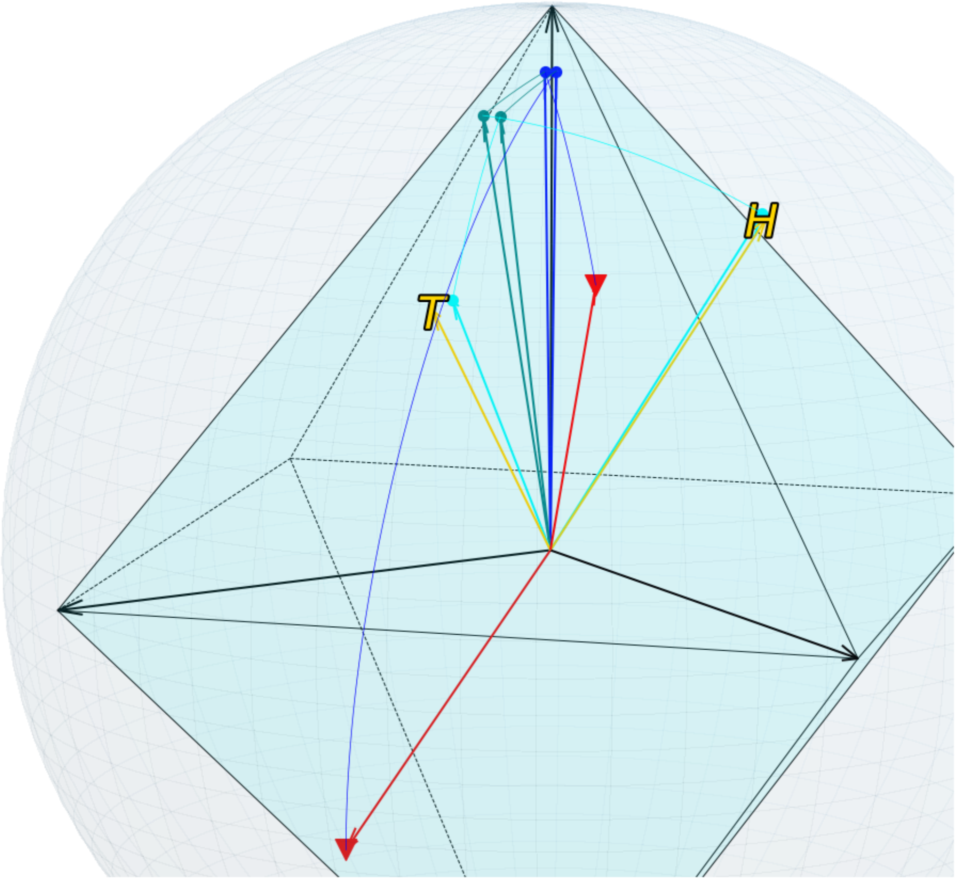
CHALMERS
UNIVERSITY OF TECHNOLOGY



Study of producing resourceful quantum states via modular combinations of two-qubit circuits

Investigation of an algorithm similar to magic state distillation using a restricted set of Clifford circuits

Master's thesis in Physics

Alex Maltesson

DEPARTMENT OF MICROTECHNOLOGY AND NANOSCIENCE

CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2024
www.chalmers.se

Master's thesis 2024

Study of producing resourceful quantum states via modular combinations of two-qubit circuits

Investigation of an algorithm similar to magic state distillation using a restricted set of Clifford circuits

ALEX MALTESSON

CHALMERS
UNIVERSITY OF TECHNOLOGY

Department of Microtechnology and Nanoscience
Division of Applied Quantum Physics
Chalmers University of Technology
Gothenburg, Sweden 2024

Study of producing resourceful quantum states via modular combinations of two-qubit circuits
Investigation of an algorithm similar to magic state distillation using a restricted set of Clifford circuits
ALEX MALTESSON

Supervisor: Cameron Calcluth, Department of Microtechnology and Nanoscience
Examiner: Giulia Ferrini, Department of Microtechnology and Nanoscience

Master's Thesis 2024
Department of Microtechnology and Nanoscience
Division of Applied Quantum Physics
Chalmers University of Technology
SE-412 96 Gothenburg
Telephone +46 31 772 1000

Cover: An illustration of how multiple optimization rounds of the modular magic synthesis algorithm can shift a rotated input state on the Bloch sphere to become closer to the target magic state.

Typeset in LaTeX
Printed by Chalmers Reproservice
Gothenburg, Sweden 2024

Study of producing resourceful quantum states via modular combinations of two-qubit circuits
Investigation of an algorithm similar to magic state distillation using a restricted set of Clifford circuits
ALEX MALTESSON
Department of Microtechnology and Nanoscience
Chalmers University of Technology

Abstract

Quantum computers are a technology that has garnered much attention throughout the last decades. This interest can partially be attributed to the realization that quantum computers seemingly can accomplish some computational tasks more efficiently than classical computers. The computational components that a quantum computer requires for these speed-ups are called resources. In this thesis, we implement and study an algorithm that aims to obtain a set of resourceful states that could mediate computational speed-up, even exponential speed-up, which are called T- and H-type magic states. This algorithm is named modular magic synthesis (MMS) and is inspired by a method proposed by Sergey Bravyi and Alexei Kitaev in Ref. [1], which can obtain magic states that are arbitrarily close to the target with a process named magic state distillation (MSD). The quality of the output state from the algorithms is characterized by fidelity, where MMS and MSD are both implemented to obtain an output state with a higher fidelity to the target by a similar optimization procedure of consuming several faulty input magic states in multiple optimization rounds. The defining characteristic of the MMS algorithm, which deviates from the MSD method, is that MMS only requires two states as inputs for each round of optimization, whereas the MSD algorithm needs at least five input states for distillation. The results we acquired from implementing the MMS algorithm showed that it could not increase the fidelity of the input state to the T state, while some improvement of the input state to the H state appeared feasible. However, the MMS algorithm could only increase the fidelity of the input state to the H state up to a certain point. It is because of this important realization that we do not consider the MMS algorithm as a genuine “distillation” method, and instead refer to it as a “synthesis” algorithm.

Keywords: Quantum computers, Modular magic synthesis, Magic state distillation, Clifford gates, post-selection

Acknowledgements

I sincerely want to thank,

Cameron Calcluth, Giulia Ferrini, and the research group for the chance to perform this thesis, the constructive guidance, and the opportunity to continue to work with you in the future.

Family and friends who have supported me throughout the process of conducting this thesis and the years that led up to this.

This thesis could not have been done without you all.

Alex Maltesson, Gothenburg, June 2024

Acronyms

Here is a list of acronyms that are commonly used throughout the thesis:

MSD	Magic state distillation
MMS	Modular magic synthesis
UQC	Universal quantum computer
TMMS	Tracing out magic state distillation

Nomenclature

This is the nomenclature of the notations, indices, groups, and quantum gates that are used in the thesis:

Notations

F_T^*	Optimal threshold fidelity for T-type distillation
$F_{\mathrm{BK},T}^*$	Threshold fidelity obtained by Bravyi & Kitaev for T-type distillation
F_H^*	Optimal threshold fidelity for H-type distillation
$F_{\mathrm{BK},H}^*$	Threshold fidelity obtained by Bravyi & Kitaev for H-type distillation
ε_T^*	Optimal threshold error probability for T-type distillation
ε_H^*	Optimal threshold error probability for H-type distillation
ε	Initial error probability
θ	Initial rotation angle

Indices

i, j, k, m, n, o	Indices for undefined number of entries
p, q, r	Indices for $\{x, y, z\}$

Groups

$\mathcal{C}_i$	The i-qubit Clifford group
$\mathcal{C}_i'$	The i-qubit Clifford quotient group
$\mathcal{P}_i$	The i-qubit Pauli group

Quantum gates

σ^x	Pauli-X

σ^y	Pauli-Y
σ^z	Pauli-Z
H	Hadamard
S	Phase shift
$CNOT$	Controlled-NOT
I	Identity
$\Sigma\left(\phi\right)$	General phase shift
L	The unitary which has the T state as its eigenvector
$R\left(\theta\right)$	y rotation with angle θ

Contents

Chapter 1

Introduction

The standard "classical" computer, which encodes the smallest unit of information as bits and uses transistors to perform its calculations, has found many applications throughout the last century and has become indispensable to our modern lives. Despite the success and the rapid development of the classical computer, researchers have simultaneously hypothesized computers that can utilize other fundamental principles to perform their objectives which might prove beneficial to solve certain computational problems. In the late 20th century, an invention was theorized in accordance with these ideas that were based on the theory of quantum physics, which had been formulated throughout the last century [2], [3], [4], and it became known as a quantum computer [5], [6].

The general idea of a quantum computer is a computational device that can harness the aspects of quantum mechanics, such as superposition and entanglement, to perform its calculations. The interest in quantum computers partially stems from the realization that they can perform certain tasks quicker than a classical computer. For instance, one of the most famous quantum algorithms that allow the quantum computer to perform its function faster than a classical computer is Shor's algorithm of factorization [7], [8]. Shor's algorithm can accomplish factorization efficiently, which means that the algorithm runtime will scale polynomially compared to the size of the problem [2], and this is not possible for any known factorization algorithm on a classical computer [9]. Other quantum algorithms that can display speed-ups in comparison with classical computer algorithms include, for example, Grover's search algorithm [10], [11], but there have additionally been studies that hint at the potential advantages of applying quantum computers to solve optimization problems [12] and enhance machine learning [13], among others. However, these benefits are still subject to debate [14].

