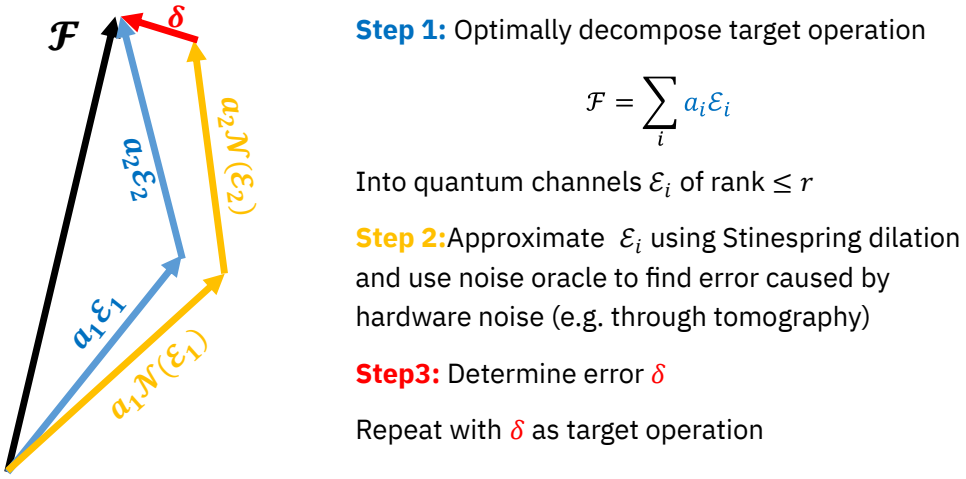


Figure 4.2: Graphical visualization of the iterative process involved in the Stinespring algorithm.

It is crucial that the Stinespring dilation allows for a reasonably accurate approximation of a desired quantum channel. In other words, the yellow arrow and blue arrow in Figure 4.2 have to be sufficiently close. In practice we observed that this was only possible with the inclusion of the rank constraint introduced before. The variational unitary approximation technique significantly improves the result further.

Remark 4.3 (Compensation method vs inverse method)**.** Algorithm 1 is formulated analogously to the compensation method, in the sense that we decompose the ideal unitary operation $[U]$ into a basis set containing the noisy variant thereof. It would also be possible to implement the inverse method instead by choosing the initial $\mathcal{F}$ to be the inverse map of the noise instead of $[U]$. But for the sake of simplicity, we will restrict ourselves to the compensation method for the rest of this chapter.

Algorithm 1: Stinespring Algorithm

Given: a target unitary operation $[U]$ for $U \in \mathrm{U}(A)$, a noise oracle $\mathcal{N}$, a threshold $dn_{\text{threshold}}$ for the diamond norm error;
Result: Decomposition set $\mathcal{D}$ and set of Stinespring dilation circuits $\mathcal{C}$

$\mathcal{C} := \{\text{circuit of } U\}$;
$\mathcal{D} := \{\mathcal{N}([U])\}$;
$\mathcal{F} := [U]$;
while *True* **do**
  $aqpd :=$ compute approximate QPD of $\mathcal{F}$ using decomposition set $\mathcal{D}$;
  $dn :=$ get diamond norm error of approximate QPD $aqpd$;
  if $dn < dn_{\text{threshold}}$ **then**
    break;
  end
  $\delta :=$ get the remaining error of the approximate qpd $aqpd$;
  $channels :=$ perform rank-constrained channel decomposition of δ into a set of channels;
  $StIsos =$ get Stinespring dilation isometries of the channels $channels$;
  $circuits =$ get variational unitary approximation circuits of the isometries $StIsos$;
  $ops_{\text{noisy}} =$ apply noise oracle $\mathcal{N}$ on the circuits in $circuits$;
  $\mathcal{C} = \mathcal{C} \cup circuits$;
  $\mathcal{D} = \mathcal{D} \cup ops_{\text{noisy}}$;
end

Remark 4.4 (Noise oracle)**.** If a good noise model of the hardware is available, it might be advantageous to use that to realize the noise oracle $\mathcal{N}$ required in the Stinespring algorithm. That way one doesn't have to iteratively perform tomography in each step of the Stinespring algorithm. Instead one could determine a suitable decomposition set using the noise model and the Stinespring algorithm, and only then perform tomography of the elements in the decomposition set at the very end.

4.4 Simulation Results

In this section we present simulation results of the Stinespring algorithm on the single- and two-qubit gates *Ry*, *CNOT* and *SWAP*, which we already analyzed in Section 2.7. For all three gates we enforce a rank-constraint $r = 2$ during the channel decomposition. This corresponds to allowing for at most one ancilla qubit during the Stinespring dilation. For the two-qubit gates

we additionally make use of the variational unitary approximation with a RYRZ variational form[4] of depth 6. The noise oracle is based on a noise model obtained from Qiskit [21], which aims to approximate the noise on the IBMQ Melbourne hardware platform. Using a noise model instead of a full tomography significantly speeds up the simulation. We use a threshold of $dn_{\text{threshold}} = 10^{-7}$ to stop the iterative procedure.

During each iteration of the Stinespring algorithm, we store the diamond norm error of the current approximate QPD (denoted dn in Algorithm 1). Figure 4.3 shows how this error decreases during the Stinespring algorithm. It is clearly visible that this decrease is exponential, which is not a surprising observation. It indicates that every Stinespring iteration reduces the remaining error by a roughly relative amount.

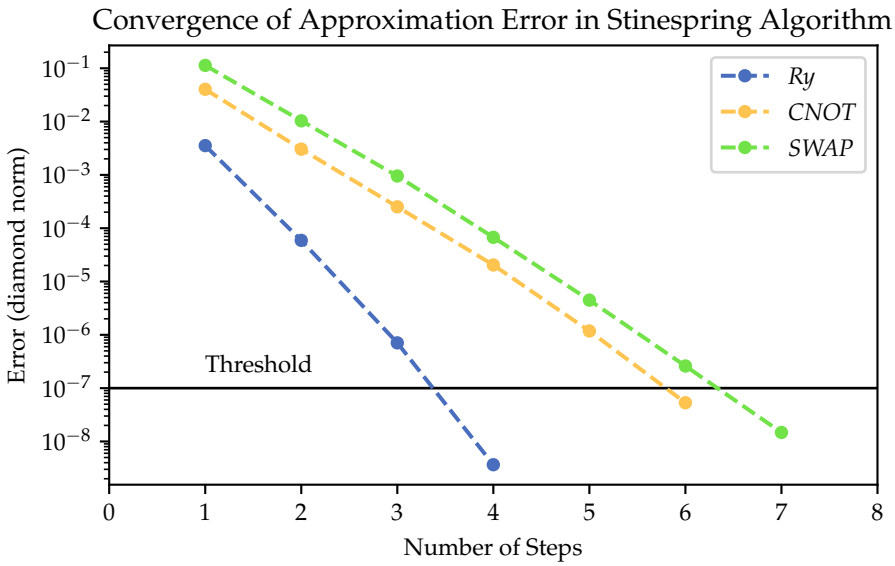


Figure 4.3: Evolution of the approximation error in each step of the Stinespring algorithm. Three different runs of the algorithm on three different gates *Ry*, *CNOT* and *SWAP* are depicted. The hardware noise is estimated using a noise model approximating the IBMQ Melbourne hardware backend. The horizontal grey line denotes the threshold at which the algorithm is programmed to stop.

For the two qubit gates, each iteration of the Stinespring algorithm extends the decomposition set by 16 operations (because $n_{pos} = n_{neg} = 8$). By considering that we need around 6-7 steps to reach the desired threshold, this implies that the produced decomposition set is significantly smaller than the Endo basis (around 70-80 elements instead of 256). This result is remarkable and indicates that the Stinespring algorithm really does find a decomposi-

[4]The details were introduced in Section 4.1.

tion set that is well adapted to the hardware noise. As a reminder, the 256 elements in the Endo basis were needed to completely span the space of Hermitian-preserving operators. The decomposition set produced by the Stinespring algorithm spans a significantly smaller space, as it is taylored to only represent one specific operation.

Table 4.1 denotes the C-factors obtained by finding the optimal QPD using the decomposition set produced by the Stinespring algorithm for the three gates in question. We see that a significant improvement can be observed compared to using the Endo basis. As previously discussed, the sampling overhead of the quasiprobability method scales exponentially in the number of gates, where the C-factor forms the basis of that exponential cost. Therefore any reduction in the C-factor is significant and allows us to implement significantly larger quantum circuits.

	C-factor	
	Endo basis	Stinespring algorithm
Ry	1.0106	1.0056
CNOT	1.1789	1.0812
SWAP	2.2095	1.2323

Table 4.1: Simulation results of the Stinespring algorithm applied to the three quantum gates $Ry(2\arccos\sqrt{0.56789})$, *CNOT* and *SWAP*. The obtained C-factor is compared to the value obtained when the respective gates are decomposed into the decomposition set consisting of the Endo basis and the corresponding noisy gate (as described in Section 2.7).

4.5 Discussion of Heuristics

It is natural to ask whether we can prove the convergence of the Stinespring algorithm or derive some bounds on the quality of the obtained solution. Unfortunately, it is seems difficult to make precise statements about the convergence, as several heuristics were used throughout this chapter. In this section we aim to provide a clear overview of these heuristics.

Existence of a low-rank channel decomposition: It is not immediately clear if Assumption 4.2 holds in practice or not. The fact that we observed a value of $\varepsilon > 0$ to be necessary for good convergence could be an indicator that it does indeed not hold. Furthermore it is not clear what minimal values of n_{pos}, n_{neg} are required to reach this minimum.

Consider a SDP of the form

$$\begin{cases} \min\limits_{X \in \mathbb{R}^{n \times n}} & \langle C, X \rangle \\ \text{s.t.} & \mathcal{A}(X) = b \\ & X \geq 0 \end{cases} \tag{4.8}$$

where $C \in \mathbb{R}^{n\times n}$ is a symmetric matrix, $\mathcal{A} : \mathbb{R}^{n\times n} \to \mathbb{R}^m$ a linear operator and $b \in \mathbb{R}^m$. A well-known result in semidefinite programming literature states that there exists a global optimum of rank r such that $r(r+1)/2 \leq m$ as long as the search space of the SDP is compact [33, 34]. It would be interesting to see if this result could somehow be applied to our problem.

Burer-Monteiro convergence: In recent years, there has been a huge progress in mathematically demonstrating the convergence and convergence speed of the Burer-Monteiro heuristic [31, 32]. However, these results usually apply to the regime $r(r+1)/2 \leq m$ and it is not clear if they can be translated to our setting.

Quality of the Stinespring dilation approximation: Even under the assumption that there always existed a solution of problem (4.2) and that we could always find this solution efficiently, the convergence speed of the Stinespring algorithm strongly depends on how well we can approximate an arbitrary quantum channel with the Stinespring dilation. More mathematically, we require the noise oracle $\mathcal{N}$ to be as closed to the identity as possible. It would be interesting for further work to derive some bound on the convergence speed and resulting C-factor that depend on $\mathcal{N}$. In practice it would still be difficult to evaluate this bound, as $\mathcal{N}$ has a very complicated structure, since it not only contains the full noise model, but also the variational unitary approximation.

Chapter 5

Interplay Between the Quasiprobability Method and Quantum Error Correction

The goal of the quasiprobability method is similar to the goal of quantum error correction: Both aim to suppress the effects of noise by simulating a noise-free quantum computer using a noisy quantum computer. The two approaches are very different in how they tackle this problem: The former tries to compensate for errors by averaging results over different runs, whereas the latter tries to detect and correct errors by encoding the quantum information redundantly in a larger Hilbert space. Quantum error correction doesn't suffer from the exponential cost which restricts the quasiprobability method to shallow circuits, so in the far future it will definitely be the more desirable alternative of the two. However, current quantum hardware does not meet the stringent requirements in fidelities and number of qubits necessary to implement quantum error correction and it will most likely not for the foreseeable future [3].

It is a natural, although vague, intuition that one would like to combine the two approaches. While full quantum error correction is currently not feasible, it might be possible to reduce the hardware requirements by assisting it with the quasiprobability method. This would most likely still imply the existence of an exponential cost in the circuit size, but it could be used as a stepping stone towards full quantum error correction. It would be ideal if there were some tunable parameter that would allow us to choose how much of the heavy lifting is done by quantum error correction and how much by quantum error mitigation. In the near future one would rely more on the error mitigation component, restricting us to shallower circuits. But as hardware would get better, although still not good enough for full quantum error correction, one could push this parameter more towards the regime where error mitigation plays a smaller role and thus reduce the C-factor cost.