The theory of quantum computers and quantum information that have emerged during the last decades was in part inspired by the well-established field of computer science. One of the foundational notions that have permeated the field of computer science since its inception, even before the first classical computers were realized, is the concept of a universal computing machine that can perform any imaginable calculation. An early articulation of these ideas is the Church-Turing thesis, which effectively states that any conceivable functions can be calculated by a universal computing device, often referred to as a Turing machine [2], [15], [16]. Given that it seems as though certain computational problems cannot be solved efficiently by some models of computation, this Church-Turing thesis has been appended to specify that the Turing machine can solve the function efficiently, but this extension is still a debated topic [2], [9].

Correspondingly to these ideas, the concept of a universal quantum computer (UQC) was formulated to characterize a quantum computer that would display the aforementioned speed-ups and could thus not be simulated by a classical computer [17]. However, the computational components that a quantum computer requires to demonstrate these enhancements in performance have proven to be a deep and nuanced topic. These important features are in this case quantum physical properties, like superposition and entanglement, or quantum states. One possible formulation of this problem is from the perspective of quantum resource theory, which is a branch of quantum computing theory where properties that enable these advantages are identified as resourceful and thus make it possible to categorize these attributes [18].

Inspired by these concepts, the goal of this thesis is to implement and study an algorithm that aims to obtain a set of resourceful states called magic states. These states are desirable because they can supplement a set of quantum computer actions that are classically simulatable with crucial resources to promote them to constitute a universal quantum computer. The implemented algorithm will be called modular magic synthesis (MMS) and it is inspired by a method proposed by Bravyi & Kitaev [1] named magic state distillation (MSD), with which MMS will resemble in many aspects. The MSD algorithm can transform several faulty input magic states to one output state that is closer to the target magic state and by utilizing multiple rounds of optimization, MSD can create a state that is arbitrarily close to the target. The MMS algorithm will be implemented in a similar manner, but one of the defining characteristics that differentiates the methods is that MMS only will require two input qubits, while the MSD algorithm needs five or fifteen inputs, depending on the state that is being distilled. However, input states to these algorithms need to be sufficiently close to the target for the protocol to function as intended. The question that this thesis will investigate is: which states can be transformed to a magic state when restricted to certain operations and can a distillation process still be possible with our version?

The thesis begins with a presentation of the theoretical background that is required to study the MMS algorithm in Chapter 2. This conceptual basis will include a preface to quantum computation and an introduction to the Bravyi & Kitaev MSD algorithm. Consequently, the implementation and construction of the MMS algorithm will be showcased in Chapter 3, which will then be followed by a presentation of the obtained results and discussion regarding the findings in Chapter 4. Finally, a short conclusion and suggestions for possible extensions of the thesis will then be given in Chapter 5.

Chapter 2

Background

In this chapter, we give a preamble to the theoretical framework which underlies this thesis. The chapter consists of three parts. The first section will contain some general concepts central to quantum physics. These ideas will then lead into the second section, where we will describe the quantum computing theory that is utilized throughout this thesis. Finally, the third section gives an overview of the Bravyi & Kitaev magic state distillation algorithm, which will be used as a reference in subsequent chapters for the implementation of the modular magic syntheses algorithm.

2.1 Quantum physics theory

Quantum physics is a extensive field of science that has allowed for the explanation of phenomena in nature that cannot be described by classical physics. In general, a quantum mechanical object exhibits the peculiar trait that its precise state cannot be known before measurement, unless a measurement has already been conducted or the object is prepared in a particular state [3], [4]. The statistics of the object can however be described by a normalized wave function, which is a mathematical formulation of the knowledge regarding a quantum system that is defined in a vector space called a Hilbert space. The wave function can be used to obtain the probability of observing the system in a particular condition, but, after measurement, the wave function will collapse to the detected outcome [4]. The observable values are described by Hermitian operators, where operators are in general objects that can transform the wave function and Hermitian means that it is invariant under conjugation, $A^\dagger = \left(A^T\right)^* = A$. Operators are often denoted with a hat-symbol ($\hat{}$), but these will be omitted in this thesis for simplicity. The wave function can be either discrete or continuous depending on the systems that are being described. For instance, spin can be formulated with a discrete wave function, while the position of an object is usually described with a continuous wave function [3]. In this thesis, only discrete wave functions will be considered.

There are many equivalent representations of states and operators in quantum physics. In the next parts of this section, some relevant representations and operators for this thesis will be presented and described.

2.1.1 Bra-ket representation of single system & the Pauli operators

The most prevalent and convenient formulation of quantum physics regarding discrete systems is linear algebra, which stems from the notion that a wave function can be described as vectors and operators as linear transformations. In this thesis, the single system quantum states will be described as a two-dimensional complex vector. Following the notation of quantum physics proposed by P. A. M. Dirac, this kind of vector will be noted as a ket $|\psi\rangle$ [3]. A general single system can be written as a linear combination of orthonormal basis states, $|\alpha_0\rangle$ and $|\alpha_1\rangle$, according to

$$|\psi\rangle = a_0 |\alpha_0\rangle + a_1 |\alpha_1\rangle \,, \quad \text{where } a_0, a_1 \in \mathbb{C}. \tag{2.1}$$

The system $|\psi\rangle$ can be in a superposition of the basis states and is defined in the two-dimensional Hilbert space $|\psi\rangle \in \mathcal{H}_2$ which is spanned by the vectors $\{|\alpha_i\rangle\}$ [2], [16]. The coefficients a_0 and a_1 are referred to as amplitudes and their squared absolute value can be interpreted as the probability of finding the system $|\psi\rangle$ in $|\alpha_0\rangle$ or $|\alpha_1\rangle$ respectively [2]. The ket vector also has a corresponding bra vector $\langle\psi|$ which exists in a duel space to the ket. These vectors are related to each other by a Hermitian conjugate according to

$$\langle\psi| = |\psi\rangle^\dagger = \langle\alpha_0|\, a_0^* + \langle\alpha_1|\, a_1^*.$$

The amplitudes and the basis that the system is considered in can be changed by acting on the system with operators. As mentioned, operators are equivalently described as linear transformations and can be represented as matrices. For example, an operator can be represented as the matrix B and it transforms the state $|\psi\rangle$ into another basis defined by the orthogonal vectors $\{|\beta_i\rangle\}$ as

$$B|\psi\rangle = b_0 |\beta_0\rangle + b_1 |\beta_1\rangle \,. \tag{2.2}$$

The probability of observing a particular outcome can be obtained by performing a measurement operation on the state. In this thesis, a specific case of the general quantum measurement will be considered called projective measurements, which can be associated with an observable A that is most conveniently defined in the same choice of basis used to define the state under consideration. A can be decomposed by spectral decomposition into a sum of projection operators in its eigenspace according to

$$A = \sum_i a_i |\alpha_i\rangle\langle\alpha_i| \,,$$

where $|\alpha_i\rangle\langle\alpha_i|$ is a projection operator and a_i are the eigenvalues of A [2]. The projective measurement can then be performed to obtain the probability of finding $|\psi\rangle$ in a particular state by calculating the expectation value of the corresponding projector according to

$$\langle\psi|\alpha_i\rangle \langle\alpha_i|\psi\rangle = |\langle\alpha_i|\psi\rangle|^2 = |a_i|^2 = p_\psi(a_i). \tag{2.3}$$

Considering that the amplitudes are related to probabilities, it can be discerned that the system must be normalized, which means that the length of the vector must be 1, by taking the scalar product of the ket state with its corresponding bra vector

$$\langle\psi|\psi\rangle = |a_0|^2 + |a_1|^2 = 1, \tag{2.4}$$

since it would otherwise lead to unrealistic results. An example of a prevalently used pair of basis states is the vectors defined as

$$|0\rangle \equiv \begin{pmatrix} 1 \\ 0 \end{pmatrix} \quad \text{and} \quad |1\rangle \equiv \begin{pmatrix} 0 \\ 1 \end{pmatrix}, \tag{2.5}$$

which are commonly referred to as the computational basis [2]. Two examples of valid states that exist in a superposition of the computational basis states are the plus- and minus-state, which are constructed as

$$|+\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 \\ 1 \end{pmatrix} \quad \text{and} \quad |-\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle) = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 \\ -1 \end{pmatrix}. \tag{2.6}$$

An interesting remark regarding these states is that it will be equally likely to project onto $|0\rangle$ or $|1\rangle$ when measuring either $|+\rangle$ or $|-\rangle$ in the computational basis, regardless of the minus sign in the $|-\rangle$ state. This sign is often referred to as a phase and can be any complex number of the form $e^{i\phi}$, where $\phi \in \mathbb{R}$. In the case of the $|-\rangle$ state, its amplitude is said to have a relative phase, since the amplitudes do not have the same sign, and it is a basis-dependent phenomenon that could be changed depending on the chosen basis [2]. If the whole state had the same phase factor, which could for example look like $e^{i\phi}|-\rangle$, it is instead called a global phase [2]. It can be seen that the global phase will not affect the measurement outcome by performing similar projective measurement as in Eq. (2.3) on the same general system but with a global phase $|\psi_\phi\rangle = e^{\phi i}|\psi\rangle$, which will yield

$$\langle\psi_\phi|\alpha_i\rangle\langle\alpha_i|\psi_\phi\rangle = e^{-\phi i}\langle\psi|\alpha_i\rangle e^{\phi i}\langle\alpha_i|\psi\rangle = |\langle\alpha_i|\psi\rangle|^2 = |a_i|^2 = p_\psi(a_i). \tag{2.7}$$

It can furthermore be noted that the states $|+\rangle$, $|-\rangle$ also form a valid orthonormal basis in themselves, since $\langle+|-\rangle = \langle-|+\rangle = 0$. An important set of operators in quantum physics that will maintain the inner product are unitary operators, which are operators U that satisfy

$$U^\dagger = U^{-1} \implies UU^\dagger = U^\dagger U = I,$$

where I is the identity of the relevant dimension. Furthermore, orthogonal vectors like $|0\rangle$ and $|1\rangle$ can also be associated with a unitary operator of which they are its eigenvectors with different eigenvalues. For the computational basis vectors, the matrix representation of the operator that satisfies this is

$$\sigma^z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}.$$

Similarly, the unitary matrix which has the plus/minus vectors as its eigenvectors is

$$\sigma^x = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}.$$

With a third matrix that can be defined as

$$\sigma^y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix},$$

these three matrices will generate the two-dimensional Pauli group $\mathcal{P}_2$. The Pauli matrices are used extensively in this thesis and are particularly important for describing density matrices and the Bloch vectors, which will be presented in Sections 2.1.2 and 2.1.3 respectively.