This chapter is meant to capture some research ideas and failed attempts that were conducted during the master thesis. The work on this project is still ongoing and therefore the results in this section are not completely fleshed out yet. We will present multiple independent attempts to merge ideas from quantum error mitigation and correction.

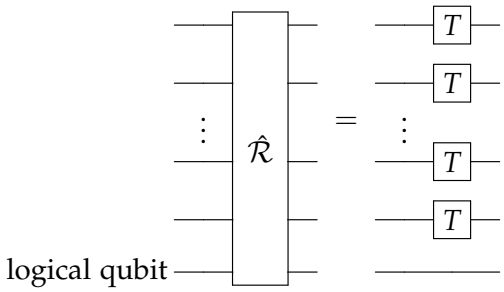
5.1 Implementing the Recovery Map using the Quasiprobability Method

The celebrated threshold theorem [35] ensures that quantum error correction is possible if a certain set of operations can be executed with a fidelity above a certain threshold. If only one operation fails to meet this criterion, fault tolerance generally cannot be achieved. The experimentally most difficult operation on current hardware platforms often tends to be the measurement operation (and the closely related reset operation). It is excessively noisy and requires orders of magnitude more time to execute than unitary gates, causing significant decoherence in the surrounding qubits which have to wait for the measurement to finish.

We asked ourselves the question, whether it might be possible to implement the recovery operation $\mathcal{R}$ of a quantum error correction code without the use of measurement-resets by employing the quasiprobability method. More precisely we would like to find a QPD of $\mathcal{R}$ into unitary operations (acting on the same number of qubits[1]). This would significantly simplify the implementation of quantum error correction codes on near-term noisy quantum hardware. Unfortunately this task is not possible, not even approximately. In the below section we will derive why this is the case.

Consider a stabilizer code which encodes 1 logical qubit into n physical qubits. By using a symplectic Gram-Schmidt procedure, one can construct an algebra of n *virtual* qubits which span the physical n-qubit Hilbert space. One of these virtual qubits plays the role of the logical qubit, whereas the other qubits are denoted as syndrome qubits. The logical subspace corresponds to the space where all syndrome qubits are in the state $|0\rangle$. There exists some unitary basis transformation $\tilde{U}$ which maps the physical qubits into the virtual qubits of the code. We consider the recovery operation in the virtual qubit basis, i.e. $\hat{\mathcal{R}} := [\tilde{U}]\mathcal{R}[\tilde{U}^{-1}]$. Since the recovery map consists of resetting the syndromes to zero and leaving the logical quantum information intact, $\hat{\mathcal{R}}$ can be expressed as a reset operation on the syndrome qubits and an identity operation on the logical qubit:

[1] If we make use of additional ancilla qubits in a fixed quantum state, then this would also implicate a need of periodic measurement-resets, if we are to reuse the ancilla qubit.



where T denotes a single-qubit reset operation. Since $\mathcal{R}$ and $\hat{\mathcal{R}}$ are related by a basis transformation, one can quickly see that finding a unitary-only QPD of one of them immediately implies that we can find a unitary-only QPD of the other. We argue that finding a unitary-only QPD of $\hat{\mathcal{R}}$ is impossible, since it is impossible to find a unitary-only QPD of the reset operation.

Lemma 5.1. *Consider the single-qubit reset operation T. There exists no QPD of T into unitary operations*

$$T = \sum_i a_i [U_i] \,. \tag{5.1}$$

Furthermore, T cannot be approximated arbitrarily closely with QPDs of this form.

Proof. Since we care about linear additivity in Equation (5.1), it is natural to represent the involved quantum channels via an isomorphic representation, such as the Choi representation. The Choi matrix of the reset operation is given by

$$\Lambda_T = \frac{1}{2} \sum_{\alpha=0,1} |\alpha\rangle\langle\alpha| \otimes |0\rangle\langle 0| = \frac{1}{2}(|00\rangle\langle 00| + |10\rangle\langle 10|) \tag{5.2}$$

and can thus be decomposed into

$$\Lambda_T = A + B \quad \text{where} \tag{5.3}$$

$$A = \frac{1}{4}(|00\rangle\langle 00| + |01\rangle\langle 01| + |10\rangle\langle 10| + |11\rangle\langle 11|) \quad \text{and} \tag{5.4}$$

$$B = \frac{1}{4}(|00\rangle\langle 00| - |01\rangle\langle 01| + |10\rangle\langle 10| - |11\rangle\langle 11|). \tag{5.5}$$

A is the fully-mixed state, so one can quickly convince oneself that $A = \frac{1}{8}(\Lambda_{Id} + \Lambda_X + \Lambda_Y + \Lambda_Z)$. This means that A is decomposable into unitary operations. However, the same does not hold for B. We show below that

$\langle B, \Lambda_U \rangle = 0$ for all unitaries U

$$\langle B, \Lambda_U \rangle \tag{5.6}$$
$$= \sum_{i,j} \mathrm{Tr}[|0\rangle\langle 0|i\rangle\langle j| \otimes |0\rangle\langle 0|U|i\rangle\langle j|U^\dagger + |0\rangle\langle 0|i\rangle\langle j| \otimes |1\rangle\langle 1|U|i\rangle\langle j|U^\dagger \tag{5.7}$$
$$+ |1\rangle\langle 1|i\rangle\langle j| \otimes |0\rangle\langle 0|U|i\rangle\langle j|U^\dagger + |1\rangle\langle 1|i\rangle\langle j| \otimes |1\rangle\langle 1|U|i\rangle\langle j|U^\dagger] \tag{5.8}$$
$$= \langle 0|U|0\rangle\langle 0|U^\dagger|0\rangle - \langle 1|U|0\rangle\langle 0|U^\dagger|1\rangle + \langle 0|U|1\rangle\langle 1|U^\dagger|0\rangle - \langle 1|U|1\rangle\langle 1|U^\dagger|1\rangle \tag{5.9}$$
$$= (|U_{00}|^2 + |U_{01}|^2) - (|U_{10}|^2 + |U_{11}|^2) \tag{5.10}$$
$$= 0\,, \tag{5.11}$$

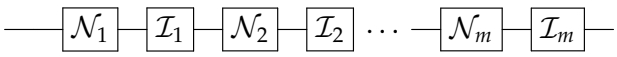
which thus implies that B is a remainder that is impossible to decompose or even approximate with unitary operations. To precisely formalize the statement about the impossibility of an approximation, one must consider that $\langle \mathcal{E}, \mathcal{F} \rangle_* := \langle \Lambda_\mathcal{E}, \Lambda_\mathcal{F} \rangle$ is an inner product on the space of quantum operations and use the fact that it is continuous (by linearity and boundedness, since we are working with finite-dimensional spaces). □

5.2 Logical Inverse

A central idea in quantum error correction is to encode logical qubits in multiple physical qubits. The effect of local quantum errors can be detected and corrected, as long as the weight of the error does not exceed a certain limit, which is determined by the distance of the code. Only errors of high weight can disturb the quantum information.

The idea presented in this section is as follows: We encode the logical quantum information using a n-qubit quantum code in an extended Hilbert space. Assume that we start with a state that lies in this logical subspace. After some noise occurs on the system, we perform an inverse operation to undo the noise, similar to the inverse method introduced in Section 2.7. But in contrary to the regular inverse method, we do not invert the complete noise that occurred, but rather only the noise on the logical subspace. We call such an inverse a *logical inverse*.

We illustrate the approach with following example on a single logical qubit: m consecutive noise operation $\mathcal{N}_i$ are applied on the encoded qubit. After every noise operation, we perform a logical inverse $\mathcal{I}_i$ that undoes the noise on the logical subspace.



The requirement that $\mathcal{I}_1$ shall perform a logical inverse can be mathematically expressed as

$$\mathcal{R} \circ \mathcal{I}_1 \circ \mathcal{N}_1 \circ \mathcal{R} = \mathcal{R} \,. \tag{5.12}$$

We want to chose a $\mathcal{I}_1$ that satisfies this criterion and minimizes the required C-factor. Once $\mathcal{I}_1$ is fixed, we can obtain $\mathcal{I}_2$ analogously with a different constraint given by

$$\mathcal{R} \circ \mathcal{I}_2 \circ \mathcal{N}_2 \circ \mathcal{I}_1 \circ \mathcal{N}_1 \circ \mathcal{R} = \mathcal{R} \,. \tag{5.13}$$

This is iteratively repeated until $\mathcal{I}_m$ is obtained. Note that the definition of $\mathcal{I}_j$ is non-local: it depends on everything that happened before. This is in stark contrast to the traditional quasiprobability method where one would perform full inverse operations, in which case $\mathcal{I}_j$ would only depend on $\mathcal{N}_j$.

The main goal is to show that this approach would allow for a reduced C-factor compared to the bare QPD, where every logical qubit is encoded directly as a physical qubit. First we try to demonstrate this numerically, but due to the computational complexity of the problem, this turns out to be infeasible. Therefore we will restrict ourselves to a simple code and noise model, which allows us to evaluate this advantage analytically.

5.2.1 Numerical Approach

Finding $\mathcal{I}_j$ numerically is done in the same way as finding any QPD, namely with the linear program in Equation (2.11). For example, $\mathcal{I}_1$ is given by $\sum_i \alpha_i \mathcal{E}_i$ where the coefficients α_i are determined with following LP:

$$\begin{array}{ll} \textbf{minimize} & \sum_i |\alpha_i| \\ \textbf{such that} & \mathcal{R} \circ (\sum_i \alpha_i \mathcal{E}_i) \circ \mathcal{N} \circ \mathcal{R} = \mathcal{R} \,, \end{array} \tag{5.14}$$

and $\{\mathcal{E}_i\}_i$ is a basis of operations which is fixed in advance. Once this LP is solved and we fixed $\mathcal{I}_1$, we can iteratively solve an analogous LP for $\mathcal{I}_2, \mathcal{I}_3$, etc...

The computational cost of this method scales linearly in m, but exponentially in n. This channel representations usually have 16^n variables. For $n = 3$ it is feasible to solve this problem numerically, as the LP has $\sim 4 \cdot 10^3$ constraints. $n = 5$ already has $\sim 10^6$ constraints, so it might be barely feasible. Anything beyond that is most likely out of our reach. Unfortunately, restricting ourselves to the parameter region where $n \leq 5$ is not very interesting, as we are interested in asymptotic statements in n.

5.2.2 Toy Example: Bit Flip Noise and the Repetition Code

We choose the most basic example of a code and noise in hope to get some useful insights from that. We encode our logical qubit in n physical qubit

using the repetition code (where n is odd). The noise map $\mathcal{N}_j$ are all identical (denoted simply as $\mathcal{N}$) and constituted of iid bit flip noise on every physical qubit with an error probability of ϵ.

Effective Bit Flip Probability

Here we describe a central insight in this toy example, which makes it analytically easy to analyze. The above described noise acts as an effective single-qubit bit flip channel on the logical subspace. To understand and derive this, we start off by considering the case of $m = 1$. The more general case will follow directly.

The noise map consists only of Pauli-X terms

$$\mathcal{N}(\rho) = \sum_{i \in \{0,1\}^n} a_i \cdot X^i \rho X^i \,, \tag{5.15}$$

where the a_i are some real coefficients. We now switch to the virtual qubit picture. The noise transforms as follows in the virtual qubit picture:

$$\hat{\mathcal{N}}(\hat{\rho}) = \sum_{i \in \{0,1\}^n} a_i \cdot (\tilde{U} X^i \tilde{U}^{-1}) \hat{\rho} (\tilde{U} X^i \tilde{U}^{-1}) \,. \tag{5.16}$$

We now use the fact that our initial state ρ is guaranteed to be in the logical subspace, i.e. it look like $\hat{\rho} = |0\rangle\langle 0|^{\otimes n-1} \otimes \rho_{logical}$ in the virtual qubit picture. To make the analysis a bit easier, let's even assume $\rho_{logical}$ is a pure state $\hat{\rho} = |\Psi\rangle\langle\Psi|$ for $|\Psi\rangle = \alpha |0 \ldots 00\rangle + \beta |0 \ldots 01\rangle$. Let's see how the above terms $(\tilde{U} X^i \tilde{U}^{-1})$ act on this state. $\tilde{U}^{-1}$ brings it to the state $\alpha |0 \ldots 0\rangle + \beta |1 \ldots 1\rangle$. X^i now causes $w(i)$ bit flips on this state, where $w(i)$ is the number of non-zero elements in the vector $i \in \{0,1\}^n$. $\tilde{U}$ brings us back to the virtual qubits. One immediately sees that, if we only consider the action on the logical qubit, then the expression $(\tilde{U} X^i \tilde{U}^{-1})$ will cause a bit flip if and only if $w(i) \geq \frac{n+1}{2}$. More precisely, the quantum channel restricted to the logical subspace acts as a bit flip channel

$$\rho \mapsto Tr_{0,1}[\hat{\mathcal{N}}(|0\rangle\langle 0|^{\otimes n-1} \otimes \rho)] = (1 - \tilde{\epsilon})\rho + \tilde{\epsilon} X \rho X \tag{5.17}$$

with an effective bit flip probability $\tilde{\epsilon}$ which is given by

$$\tilde{\epsilon}^{(n)} = \sum_{i \in \{0,1\}^n : w(i) \geq \frac{n+1}{2}} a_i. \tag{5.18}$$

Here the superscript (n) is meant to differentiate between the effective bit flip probability when different amount of physical qubits are used.