2.1.2 Density matrices

Despite the fact that the previously shown ket and bra depiction is one of the most established structures of quantum physics, there exist numerous other formulations that can be advantageous depending on the situation. An example of one such representation, which is mathematically equivalent to the earlier defined rules [2], is the density matrix formulation of quantum physics, which will be used throughout the coming thesis.

When the information about a state is completely known, its density matrix will simply be the corresponding projection operator to that state ket vector. For example, this gives the density matrix of the state $|\psi\rangle$ from Eq. (2.1) as

$$\rho_{\text{pure}} = |\psi\rangle\langle\psi|. \tag{2.8}$$

All the states that were previously presented have density matrices of this form and are called pure states [2]. From this example, the advantages of using the density matrix formulation are not obvious. However, in the case that the knowledge about a state is not fully determined the state is said to be in a mixed state. This can physically happen, for example, when the quantum system interacts with its environment and is affected by noise. For this scenario, the use of the density matrix formulation does have clear benefits as it can greatly simplify calculations. A mixed state density matrix is a probabilistic mixture of pure states and can be constructed as a linear combination of pure states density matrices that act on the same Hilbert space. This can equivalently be considered to be an ensemble of pure states expressed as

$$\rho_{\text{mix}} = \sum_i p_i \rho_i = \sum_i p_i \left|\psi_i\right\rangle\!\left\langle\psi_i\right| = \sum_i \sum_j \sum_k p_i a_j a_k^* \left|\alpha_j\right\rangle\!\left\langle\alpha_k\right| , \tag{2.9}$$

where the constants p_i are the probabilities of finding ρ_{mix} in the pure state $\left|\psi_i\right\rangle\!\left\langle\psi_i\right|$. It can then be noted that both the pure and mixed density matrices have unit trace, since the probabilities p_i must sum to one, and will be Hermitian because $(\left|\psi_i\right\rangle\!\left\langle\psi_i\right|)^\dagger = \left|\psi_i\right\rangle\!\left\langle\psi_i\right|$. The evolution of a density matrix by an operator B is given as

$$\sum_i p_i B \left|\psi_i\right\rangle\!\left\langle\psi_i\right| B^\dagger = B \rho_{\text{mix}} B^\dagger, \tag{2.10}$$

which can be discerned from investigating how the ket vector evolved in Eq. (2.2). Similarly, the projective measurements of density matrices can be recognized as

$$\sum_i p_i \left\langle\psi_i|\alpha_j\right\rangle \left\langle\alpha_j|\psi_i\right\rangle = \text{Tr}\left[\left|\alpha_j\right\rangle\!\left\langle\alpha_j\right| \sum_i p_i \left|\psi_i\right\rangle\!\left\langle\psi_i\right|\right] = \text{Tr}\left[\left|\alpha_j\right\rangle\!\left\langle\alpha_j\right| \rho_{\text{mix}}\right] = p_{\rho_{\text{mix}}}(a_j).$$

Furthermore, a feature that only relates to pure states is that they are idempotent, which means that a pure density matrix squared will be equal to itself, which can be seen from the normalization condition in Eq. (2.4). This characteristic can however be used to ascertain if a state is pure by using a quantity called the purity, which can be calculated by taking the trace of the squared density matrix [2]. In the case of a pure state, the purity will be equal to one

$$\text{Tr}\left[\rho_{\text{pure}}^2\right] = \text{Tr}\left[\left|\psi\right\rangle \left\langle\psi|\psi\right\rangle \left\langle\psi\right|\right] = \text{Tr}\left[\left|\psi\right\rangle\!\left\langle\psi\right|\right] = \text{Tr}\left[\rho_{\text{pure}}\right] = 1, \tag{2.11}$$

and when the state is mixed, the purity will be

$$\text{Tr}\left[\rho_{\text{mix}}^2\right] \leq 1.$$

2.1.3 Bloch sphere & Bloch vectors

We will in this section show that the aforementioned density matrices can be described by a three-dimensional vector, known as a Bloch vector. This representation of the state of a system has the advantage that certain features, such as purity and closeness between states, can be investigated and estimated by simply plotting the corresponding Bloch vector in a unit sphere, which is called the Bloch sphere.

The previously described single system density matrix is in general a 2×2 matrix with complex elements. Such a matrix would have eight degrees of freedom, but because of the constraints that the density matrix needs to be Hermitian and have unit trace, it can be defined by only three real variables [4], [19]. With these restrictions, the most general expression for a density matrix is

$$\rho = \frac{1}{2} \begin{pmatrix} 1 + u_{z,\rho} & u_{x,\rho} - i u_{y,\rho} \\ u_{x,\rho} + i u_{y,\rho} & 1 - u_{z,\rho} \end{pmatrix}, \quad \text{where } u_{x,\rho},\ u_{y,\rho},\ u_{z,\rho} \in \mathbb{R}.$$

Consequently, it is then straightforward to see that the density matrix can be constructed by utilizing the previously mentioned two-dimensional Pauli matrices according to

$$\rho = \frac{1}{2}\left(I + u_{x,\rho}\sigma^x + u_{y,\rho}\sigma^y + u_{z,\rho}\sigma^z\right). \tag{2.12}$$

The elements $u_{x,\rho}$, $u_{y,\rho}$ and $u_{z,\rho}$ naturally lead to the construction of the Bloch vector of the density matrix ρ according to $\vec{u}_\rho = (u_{x,\rho}, u_{y,\rho}, u_{z,\rho})$ [2], which then can be plotted on the Bloch sphere. Examples of some states plotted on the Bloch sphere are shown in Figure 2.1 [2]. The Bloch sphere is often used to visualize two-level systems and it will be utilized throughout this thesis to convey essential arguments regarding the implementation of the MMS algorithm. It can then be shown that the length of the Bloch vector will always satisfy $||\vec{u_\rho}|| \leq 1$, where the Bloch vector of a pure state is equal to one, while a mixed state vector will be located inside of the sphere.

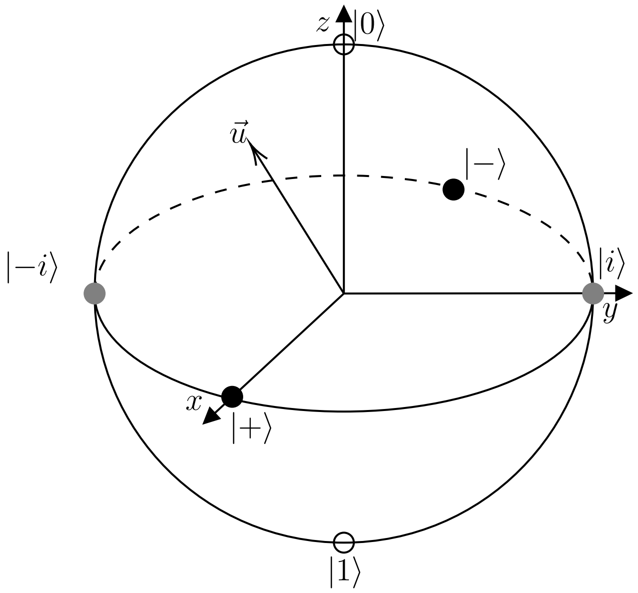


Figure 2.1: The Bloch sphere is depicted along with three different basis states and a Bloch vector $\vec{u}$ of a generic state. Each set of orthogonal basis states corresponds to opposite points along each axis of the Bloch sphere. The hollow dots are located on the z-axis and mark the computational basis states, the filled black dots lie on the x-axis and correspond to the plus/minus state, while filled gray dots are on the y-axis and display the eigenvectors states which are associated to the Pauli-Y matrix.

2.1.4 Multiple systems & entangled systems

All the quantum systems that have been previously investigated were single systems that only consisted of states that existed in the same Hilbert space. However, to study important quantum phenomena like entanglement [20], which appear to be essential to understanding the resources of quantum computers, systems composed of multiple quantum systems from different Hilbert spaces need to be considered. These types of systems will be presented in this section along with the commonly used actions of tracing out and post-selection with examples.

If two general states that are defined in two different Hilbert spaces,

$$|\psi_1\rangle = a\,|0\rangle + b\,|1\rangle \in \mathcal{H}_1 \quad \text{and} \quad |\psi_2\rangle = c\,|0\rangle + d\,|1\rangle \in \mathcal{H}_2,$$

would be considered simultaneously, the resulting composite system of then the individual systems is the tensor product of these,

$$\begin{aligned} |\psi_{12}\rangle &= |\psi_1\rangle \otimes |\psi_2\rangle \\ &= (a\,|0\rangle + b\,|1\rangle) \otimes (c\,|0\rangle + d\,|1\rangle) \\ &= ac\,|0\rangle \otimes |0\rangle + ad\,|0\rangle \otimes |1\rangle + bc\,|1\rangle \otimes |0\rangle + bd\,|1\rangle \otimes |1\rangle \\ &= ac\,|00\rangle + ad\,|01\rangle + bc\,|10\rangle + bd\,|11\rangle\,, \end{aligned}$$

where the total system is defined in the Hilbert space $|\psi_{12}\rangle \in \mathcal{H}_1 \otimes \mathcal{H}_2 = \mathcal{H}_{12}$.