With the above insight, we now naturally see a simple strategy how to do a restricted inverse on the logical subspace. For any single-qubit bit-flip

channel with probability p there is a (non-physical) inverse map given by

$$\rho \mapsto \frac{1-p}{(1-p)^2}\rho - \frac{p}{(1-p)^2 - p^2} X\rho X\,. \tag{5.19}$$

We can apply this inverse map on the logical qubit. Translating this back to the picture of the physical qubits, the inverse maps looks like

$$\mathcal{I}_1(\rho) = \frac{1-\tilde{\epsilon}}{(1-\tilde{\epsilon})^2}\rho - \frac{\tilde{\epsilon}}{(1-\tilde{\epsilon})^2 - \tilde{\epsilon}^2} X^{\otimes n}\rho X^{\otimes n} \tag{5.20}$$

and is implementable with a C-factor of

$$C = \frac{1}{1-2\tilde{\epsilon}}\,. \tag{5.21}$$

Now that we know how $\mathcal{I}_1$ looks like, we finally can consider the $m > 1$ case. Because $\mathcal{I}_1$ also only contains Pauli-X terms, we can write $\mathcal{N} \otimes \mathcal{I}_1 \otimes \mathcal{N}$ in the form of Equation (5.15). Therefore we can repeat the same argument for $\mathcal{I}_2$, and thus iteratively for all $\mathcal{I}_j$. We denote the effective bit-flip probability seen by $\mathcal{I}_j$ as $\tilde{\epsilon}_j^{(n)}$. To also include the dependence on the original physical bit flip probability, we write $\tilde{\epsilon}_j^{(n)}(\epsilon)$.

Commutation of Noise and Logical Inverse

Recall that $\tilde{\epsilon}_j^{(n)}(\epsilon)$ is the quantity we care about, since it relates 1-to-1 to the C-factor of $\mathcal{I}_j$. From Equation (5.18) in the previous section, we can easily write down the $j = 1$ formula:

$$\tilde{\epsilon}_1^{(n)} = \sum_{k=\frac{n+1}{2}}^{n} \binom{n}{k} \epsilon^k (1-\epsilon)^{n-k}. \tag{5.22}$$

Note that this is basically one minus the CDF of the binomial distribution. The general formula for $j > 1$ seems rather daunting to write down it analytically. Fortunately, thanks to the simplicity of our model, we can use a trick to get an implicit relation which will simplify many things further down the line. Because $\mathcal{N}$ and all $\mathcal{I}_j$ are all Pauli-X channels, they commute. So we can group together all the noise into one big noise channel and compound all the logical inverses at the end. One can quickly convince oneself that the composition of m bit flip channels of probability ϵ is again a bit flip channel of probability $\frac{1}{2}(1-(1-2\epsilon)^m)$. Similarly the logical inverses aggregate into a $X^{\otimes n}$ joint n-qubit flip channel. Due to the commutativity, one can see after a bit of thinking, that searching for optimal $\mathcal{I}_j$ iteratively (i.e. first $\mathcal{I}_1$, then $\mathcal{I}_2$, etc...) must lead to the equivalent result to finding an optimal inverse

for the aggregated noise. Thanks to this insight we can now get following formula with the effective bit flip rate:

$$\text{bfprob}(\tilde{\epsilon}_1^{(n)}(\epsilon),\ldots,\tilde{\epsilon}_m^{(n)}(\epsilon)) = \tilde{\epsilon}_1^{(n)}(\frac{1}{2}-\frac{1}{2}(1-2\epsilon)^m)\,, \tag{5.23}$$

where $\text{bfprob}(p_1,\ldots,p_m)$ is the bit flip rate of a channel consisting of the composition of bit flip channels with probabilities $p_1,\ldots,p_m$, i.e. $\text{bfprob}(q) := q$ and

$$\text{bfprob}(q_1,\ldots,q_n) = \text{bfprob}(q_1,\ldots,q_{n-1})(1-q_n) + \text{bfprob}(q_1,\ldots,q_{n-1})q_n\,. \tag{5.24}$$

Equation (5.23) can be used as an implicit formula which allows us to efficiently compute $\tilde{\epsilon}_j^{(n)}(\epsilon)$ for $j > 1$.

Our insight also translates directly into the C-factor. The C-factor of doing the restricted inverses iteratively is the same C-factor of a single restricted inverse applied on the the bit flip noise with probability $\tilde{\epsilon}_1^{(n)}(\frac{1}{2}-\frac{1}{2}(1-2\epsilon)^m)$.

Bounds and Asymptotic Behavior

In Section 5.2.2 we realized that the quantity of interest is the effective bit flip rate. In Section 5.2.2 we saw that we can replace a system with $m > 1$ by a system with $m = 1$ with a modified physical bit flip rate and both will have the identical C-factor and inverse map (just split up differently across the circuit). So the quantity of interest we want to compute is

$$\bar{\epsilon} := \tilde{\epsilon}_1^{(n)}(\frac{1}{2}-\frac{1}{2}(1-2\epsilon)^m) \tag{5.25}$$

which is 1-to-1 related to the C-factor by $C = \frac{1}{1-2\bar{\epsilon}}$. We would now like to analyze the asymptotic behavior of $\bar{\epsilon}$ in n and m.

We can formulate $\bar{\epsilon}$ in terms of the CDF $F(k,n,p)$ of the binomial distribution

$$\bar{\epsilon} = 1 - F(\frac{n-1}{2}-1, n, \frac{1}{2}(1-(1-2\epsilon)^m) \tag{5.26}$$

$$= F(\frac{n-1}{2}, n, \frac{1}{2}(1+(1-2\epsilon)^m)\,. \tag{5.27}$$

This is useful, as we can now use the Chernoff and anti-concentration bounds from above and below

$$\frac{1}{\sqrt{2n}}\exp(-nD(\frac{k}{n}\|p)) \leq F(k,n,p) \leq \exp(-nD(\frac{k}{n}\|p))\,, \tag{5.28}$$

where $D(\cdot\|\cdot)$ denotes the relative entropy. Inserting k and p we quickly see that both $\frac{k}{n}$ and p converge to $\frac{1}{2}$. This allows us to perform the Talyor expansion

$$D(x\|y) \approx 2(x-\frac{1}{2})^2 + 2(y-\frac{1}{2})^2 - 4(x-\frac{1}{2})(y-\frac{1}{2})\,. \tag{5.29}$$

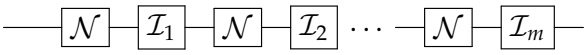
Furthermore the $\sqrt{2n}$ term becomes irrelevant for large n. We thus obtain

$$\bar{\epsilon} \sim \exp(-\frac{1}{2n} - \frac{1}{2}n(1-2\epsilon)^{2m} - (1-2\epsilon)^{m}) \,. \tag{5.30}$$

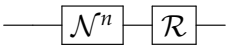
One direct consequence of this result is that the effective bit flip rate decays exponentially in n.

Summary of results

We compare the previously described setting



with regular quantum error correction



under the assumption that we have access to an ideal recovery map $\mathcal{R}$. Quantum error correction does not incur an additional sampling overhead, but it also doesn't correct logical errors perfectly. This is why fault tolerant quantum error correction either requires large n or code concatenation. Our method can correct any logical errors perfectly, but the sampling overhead means that we have to pay a price that is exponential in the circuit depth. By performing the Taylor expansion at $\epsilon \approx 0$ of Equation (5.22) one can analytically compute the C-factor and the error that occur in our method and in the QEC setting respectively. The result is depicted in Table 5.1. Interestingly, the C-factor of the logical inverse approach behaves in a very analogous manner as the error in the quantum error correction scheme: It is suppressed exponentially in the number of qubits in the code. We expect a similar behavior to manifest for general codes and general noise models, with the difference that the suppression would behave as $\epsilon^{\frac{d+1}{2}}$ where d is the distance of the code in question.

5.2.3 Drawbacks of the Approach

The major drawback of this approach is that we require knowledge of quantum channels that span wide parts of our circuit. More precisely, to compute $\mathcal{I}_j$ one requires knowledge about the channel $\mathcal{N}_j \circ \mathcal{I}_{j-1} \circ \cdots \circ \mathcal{I}_1 \circ \mathcal{N}_1$. It is not tractable to experimentally perform tomography on this channel, especially if one considers a more realistic setup with multiple logical qubits, where multi-qubit gates entangle different logical qubits between two logical inverses. Generally the method would require tomography of subcircuits

	C-factor	Error (diamond norm)
Bare QPD	$1+2m\epsilon$	0
Quantum Error Correction	0	$2\binom{n}{\frac{2+1}{2}}(m\epsilon)^{\frac{n+1}{2}}$
Logical Inverse Approach	$1+2\binom{n}{\frac{2+1}{2}}(m\epsilon)^{\frac{n+1}{2}}$	0

Table 5.1: Benchmark of three different approaches to correcting m consequent applications of a bit flip noise with iid rate ϵ to a single logical qubit. For each method the resulting error and C-factor are depicted. Bare QPD denotes using the regular quasiprobability method where 1 logical qubit is encoded in 1 physical qubit. Quantum Error Correction denotes the case where the logical qubit is encoded in n physical qubits using the repetition code. Logical inverse approach denotes the method where the qubit is encoded in n physical qubits using the repetition code and a logical inverse map is implemented after every noise channel using the quasiprobability method. The formulas in the table are valid in the regime $\epsilon \to 0$.

that are as wide as the total circuit in question, a task which is as expensive as classically simulating the circuit in the first place.

The only manner how this method could be applied as-is would be to somehow restrict it to small subcircuits that are not too wide. A larger circuit could be subdivided in smaller manageable circuits by periodically inserting recovery operations inside the circuit. Each chunk between two recovery operations would then be individually corrected with the logical inverse method. Generally, there is a tradeoff to be made here: If the chunks are very small, then we gain no C-factor over the the bare quasiprobability method. If the chunks are very large, we have to pay with more expensive tomography operations, and at some point this tomography becomes intractable. Depending on the circuit in question, there might be a sweet spot in between these two extremes where our method could be of practical interest.

Acknowledgement

I would like to thank David Sutter who provided me exceptional support for all matters related to my master thesis. He provided me extensive guidance on the scientific and the non-scientific aspects of the research process. Furthermore I would like to thank Stefan Wörner, Kristan Temme, Sergey Bravyi and Jay Gambetta for the many insightful discussions which have significantly contributed to the development of the thesis. Finally I would like to thank the whole quantum theory and application team at IBM Rüschlikon for the friendly and motivating atmosphere under which I had the pleasure to write my thesis.

Appendix A

Initial Guess for Rank-Constrained Channel Decomposition

The task at hand is to solve following optimization problem:

$$\begin{cases} \min\limits_{a_i^{\pm}\in\mathbb{R}^+, X_i^{\pm}\in\mathbb{C}^{2\times 4^n}} & \left\| \Lambda_{\mathcal{F}} - \left(\sum\limits_{j=1}^{n_{pos}} \tilde{\Lambda}^{+}_{\mathcal{E}_j} - \sum\limits_{j=1}^{n_{neg}} \tilde{\Lambda}^{-}_{\mathcal{E}_j}\right)\right\|_2^2 + \\ & \sum\limits_{j=1}^{n_{pos}} \left\| \mathrm{Tr}_2[\tilde{\Lambda}^{+}_{\mathcal{E}_j}] - a_j^{+}\tfrac{1}{2^n}\mathbb{1}\right\|_2^2 + \sum\limits_{j=1}^{n_{neg}} \left\| \mathrm{Tr}_2[\tilde{\Lambda}^{-}_{\mathcal{E}_j}] - a_j^{-}\tfrac{1}{2^n}\mathbb{1}\right\|_2^2 \\ \text{s.t.} & f^* \leq \sum_{j=1}^{n_{pos}} a_j^{+} + \sum_{j=1}^{n_{neg}} a_j^{-} \leq f^*\cdot(1+\varepsilon) \\ & \tilde{\Lambda}^{\pm}_{\mathcal{E}_i} = (X_i^{\pm})^{\dagger}\cdot X_i^{\pm}\,. \end{cases} \tag{A.1}$$

We restrict ourselves to the case of $r = 2$, which corresponds to allowing a single ancilla qubit in the Stinespring dilation. In order to solve this problem with a local method, an initial guess $a_i^{\pm,0}, X_i^{\pm,0}$ for the parameters is required. In this section we present a heuristic to find such initial values which seems to work well in our experience.