An important phenomenon that can occur when composite systems are considered is when the individual systems become entangled. Two systems are said to become entangled when the two individual systems can no longer be described on their own [2]. In the bra-ket notation, this corresponds to the case when the system cannot be separated into the individual system on a form similar to the state $|\psi_{12}\rangle$, which can be decomposed into $|\psi_1\rangle$ and $|\psi_2\rangle$. However, this is not the case for the state

$$|\psi_{\text{Bell}}\rangle = \frac{1}{\sqrt{2}}\left(|00\rangle + |11\rangle\right). \tag{2.13}$$

This state is one of the Bell states and is a classic example of an entangled state. However, in the density matrix representation, there is an immensely useful operation for the analysis of entangled systems called the partial trace [2]. The partial trace can be used to reduce the composite system to a single system that accurately describes the statistics of the remaining composite system. The partial trace is noted with Tr_i, where the index i references which system is being traced out. When the partial trace is applied on a composite system ρ_{12} with respect to system 2, the residual will then be

$$\text{Tr}_2\left[\rho_{12}\right] = \rho_1.$$

The partial trace is essentially equivalent to performing a measurement on the traced-out system and then discarding the results. This operation could therefore be derived from the previously mentioned measurements of quantum physics. A more direct and interesting example that highlights the usefulness of this operation is when the partial trace is used on a state like the previously mentioned Bell state from Eq. (2.13) that could not be separated into single systems. If the partial trace is applied on the density matrix representation of the Bell state $\rho_{\text{Bell}} = |\psi_{\text{Bell}}\rangle\langle\psi_{\text{Bell}}|$ with respect to the second system, the resulting system would then be

$$\text{Tr}_2\left[\rho_{\text{Bell}}\right] = \sum_{i\in\{0,1\}} \left(I \otimes \langle i|\right)\rho_{\text{Bell}}\left(I \otimes |i\rangle\right) = \frac{1}{2}\left(|0\rangle\langle 0| + |1\rangle\langle 1|\right), \tag{2.14}$$

which is simply a mixed single system. A powerful variation with respect to partial trace that will be used in the implementation of the MMS algorithm is the post-selection measurement, which effectively means that a projective measurement is performed with regards to a specific sought outcome and all the other scenarios are discarded [21]. For example, performing a post-selection measurement on the second system on ρ_{Bell} with regards to the $|0\rangle$ state would result in

$$\left(I \otimes \langle 0|\right)\rho_{\text{Bell}}\left(I \otimes |0\rangle\right) = \frac{1}{2}|0\rangle\langle 0|\,. \tag{2.15}$$

Evidently, this state needs to be normalized, which can be done by simply dividing with the trace, which yields the resulting system

$$\frac{\frac{1}{2}|0\rangle\langle 0|}{\text{Tr}\left[\frac{1}{2}|0\rangle\langle 0|\right]} = \frac{\frac{1}{2}|0\rangle\langle 0|}{\frac{1}{2}} = |0\rangle\langle 0|\,. \tag{2.16}$$

2.1.5 Clifford group

In addition to the composite systems that describe multiple single systems simultaneously, there are also operators that can act on several systems at the same time. An important group of operators that can be specified for any number of systems is the Clifford group, which will be described in this section and is defined in Eq. (2.17). The properties of the Clifford group are essential to a multitude of areas in the field of quantum computing, for example, they are often utilized in quantum error correction [22], [23]. In this thesis, the Clifford group operators will perform a crucial function in the implemented algorithms.

For i number of systems in a composite system, the Clifford group $\mathcal{C}_i$ is the set of unitary operators that normalizes the Pauli matrices, which is often noted as $\mathcal{N}(\mathcal{P}_i)$, and this means that they map Pauli matrices to Pauli matrices under conjugation. This can be written as

$$\mathcal{C}_i = \mathcal{N}(\mathcal{P}_i) = \{C \in U(2^i) | C\sigma^p C^\dagger \in \mathcal{P}_i,\, \forall \sigma^p \in \mathcal{P}_i\}. \tag{2.17}$$

However, by disregarding the global phases, the Clifford group can be defined in terms of the quotient group $\mathcal{N}(\mathcal{P}_i)/\mathcal{U}(1)$, where $\mathcal{U}(1)$ in this case denotes the complex numbers that have absolute value equal to one [19], [24]. It is this abbreviated specification of the Clifford group that will be used throughout this thesis, since the global phases will not affect the results from the algorithm, as can be seen from the calculations in Eq. (2.7). Following the notation of Ref. [22], this group will be called the Clifford quotient group and is noted as $\mathcal{C}'_i$. The number of elements in this quotient definition for i states will be given by [24]

$$\#\text{ elements in } \mathcal{C}'_i = 2^{i^2+2i} \prod_{j=1}^{i} \left(4^j - 1\right). \tag{2.18}$$

For instance, the single system Clifford quotient group will contain twenty-four elements. An example of the elements that can constitute the complete group $\mathcal{C}'_1$ is summarized in Table E.1 of Appendix E.

The Clifford operators are an essential component in the implementation of the magic state distillation algorithm by Bravyi & Kitaev and the modular magic synthesis algorithm, which makes them integral to this thesis. A practically significant aspect of the Clifford operations, which emphasizes the importance of algorithms that are based on them, is that Clifford operations can be fault-tolerantly realized by implementing the algorithm on a quantum computer using, for example, stabilizer codes [25]. Additionally, in this thesis, we will utilize the fact that the Clifford operators can be associated with a binary matrix, called a tableau, to generate the group $\mathcal{C}'_2$ [26], which is later described in Chapter 3.

2.2 Quantum computing

A quantum computer is a computational device that can exploit the aforementioned principles of quantum physics to perform its computations. By utilizing concepts like superposition [27] and entanglement [20], quantum computers are theorized to accomplish some calculations more efficiently than a classical computer [7], [8]. In this thesis, we will focus on how these systems can be used to perform calculations, rather than describing a specific physical system.

First, we introduce some commonly used language of quantum computation theory, which mostly is a reformulation of the previously introduced concepts. Later, we also study what constitutes a universal quantum computer with the use of the Gottesman-Knill theorem.

2.2.1 Qubits & quantum gates

In this section, we will cover some of the reformulations of the previously mentioned concepts from Section 2.1 to align with the commonly used language of quantum computers. This is partly done by giving some examples of how these attributes are implemented and visualized, which will be relevant for the later sections.

Quantum computers use quantum systems to conduct their objective, and like the previously mentioned wave function, these systems can be either discrete or continuous. In this thesis, only discrete two-level systems will be considered and in this context, these systems are called qubits. Qubits constitute the fundamental unit of information that is used in the quantum computer, which makes them the quantum equivalent of the classical bit. This two-level system can in general be described by Eq. (2.1) and is only considered in this thesis as an abstract mathematical object that can interact with other systems and can be transformed by operators [2].

By manipulating these qubits, the quantum computer can perform its calculations. As mentioned in the previous section, a general quantum system can be transformed by operators, but in the quantum computing model, these operators are called quantum gates. These are similarly the quantum analog to the logic gates used in classical computers to manipulate their bits. Operators like the three two-level system Pauli matrices can be expressed in the same way as their matrix representation when they are perceived as gates. Two other examples of important gates that will be used in this thesis are the Hadamard gate H and the phase gate S. These gates can for example be used to generate the single-qubit Clifford group and have the matrix representations

$$H \equiv \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \quad \text{and} \quad S \equiv \begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix}.$$

It can be seen that the Hadamard gate can change a state in the computational basis to the plus/minus basis, similar to the calculation in Eq. (2.2), by investigating how it acts on the $|0\rangle$ state,

$$H|0\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) = |+\rangle . \tag{2.19}$$

As might be discernible from the name, the phase gate can be used to add a phase to a system. The phase gate can also be applied two times on the $|+\rangle$ state to add the relative phase between its terms to transform it into the $|-\rangle$ state,

$$S^2|+\rangle = \frac{S}{\sqrt{2}}(|0\rangle + i|1\rangle) = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle) = |-\rangle . \tag{2.20}$$

Additionally, we can also notice from the definition of S that $S^2 = \sigma^z$.

2.2.2 Quantum circuit model

In quantum computing theory, transformations like Eqs. (2.19) and (2.20) are commonly portrayed as quantum circuits, which are introduced in this section. This will be done by giving simple examples of circuits that are the corresponding analog to some previously mentioned transformations or can be used to obtain some previously mentioned systems.

Quantum circuits are a convenient illustration that shows the evolution sequence of the qubit along a line, often referred to as a wire, where the gates that transform the state are depicted as boxes that the state encounters [2], [8]. The transformation in Eq. (2.19) could equally be visualized with the quantum circuit in Figure 2.2.

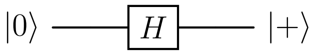


Figure 2.2: The quantum circuit which depicts the evolution of the input state $|0\rangle$ from Eq. (2.19).

Similar to the previously described quantum systems, the measurement of a qubit would make it collapse into one of its basis states. This can also be visualized by using quantum circuits. For example, performing a measurement in the computational basis, which is also referred to as the z-basis, of the resulting $|+\rangle$ system from Figure 2.2 will collapse the state into $|0\rangle$ or $|1\rangle$ with equal probabilities. The quantum circuit for this operation is shown in Figure 2.3.

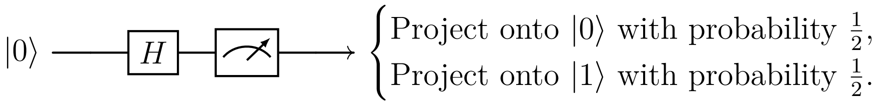


Figure 2.3: An extension of the circuit in Figure 2.2 where the measurement of the output state also is included.