We start off by computing the CDD $\mathcal{F} = a^+\mathcal{E}^+ - a^-\mathcal{E}^-$ of our target operation. We consider the spectral decomposition of the Choi matrices of $\mathcal{E}^{\pm}$

$$\Lambda_{\mathcal{E}^{\pm}} = \begin{pmatrix} | & & | \\ u_1^{\pm} & \dots & u_{4^n}^{\pm} \\ | & & | \end{pmatrix} \cdot diag(\lambda_1,\dots,\lambda_{4^n}) \cdot \begin{pmatrix} — & (u_1^{\pm})^{\dagger} & — \\ & \vdots & \\ — & (u_{4^n}^{\pm})^{\dagger} & — \end{pmatrix} \tag{A.2}$$

$$= \sum_{i=1}^{4^n} \lambda_i u_i^{\pm}\cdot(u_i^{\pm})^{\dagger} \tag{A.3}$$

where $\lambda_i^{\pm}$ denote the eigenvalues and $u_i^{\pm}$ denote the corresponding eigenvectors. We chose the indices such that the $\lambda_i^{\pm}$ are ordered non-increasingly

in i. Equation (A.3) is a decomposition of $\mathcal{E}^{\pm}$ into rank-1 operations. Since we want to find a decomposition into rank-2 operations, we group these rank-1 matrices into pairs:

$$\Lambda_{\mathcal{E}^{\pm}} = \sum_{i=1}^{4^n/2} Y_i^{\pm} \cdot (Y_i^{\pm})^{\dagger} \text{ where} \tag{A.4}$$

$$Y_i = \begin{pmatrix} | & | \\ \sqrt{\lambda_{2i}} u_{2i}^{\pm} & \sqrt{\lambda_{2i+1}} u_{2i+1}^{\pm} \\ | & | \end{pmatrix} \tag{A.5}$$

We chose our initial guess as

$$a_i^{\pm,0} := \mathrm{Tr}[Y_i] \text{ and } X_i^{\pm,0} := \frac{Y_i}{\mathrm{Tr}[Y_i]}\,. \tag{A.6}$$

If $n_{pos} = n_{neg} < \frac{1}{2}4^n$ this implies that we only consider the $2n_{pos}$ largest eigenvalues and discard the others. As already mentioned, in our implementation we chose $n_{pos} = n_{neg} = \frac{1}{2}4^n$[1]. This way

$$\Lambda_{\mathcal{F}} = (\sum_{i=1}^{n_{pos}} \tilde{\Lambda}_{\mathcal{E}_i}^{+} - \sum_{i=1}^{n_{neg}} \tilde{\Lambda}_{\mathcal{E}_i}^{-}) \tag{A.7}$$

is already fulfilled by the initial guess. Furthermore we also already fulfill that the C-factor

$$\sum_{i=1}^{n_{pos}} a_i^{+} + \sum_{i=1}^{n_{neg}} a_i^{-} = f^*\,. \tag{A.8}$$

Therefore the only condition that is not yet fulfilled is the trace-preservingness of the $\tilde{\Lambda}_{\mathcal{E}_i}^{\pm}$.

[1] We chose $n_{pos} = n_{neg} = 1$ for $n = 2$ and $n_{pos} = n_{neg} = 8$ for $n = 2$.

Bibliography

[1] K. Temme, S. Bravyi, and J. M. Gambetta. Error mitigation for short-depth quantum circuits. *Physical Review Letters*, 119(18), 2017. DOI: 10.1103/physrevlett.119.180509.

[2] P. W. Shor. Scheme for reducing decoherence in quantum computer memory. *Phys. Rev. A*, 52:R2493–R2496, 1995. DOI: 10.1103/PhysRevA.52.R2493.

[3] J. Preskill. Quantum computing in the NISQ era and beyond. *Quantum*, 2:79, 2018. DOI: 10.22331/q-2018-08-06-79.

[4] J. R. McClean, M. E. Kimchi-Schwartz, J. Carter, and W. A. de Jong. Hybrid quantum-classical hierarchy for mitigation of decoherence and determination of excited states. *Phys. Rev. A*, 95:042308, 2017. DOI: 10.1103/PhysRevA.95.042308.

[5] Y. Li and S. C. Benjamin. Efficient variational quantum simulator incorporating active error minimization. *Phys. Rev. X*, 7:021050, 2017. DOI: 10.1103/PhysRevX.7.021050.

[6] M. Otten and S. Gray. Accounting for errors in quantum algorithms via individual error reduction. *npj Quantum Information*, 5, 2018. DOI: 10.1038/s41534-019-0125-3.

[7] H. Pashayan, J. J. Wallman, and S. D. Bartlett. Estimating outcome probabilities of quantum circuits using quasiprobabilities. *Phys. Rev. Lett.*, 115:070501, 2015. DOI: 10.1103/PhysRevLett.115.070501.

[8] N. Delfosse, P. Allard Guerin, J. Bian, and R. Raussendorf. Wigner function negativity and contextuality in quantum computation on rebits. *Phys. Rev. X*, 5:021003, 2015. DOI: 10.1103/PhysRevX.5.021003.

[9] S. Endo, S. C. Benjamin, and Y. Li. Practical quantum error mitigation for near-future applications. *Phys. Rev. X*, 8:031027, 2018. DOI: 10.1103/PhysRevX.8.031027.

[10] A. Peruzzo, J. McClean, P. Shadbolt, M.-H. Yung, X.-Q. Zhou, P. J. Love, A. Aspuru-Guzik, and J. L. O'Brien. A variational eigenvalue solver on a photonic quantum processor. *Nature Communications*, 5(1), 2014. DOI: 10.1038/ncomms5213.

[11] J. R. McClean, J. Romero, R. Babbush, and A. Aspuru-Guzik. The theory of variational hybrid quantum-classical algorithms. *New Journal of Physics*, 18(2):023023, 2016. DOI: 10.1088/1367-2630/18/2/023023.

[12] D. Grinko, J. Gacon, C. Zoufal, and S. Woerner. Iterative quantum amplitude estimation, 2019. Available online: https://arxiv.org/abs/1912.05559.