The circuit model visualization scheme can also take into account the case when multiple qubits and operators that act on several qubits are considered [2]. One important multi-qubit gate is the $CNOT$ gate, which is defined by choosing a control qubit and a target qubit, and its function is to apply a Pauli-X operator on the target qubit if the control qubit is in the $|1\rangle$ state. The $CNOT$ gate could be denoted as $CNOT_{ij}$, where the i^{th} qubit is the control and the j^{th} qubit is the target. $CNOT$ in addition to the H and S gates will generate the two-qubit Clifford group. The $CNOT_{01}$ gate can have the matrix representation

$$CNOT_{01} = |0\rangle\langle 0| \otimes I + |1\rangle\langle 1| \otimes \sigma^x = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix},$$

where it can be noted that we have followed the common convention of naming the upmost wire the 0^{th} wire. The $CNOT$ gate can be used to mediate entanglement, as can for example be seen in the circuit in Figure 2.4, where it is utilized to obtain the Bell state in Eq. (2.13) from two $|0\rangle$ states.

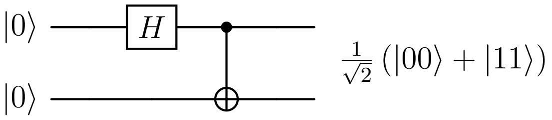


Figure 2.4: The circuit that shows how the Hadamard and $CNOT$ gates can be used to create the Bell state from Eq. (2.13).

These quantum circuits will be used throughout the thesis to show how the studied algorithms are implemented and illustrate calculations.

2.2.3 Universality & the Gottesman-Knill theorem

We will start this section by presenting some observations of properties, which can be mediated with the gates that were previously mentioned, that a universal quantum computer would require to perform any imaginable unitary transformation up to any desired accuracy [2]. Then, these properties will be discussed with regard to the seminal Gottesman-Knill theorem.

The question of what constitutes a universal quantum computer is a well-researched question. In the previous section, it was shown that the circuit model could be used to obtain superposition, add phases, and create entanglement between different states, which would be necessary attributes for a quantum computer to produce any desired state. Particularly, entanglement [20] and superposition [27] seem to be important aspects that hint at the resources that are required to create a universal quantum computer that can perform some calculations more efficiently than a classic computer.

However, an important result in relation to this question is the Gottesman-Knill theorem, which states that a quantum computing scheme that can

i) prepare states in the computational basis,

ii) use operations from the Clifford group (including operations that depend on the previous measurement outcome),

iii) measure the output in the computational basis,

can efficiently be simulated by a classical computer. This implies that these actions are not enough to build a universal quantum computer [2]. These actions will henceforth be referred to as $\mathcal{A}_{\text{GK}}$. By considering that the Clifford operations can achieve the three attributes that were previously mentioned at the beginning of this section, this theorem states that these properties are not enough for a quantum computer to be universal.

Using only the restricted actions in $\mathcal{A}_{\text{GK}}$, it is only possible to obtain a particular set of states called the stabilizer states, which are states that are invariant under the action of the Pauli operators. The computational basis states $|0\rangle$ and $|1\rangle$ are examples of stabilizer states since $\sigma^z |0\rangle = |0\rangle$ and $\sigma^x \sigma^z \sigma^x |1\rangle = |1\rangle$. These states will correspond to the vertices of the octahedron, which contains all the Bloch vectors that satisfy $|u_x| + |u_y| + |u_z| \leq 1$, in Figure 2.6. All the states in the octahedron can be obtained with these actions by constructing convex linear combinations of the Pauli eigenstates and tossing an adequately weighted coin which has probabilities that correspond to the sought state [1], [28].

However, it has previously been shown the set of actions in $\mathcal{A}_{\text{GK}}$ can be promoted to constitute a universal quantum computer quite simply by adding almost any other unitary gate outside of the Clifford group [2]. For example, by adding the general phase shift gate $\Sigma(\phi)$, which can be defined as

$$\Sigma(\phi) = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\phi} \end{pmatrix}, \tag{2.21}$$

where the phase ϕ is not a multiple of $\frac{\pi}{2}$ since that will give back the S gate, would promote the set of actions to delegate a universal quantum computer.

2.3 Magic states & and magic state distillation

We will in this section present the magic states, which is the most important set of states for this thesis, and demonstrate why they are desirable. Additionally, we will also give a background to the established method of obtaining magic states called magic states distillation which was originally proposed by Sergey Bravyi and Alexei Kitaev in Ref. [1].

2.3.1 Magic states

We will start by introducing magic states and show how they are useful with an example. Then the magic states that have been used in the coming study of the modular magic synthesis algorithm will also be defined.

As mentioned in the previous section, some unitary gate outside of the Clifford group needs to be added to $\mathcal{A}_{\mathrm{GK}}$ for it to constitute a universal quantum computer. However, there is an alternative, but equivalent, solution where the quantum computer is instead supplemented with a single qubit state that can be consumed by the algorithm to apply a gate outside of the Clifford group. This would then promote a quantum computer that only has access to $\mathcal{A}_{\mathrm{GK}}$ to become universal [1]. These special states are called magic states, and they are the main focus of this thesis.

To illustrate how these magic states can mediate a non-Clifford gate, we can follow the example which is given in Ref. [1], where they show that the state

$$|\psi_\phi\rangle = \frac{1}{\sqrt{2}}\left(|0\rangle + e^{i\phi}|1\rangle\right)$$

can be used to apply the general phase shift gate $\Sigma(\phi)$ from Eq. (2.21) to another qubit in the composite system of the quantum computer. This would then similarly promote the $\mathcal{A}_{\mathrm{GK}}$ actions to be enough to simulate a universal quantum computer. The objective for this example is to apply the $\Sigma(\phi)$ gate to the general state $|\psi_1\rangle$, which was previously defined as $|\psi_1\rangle = a|0\rangle + b|1\rangle$. This can be done by considering the circuit shown in Figure 2.5, where the input composite system is initialized as

$$|\psi_{\mathrm{in}}\rangle = |\psi_1\rangle \otimes |\psi_\phi\rangle = \frac{1}{\sqrt{2}}\left(a|00\rangle + ae^{i\phi}|01\rangle + b|10\rangle + be^{i\phi}|11\rangle\right).$$

First, the stabilizer measurement $\sigma^z \otimes \sigma^z$, which is an operation that can give the relation between states without collapsing the system, is applied to the state $|\psi_{\mathrm{in}}\rangle$. This measurement will either yield the result +1, which implies that the state is $|\psi_{\mathrm{I}}(+1)\rangle = \frac{1}{\sqrt{2}}\left(a|00\rangle + be^{i\phi}|11\rangle\right)$, or -1, which means that the state is $|\psi_{\mathrm{I}}(-1)\rangle = \frac{1}{\sqrt{2}}\left(ae^{i\phi}|01\rangle + b|10\rangle\right)$. These measurements are obtained with an equal probability of $\frac{1}{2}$. Applying a $CNOT_{01}$ will then produce the system $|\psi_{\mathrm{II}}(+1)\rangle = \frac{1}{\sqrt{2}}\left(a|00\rangle + be^{i\phi}|10\rangle\right)$ or $|\psi_{\mathrm{II}}(+1)\rangle = \frac{1}{\sqrt{2}}\left(ae^{i\phi}|01\rangle + b|11\rangle\right)$, dependent on the outcome of the previous stabilizer measurement. By tracing out the second qubit, the possible results will thus be $|\psi_{\mathrm{III}}(+1)\rangle = a|0\rangle + be^{i\phi}|1\rangle$ or $|\psi_{\mathrm{III}}(-1)\rangle = ae^{i\phi}|0\rangle + b|1\rangle$.

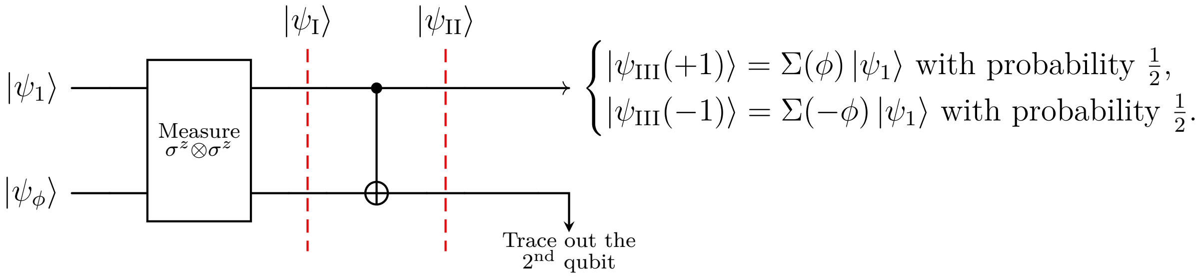


Figure 2.5: A circuit which depicts how a state $|\psi_\phi\rangle$ can be utilized to apply a $\Sigma(\phi)$ gate on the first qubit $|\psi_1\rangle$.

By applying the procedure in Figure 2.5 multiple times and discarding the results when the stabilizer measurement gives -1, the circuit would eventually yield the desired result of applying a $\Sigma(\phi)$ gate to the first qubit. Thus, if the algorithm can use actions from $\mathcal{A}_{\mathrm{GK}}$ and have access to a state similar to $|\psi_\phi\rangle$ that can mediate a non-Clifford group operator, the algorithm would simulate a UQC.