[13] Y. Suzuki, S. Uno, R. Raymond, T. Tanaka, T. Onodera, and N. Yamamoto. Amplitude estimation without phase estimation. *Quantum Information Processing*, 19(2), 2020. DOI: 10.1007/s11128-019-2565-2.

[14] S. Bravyi, A. Kliesch, R. Koenig, and E. Tang. Obstacles to state preparation and variational optimization from symmetry protection, 2019. Available online: https://arxiv.org/abs/1910.08980.

[15] D. Greenbaum. Introduction to quantum gate set tomography, 2015. Available online: https://arxiv.org/abs/1509.02921.

[16] A. Strikis, D. Qin, Y. Chen, S. C. Benjamin, and Y. Li. Learning-based quantum error mitigation, 2020. Available online: https://arxiv.org/abs/2005.07601.

[17] R. Harper, S. T. Flammia, and J. J. Wallman. Efficient learning of quantum noise. *Nature Physics*, 2020. DOI: 10.1038/s41567-020-0992-8.

[18] R. C. Bialczak, M. Ansmann, M. Hofheinz, E. Lucero, M. Neeley, A. D. O'Connell, D. Sank, H. Wang, J. Wenner, M. Steffen, and et al. Quantum process tomography of a universal entangling gate implemented with josephson phase qubits. *Nature Physics*, 6(6):409–413, 2010. DOI: 10.1038/nphys1639.

[19] M. Neeley, R. C. Bialczak, M. Lenander, E. Lucero, M. Mariantoni, A. D. O'Connell, D. Sank, H. Wang, M. Weides, J. Wenner, and et al. Generation of three-qubit entangled states using superconducting phase qubits. *Nature*, 467(7315):570–573, 2010. DOI: 10.1038/nature09418.

[20] J. M. Gambetta, A. D. Córcoles, S. T. Merkel, B. R. Johnson, J. A. Smolin, J. M. Chow, C. A. Ryan, C. Rigetti, S. Poletto, T. A. Ohki, M. B. Ketchen, and M. Steffen. Characterization of addressability by simultaneous randomized benchmarking. *Phys. Rev. Lett.*, 109:240504, 2012. DOI: 10.1103/PhysRevLett.109.240504.

[21] H. Abraham, AduOffei, R. Agarwal, et al. Qiskit: An open-source framework for quantum computing, 2019. DOI: 10.5281/zenodo.2562110.

[22] S. Diamond and S. Boyd. CVXPY: A Python-embedded modeling language for convex optimization. *Journal of Machine Learning Research*, 17(83):1–5, 2016.

[23] A. Agrawal, R. Verschueren, S. Diamond, and S. Boyd. A rewriting system for convex optimization problems. *Journal of Control and Decision*, 5(1):42–60, 2018. DOI: 10.1080/23307706.2017.1397554.

[24] MOSEK ApS. *MOSEK Optimizer API for Python 9.2.8*, 2020. Available online: https://docs.mosek.com/9.2/pythonapi/index.html.

[25] J. Watrous. Simpler semidefinite programs for completely bounded norms. *Chicago Journal of Theoretical Computer Science*, 2013(8), 2013. DOI: 0.4086/cjtcs.2013.008.

[26] E. Jones, T. Oliphant, P. Peterson, et al. SciPy: Open source scientific tools for Python, 2001–. Available online: http://www.scipy.org/.

[27] C. J. Wood, J. D. Biamonte, and D. G. Cory. Tensor networks and graphical calculus for open quantum systems, 2011. Available online: https://arxiv.org/abs/1111.6950.

[28] D. Maclaurin, D. Duvenaud, and R. P. Adams. Autograd: Effortless gradients in NumPy. *ICML 2015 AutoML Workshop*, 2015. Available online: https://indico.lal.in2p3.fr/event/2914/session/1/contribution/6/3/material/paper/0.pdf.

[29] S. Burer and R. Monteiro. A nonlinear programming algorithm for solving semidefinite programs via low-rank factorization. *Mathematical Programming, Series B*, 95:329–357, 2003. DOI: 10.1007/s10107-002-0352-8.

[30] S. Burer and R. Monteiro. Local minima and convergence in low-rank semidefinite programming. *Mathematical Programming*, 103:427–444, 2005. DOI: 10.1007/s10107-004-0564-1.

[31] D. Cifuentes. On the burer-monteiro method for general semidefinite programs, 2019. Available online: https://arxiv.org/abs/1904.07147.

[32] N. Boumal, V. Voroninski, and A. S. Bandeira. The non-convex burer-monteiro approach works on smooth semidefinite programs, 2016. Available online: https://arxiv.org/abs/1606.04970.

[33] G. Pataki. On the rank of extreme matrices in semidefinite programs and the multiplicity of optimal eigenvalues. *Mathematics of Operations Research*, 23(2):339–358, 1998. DOI: 10.1287/moor.23.2.339.

[34] A. Barvinok. Problems of distance geometry and convex properties of quadratic maps. *Discrete and Computational Geometry*, 13:189–202, 1995. DOI: 10.1007/BF02574037.

[35] D. Aharonov and M. Ben-Or. Fault-tolerant quantum computation with constant error rate. *SIAM Journal on Computing*, 38(4):1207–1282, 2008. DOI: 10.1137/S0097539799359385.

ETH

Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Declaration of originality

The signed declaration of originality is a component of every semester paper, Bachelor's thesis, Master's thesis and any other degree paper undertaken during the course of studies, including the respective electronic versions.

Lecturers may also require a declaration of originality for other written papers compiled for their courses.

I hereby confirm that I am the sole author of the written work here enclosed and that I have compiled it in my own words. Parts excepted are corrections of form and content by the supervisor.

Title of work (in block letters):

Advanced Methods for Quasiprobabilistic Quantum Error Mitigation

Authored by (in block letters):

For papers written by groups the names of all authors are required.

Name(s):	**First name(s):**
Piveteau	Christophe

With my signature I confirm that

- I have committed none of the forms of plagiarism described in the 'Citation etiquette' information sheet.
- I have documented all methods, data and processes truthfully.
- I have not manipulated any data.
- I have mentioned all persons who were significant facilitators of the work.

I am aware that the work may be screened electronically for plagiarism.

Place, date	**Signature(s)**
Zürich, 28.09.2020	C. Piveteau

For papers written by groups the names of all authors are required. Their signatures collectively guarantee the entire content of the written paper.