Two kinds of magic states that similarly to $|\psi_\phi\rangle$ can be spent to produce a non-Clifford gate are the H-type and T-type magic states, which will both be studied throughout the rest of the thesis. There are in total twelve H-type states and eight T-type states, which are all plotted on the Bloch sphere in Figure 2.6. The magic states that are of the same set are related to each other by single qubit Clifford operations. A ket representation of paradigmatic H- and T-type magic states could for example be given as

$$\begin{aligned} |H_0\rangle &= \cos\left(\frac{\pi}{8}\right)|0\rangle + i\sin\left(\frac{\pi}{8}\right)|1\rangle\,, \\ |T_0\rangle &= \cos(\beta)\,|0\rangle + e^{i\frac{\pi}{4}}\sin(\beta)\,|1\rangle\,, \quad \text{where } \cos(2\beta) = \frac{1}{\sqrt{3}}. \end{aligned}$$

The states were given the name of H and T because they are eigenvectors to the Hadamard gate and a Clifford gate that was previously called the T gate in [1]. However, we have chosen to name this Clifford operation L, such that it is not confused with the gate that applies a $\frac{\pi}{8}$ phase, which is commonly called the T gate [2]. The L gate is defined as

$$L \equiv e^{i\frac{\pi}{4}} SH = \frac{e^{i\frac{\pi}{4}}}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ i & -i \end{pmatrix}.$$

2.3.2 Magic state distillation

In this section, we will give a summary of the magic state distillation algorithm by Bravyi & Kitaev and cover the main results that they obtained from analytically investigating their methods. This protocol was the original inspiration for the algorithm that is studied in this thesis and it will be an important comparison to assess the results from the MMS algorithm.

It was shown in the previous section that magic states, such as H- and T-type magic states, can be used to implement a non-Clifford gate, which would promote the set of actions in $\mathcal{A}_{\mathrm{GK}}$ to be enough for simulating UQC. However, the natural question then becomes, are there any other states that would accomplish the same objective? In the pioneering paper by Sergey Bravyi and Alexei Kitaev, they ask a similar question and present a method called magic state distillation (MSD) which can consume several slightly faulty magic states to create a magic

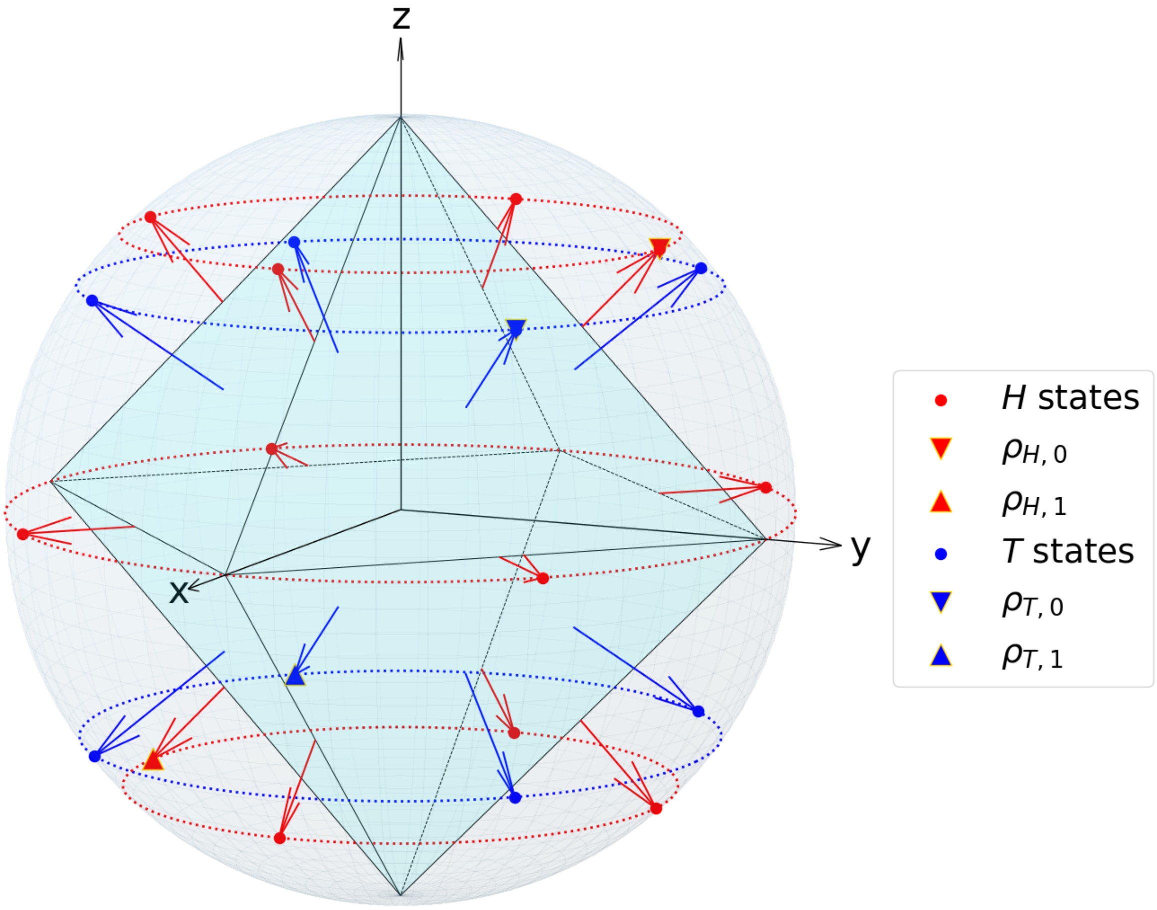


Figure 2.6: The twelve H states and the eight T states are plotted on the Bloch sphere alongside the octahedron that encompasses all the states which are defined by Bloch vector elements that satisfy the condition $|u_x| + |u_y| + |u_z| \leq 1$. The magic states that are marked with down-pointing triangles are the states that will be used as target states for the analysis of the MMS algorithm, which is described in Section 3.1, while the states denoted with up-pointing triangles are the orthogonal states, which point in the opposite direction, to the target magic states.

state that is closer to the target magic state [1], [29]. Two different, yet related, methods for T- and H-type distillation were derived with the results showing that these algorithms could produce magic states arbitrarily close to the target state from a number of non-stabilizer states that were adequately close to the target state.

Among the two distillation methods that were presented by Bravyi & Kitaev, it is the T-type distillation that bears the closest resemblance to the implemented MMS algorithm of this thesis. T-type distillation is carried out by only using actions from $\mathcal{A}_{\mathrm{GK}}$ and preparing input state that are mixed magic state. The algorithm itself has the general structure:

1) five identical input states are first initialized as mixed states of the target T state and its orthogonal counterpart,
2) four stabilizer measurements with cyclical symmetry are then applied from a five-qubit error correction code [30], [31],
3) If the measurements yield $+1$, the round was successful and the system becomes decoded by five qubit Clifford operations, and the result is a single-qubit state that is closer to the target state.

The mixedness of the input state is categorized with a parameter called error probability and will be noted as ε throughout this thesis. A related parameter that was utilized in Ref. [1] and will be employed in this thesis to assess the quality of an output state to the target magic state is fidelity, which is a general measure of the proximity between two states. The fidelity between two arbitrary states is defined as

$$F(\eta,\rho) \equiv \left(\mathrm{Tr}\left[\sqrt{\sqrt{\rho}\eta\sqrt{\rho}}\right]\right)^2, \tag{2.22}$$

for two arbitrary density matrices ρ and η. However, because the output state always will be compared to a pure magic state the expression in Eq. (2.22) can be simplified to

$$F(\eta,\rho) = \frac{1}{2}\left(1+\vec{u}_\eta\cdot\vec{u}_\rho\right), \tag{2.23}$$

where $\vec{u}_\eta$ and $\vec{u}_\rho$ are the Bloch vectors defined from Eq. (2.12). In Appendix A, we have collected derivations and mathematical arguments that are used in the thesis, where for example we perform the derivation from Eq. (2.22) to Eq. (2.23) in Appendix A.1. Furthermore, it is also shown in Appendix A.2 that the fidelity can be related to the error probability of a mixed state that is in an ensemble of the target state and its orthogonal counterpart according to

$$F\left(\eta,\rho\right) = 1-\varepsilon. \tag{2.24}$$

By utilizing the method which is described above, Bravyi & Kitaev analytically obtained an output fidelity function for T-type distillation which depends on the input error probability according to

$$F_{\mathrm{BK},T}(\varepsilon) = 1-\frac{\varepsilon^5+5(1-\varepsilon)^3\varepsilon^2}{(1-\varepsilon)^5+5(1-\varepsilon)^3\varepsilon^2+5(1-\varepsilon)^2\varepsilon^3+\varepsilon^5}. \tag{2.25}$$

The corresponding probability to obtain the state which gives this output fidelity is given as

$$p_{\mathrm{BK},T}(\varepsilon) = \frac{\varepsilon^5+5(1-\varepsilon)^3\varepsilon^2+5(1-\varepsilon)^2\varepsilon^3+(1-\varepsilon)^5}{6}. \tag{2.26}$$

Eqs. (2.25) and (2.26) are also plotted in Figure 2.7.

The distillation method for the H state is generally the same as the described procedure, but it does have some important differences. For example, the Bravyi & Kitaev H-type distillation requires fifteen input states per round. The output fidelity curve that they similarly obtained analytically for this method was

$$F_{\mathrm{BK},H}(\varepsilon) = 1-\frac{1-15(1-2\varepsilon)^7+15(1-2\varepsilon)^8-(1-2\varepsilon)^{15}}{2(1+15(1-2\varepsilon)^8)}, \tag{2.27}$$

and the corresponding probability was

$$p_{\mathrm{BK},H}(\varepsilon) = \frac{1+15(1-2\varepsilon)^8}{16}, \tag{2.28}$$

which are also plotted in Figure 2.8.

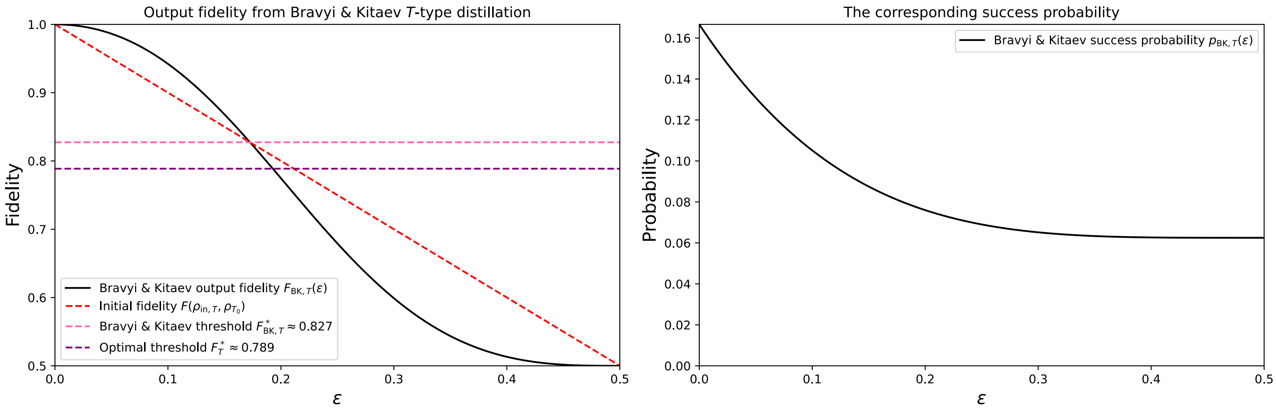


Figure 2.7: The left plot shows the fidelity curve from the Bravyi & Kitaev T-type magic state distillation, alongside the threshold fidelity for that distillation process and the theoretical optimal threshold fidelity, while the right plot displays the corresponding success probability curve. The equations for these curves are shown in Eqs. (2.25) and (2.26).

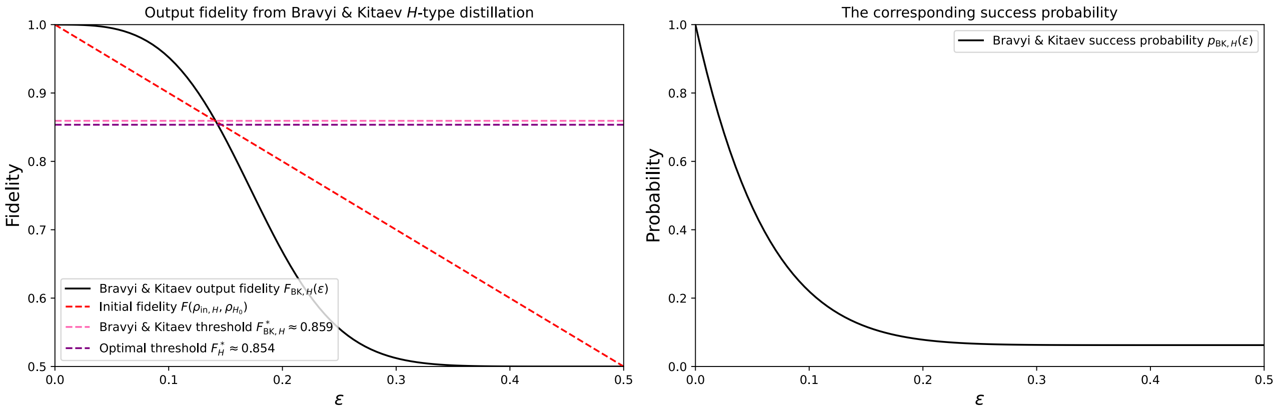


Figure 2.8: Similar to Figure 2.7, these plots show the output fidelity and success probability curves that Bravyi & Kitaev obtained, but this figure corresponds to the H-type distillation process. The equations for the fidelity and its corresponding success probability are shown in Eqs. (2.27) and (2.28).

It can be seen that the fidelity plots for these results share a similar structure, where the output fidelity curve appears to be above the initial fidelity line when the input fidelity is higher than a certain threshold. This implies that their protocol only can produce a magic state that is closer to the target if the input state is above this threshold fidelity. The threshold fidelities that Bravyi & Kitaev obtained for the two protocols were equal to

$$F^*_{\mathrm{BK},T}\left(\rho_{\mathrm{in}}, \rho_T\right) \approx 0.827 \quad \text{and} \quad F^*_{\mathrm{BK},H}\left(\rho_{\mathrm{in}}, \rho_H\right) \approx 0.859.$$

If the input state is above this threshold, the output state will have a higher fidelity to the target magic state than the initial state. The fidelity could then be increased arbitrarily close to 1 by continuing to use the output state as input for another round of the algorithm. This procedure is illustrated in Figure 2.9.

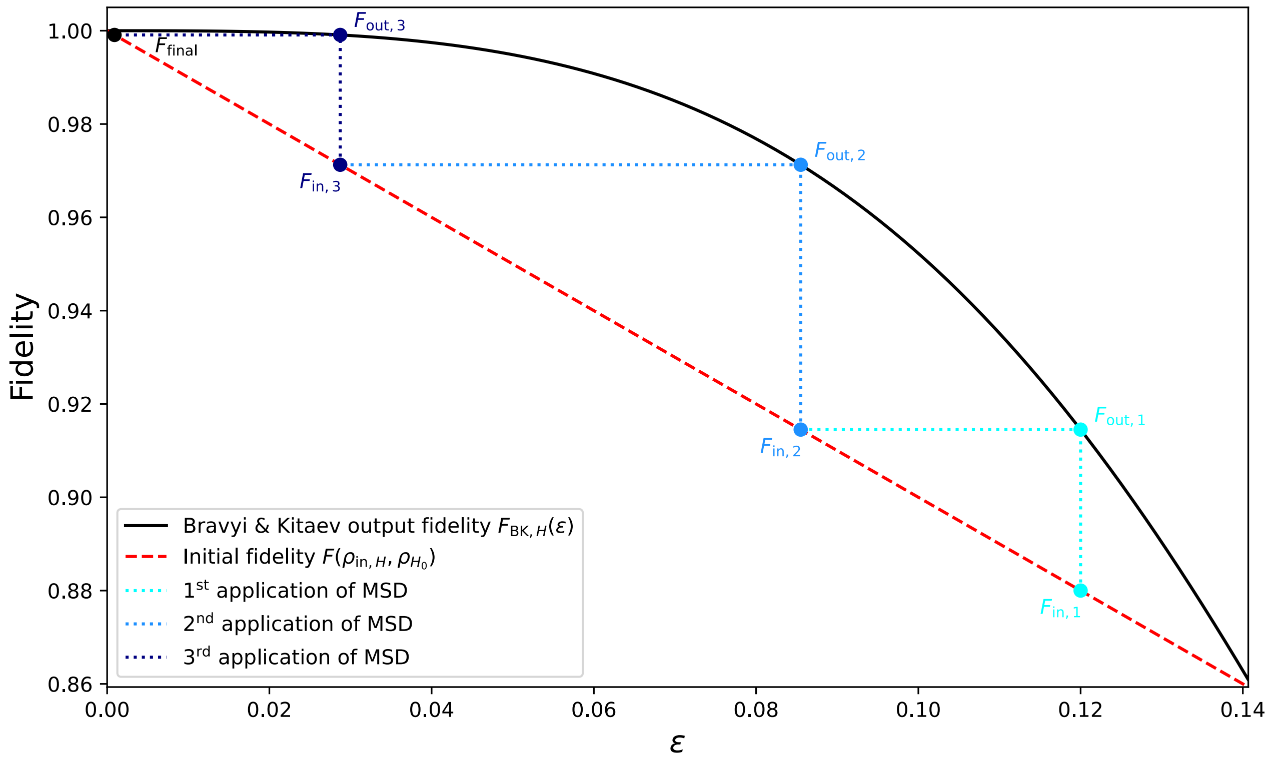


Figure 2.9: This plot visualizes how subsequent use of the Bravyi & Kitaev distillation methods for H-type state can yield a state that is arbitrarily close to the target state. The initial input state for this example was mixed with an error probability $\varepsilon_{\text{in},1} = 0.12$, which corresponded to the input fidelity $F_{\text{in},1}$. After the first application of MSD, the output state had the fidelity $F_{\text{out},1}$, but this could then be further improved by using the output state as input for the second round, where this output state corresponds to the input $F_{\text{in},2}$. The output state can then be improved arbitrarily close to one by continuously using the MSD algorithm until the desired fidelity is achieved.

In addition to the threshold fidelity that Bravyi & Kitaev obtained for their algorithm, another threshold that is called the theoretical optimal threshold for the respective cases is also plotted in Figures 2.7 and 2.8. The optimal threshold fidelities correspond to an initial state that is mixed with an error probability which places the state on the octahedron from Figure 2.6, thus making it a stabilizer state. These initial error probabilities and fidelities are calculated in Appendix A.5. The reason these values are the theoretical optimal threshold stems from the fact that the actions that define the algorithm will exactly be $\mathcal{A}_{\text{GK}}$ if the input is initiated inside of the octahedron, and these are known to be classically simulatable by the Gottesman-Knill theorem. Therefore, it would lead to a contradiction if a stabilizer state could be distilled to a magic state, since that would promote $\mathcal{A}_{\text{GK}}$ to constitute a universal quantum computer.

Other protocols have been proposed that can diminish this void between the octahedron and the threshold fidelities from Bravyi & Kitaev. In Ref. [32], the author investigates a moderately diminished threshold volume on the face of the octahedron. Additionally, in Ref. [32], it was shown that the threshold fidelity for H state distillation could be decreased to the optimal threshold. Despite these improvements, there are still some states that exist outside of the octahedron that cannot be distilled to a magic state [33]. These states are often referred to as bound magic states.

Chapter 3

Methods

We will in this chapter present the modular magic synthesis algorithm and describe its implementation. We will start by first establishing how the input state is prepared and then initiate the description of the algorithm by presenting one optimization round, which then leads to the full description of the algorithm with multiple rounds. We will simultaneously mention the abbreviations that were utilized to ease the simulation of the algorithm. In practice, this algorithm was realized and studied by implementing a code in Python that performed the corresponding calculations. However, this code will not be presented here.

3.1 The modular magic synthesis algorithm

A major goal of this thesis is to implement and study an algorithm that is inspired by the magic state distillation algorithm that was proposed by Bravyi & Kitaev in Ref. [1], which similarly aims to obtain magic states from several faulty input magic states. The implemented algorithm will share many characteristics with MSD, such as only using the same actions, and multiple rounds of optimization to achieve an improved magic state. However, the implemented algorithm will have two defining features that deviate from the Bravyi & Kitaev MSD. These attributes are that the algorithm only requires two qubits as input for each round and that the structure of the operations might vary for different rounds, such that the improvement of the state becomes enhanced to the fullest extent. It is because of this responsiveness that this implemented algorithm will be called Modular Magic Synthesis (MMS). Nonetheless, both MMS and MSD share the mutual aim of producing a state that to the greatest extent achievable approximates the target magic state.

The general implementation of MMS follows the approach of considering all the possible evolution paths that can be created by only using two-qubit Clifford operations from $\mathcal{C}_2'$ for a given input state and then picking the procedure that yields the best resulting state. The following sections will contain the initiation of the input states to MMS and an initial description of one round of optimization, which is then followed by a description of the full algorithm where multiple rounds are considered.

3.1.1 Target magic states and initial input

The initial input state ρ_{in} of the algorithm is the state that will be distilled to approach a target magic state. In this project, the input will mainly be created by applying a mixing operation to a magic state, which is meant to simulate a faulty preparation. A rotated magic state will also be considered as input, but it will only be used with a demonstrative purpose, since this case has already been studied in previous works [32] and is a trivial case that will not yield any interesting results. The onset magic states that both will be used to generate the input state and then defined as the target states in the coming analysis are specifically the H and T states

$$\rho_{H_0} = |H_0\rangle\langle H_0| = \frac{1}{2}\left(I + \frac{1}{\sqrt{2}}\left(\sigma^y + \sigma^z\right)\right), \tag{3.1}$$

$$\rho_{T_0} = |T_0\rangle\langle T_0| = \frac{1}{2}\left(I + \frac{1}{\sqrt{3}}\left(\sigma^x + \sigma^y + \sigma^z\right)\right), \tag{3.2}$$

which will have the corresponding orthogonal states

$$\rho_{H_1} = |H_1\rangle\langle H_1| = \frac{1}{2}\left(I - \frac{1}{\sqrt{2}}\left(\sigma^y + \sigma^z\right)\right),$$

$$\rho_{T_1} = |T_1\rangle\langle T_1| = \frac{1}{2}\left(I - \frac{1}{\sqrt{3}}\left(\sigma^x + \sigma^y + \sigma^z\right)\right).$$

All of these states are plotted in Figure 2.6. A dephasing transformation $D(\rho)$ is applied to each of the inputs before they enter the algorithm, since this will simplify the calculations without diminishing the results. By using the general definition of a state from Eq. (2.9) as an example, the dephasing transformation will initialize the input state in a diagonal superposition

$$D\left(\rho_{\text{in}}\right) = D\left(\sum_i\sum_j\sum_k p_i a_j a_k^* \left|\alpha_j\rangle\langle\alpha_k\right|\right) = \sum_i\sum_j p_i a_j a_j^* \left|\alpha_j\rangle\langle\alpha_j\right|. \tag{3.3}$$

For this thesis, applying the dephasing operator to the input state when it is a mixed H or T state will result in initial states on the form

$$D\left(\rho_{\text{in},H}\right) = \left(1-\varepsilon\right)|H_0\rangle\langle H_0| + \varepsilon\,|H_1\rangle\langle H_1|, \tag{3.4}$$

$$D\left(\rho_{\text{in},T}\right) = \left(1-\varepsilon\right)|T_0\rangle\langle T_0| + \varepsilon\,|T_1\rangle\langle T_1|, \tag{3.5}$$

where ε is the initial error probability. Furthermore, this is the same input that was used by Bravyi & Kitaev.

A rotated input state will simply be created by applying a rotation operation that rotates the magic state around the y-axis according to

$$\rho_{\text{in},H} = R\left(\theta\right)\rho_{H_0}R\left(\theta\right)^\dagger,$$

$$\rho_{\text{in},T} = R\left(\theta\right)\rho_{T_0}R\left(\theta\right)^\dagger,$$

where the rotation operator which rotates the state with an angle θ will be defined as

$$R\left(\theta\right) = \begin{pmatrix} \cos\left(\frac{\theta}{2}\right) & -\sin\left(\frac{\theta}{2}\right) \\ \sin\left(\frac{\theta}{2}\right) & \cos\left(\frac{\theta}{2}\right) \end{pmatrix}.$$

When the initial state was rotated with this operation, we did not apply the dephasing operations before the input state entered the algorithm.

3.1.2 Description of one optimization round

We will now start the description of the algorithm with the simplest case of just one round of optimization, which will then build toward the full description of the algorithm in subsequent sections. The input to one round of optimization is commenced as two identical qubits in a composite system, which can be represented as a tensor product of the two initial inputs as

$$\rho_{\text{in}} \otimes \rho_{\text{in}}.$$

Applying the dephasing operations individually to these states and using the same general expression for the density matrix as in Eq. (2.9) will give the same results that are shown in Eq. (3.3), which will result in

$$\begin{aligned} D\left(\rho_{\text{in}}\right) \otimes D\left(\rho_{\text{in}}\right) &= D\left(\sum_i \sum_j \sum_k p_i a_j a_k^* \left|\alpha_j\right\rangle\!\left\langle\alpha_k\right|\right) \otimes D\left(\sum_m \sum_n \sum_o p_m a_n a_o^* \left|\alpha_n\right\rangle\!\left\langle\alpha_o\right|\right) \\ &= \sum_i \sum_j p_i a_j a_j^* \left|\alpha_j\right\rangle\!\left\langle\alpha_j\right| \otimes \sum_m \sum_n p_m a_n a_n^* \left|\alpha_n\right\rangle\!\left\langle\alpha_n\right|. \end{aligned}$$

This state will then enter the actual algorithm, where they will be evolved by a two-qubit Clifford operation $C \in \mathcal{C}_2'$ according to the evolution in Eq. (2.10), which will give

$$C\left(D\left(\rho_{\text{in}}\right) \otimes D\left(\rho_{\text{in}}\right)\right) C^{\dagger}. \tag{3.6}$$

A post-selection measurement, which is presented in Section 2.1.4, with respect to the $|0\rangle$ state is then applied to the second qubit, which will give an unnormalized single-qubit output state. This procedure can be implemented in a similar manner as was done in the calculation of Eq. (2.15), and this results in

$$\tilde{\rho}_{\text{out}} = \left(I \otimes \langle 0|\right) C\left(D\left(\rho_{\text{in}}\right) \otimes D\left(\rho_{\text{in}}\right)\right) C^{\dagger}\left(I \otimes |0\rangle\right).$$

The output state can then be normalized by dividing with the trace of $\tilde{\rho}_{\text{out}}$. It can also be noted that this trace will be the probability of measuring the $|0\rangle$ state on the second qubit, which will then effectively correspond to the success probability of obtaining the output state. The output state will then give the output of one round of evolution as

$$\rho_{\text{out}} = \frac{\tilde{\rho}_{\text{out}}}{\operatorname{Tr}\left[\tilde{\rho}_{\text{out}}\right]}, \quad \text{where } \operatorname{Tr}\left[\tilde{\rho}_{\text{out}}\right] = p_{\rho_{\text{out}}}(+1). \tag{3.7}$$

One round of this evolution protocol can be visualized as the circuit in Figure 3.1. Additionally, the operations that are performed for one round of evolution will be noted in coming calculations and figures as a Kraus operator K, which will summarize the evolution of an input state as $\rho_{\text{out}} = K\rho_{\text{in}}K^{\dagger}$. Additionally, in Appendix C, we present a slight variation of the MMS algorithm which we have named tracing out modular magic synthesis (TMMS), where we perform a partial trace operation on the second qubit instead of post-selection.

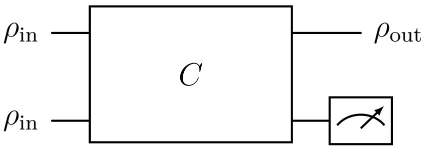


Figure 3.1: This circuit represents one round of evolution of MMS for some input state ρ_{in} and Clifford operation $C \in \mathcal{C}_2'$, which results in the state ρ_{out} after post-selection.

By repeatedly evolving the input state with each individual Clifford operation in $\mathcal{C}_2'$ and comparing the output states to some target magic state, the algorithm can optimize which Clifford operations yield the most beneficial output. The algorithm will utilize fidelity, which is described in Section 2.3.2, to assess the quality of the output state and can thus determine which

Clifford operations are the most advantageous.

From Eq. (2.18), it can be seen that there will be 11520 individual Clifford operations in $\mathcal{C}_2'$ that need to be considered, which consequently implies the possibility that there will be equally many output states. The Clifford operators in the group $\mathcal{C}_2'$ were practically generated in the Python code by generating all the valid tableaux and then utilizing the Qiskit [34] Clifford function to convert the tableaux into Clifford operators [26].

To illustrate how the elements in $\mathcal{C}_2'$ will affect the fidelity between the input state and a target magic state, all the fidelities that can be obtained by using the circuit in Figure 3.1 for the trivial case when the input is a rotated H state are shown in Figure 3.2. It can be seen in this plot that the fidelity for the majority of the output states will become worse by evolving the initial state and the rotation angle needs to be moderately high for the algorithm to see an improvement in the maximum fidelity compared to the fidelity corresponding to doing nothing to the initial state.

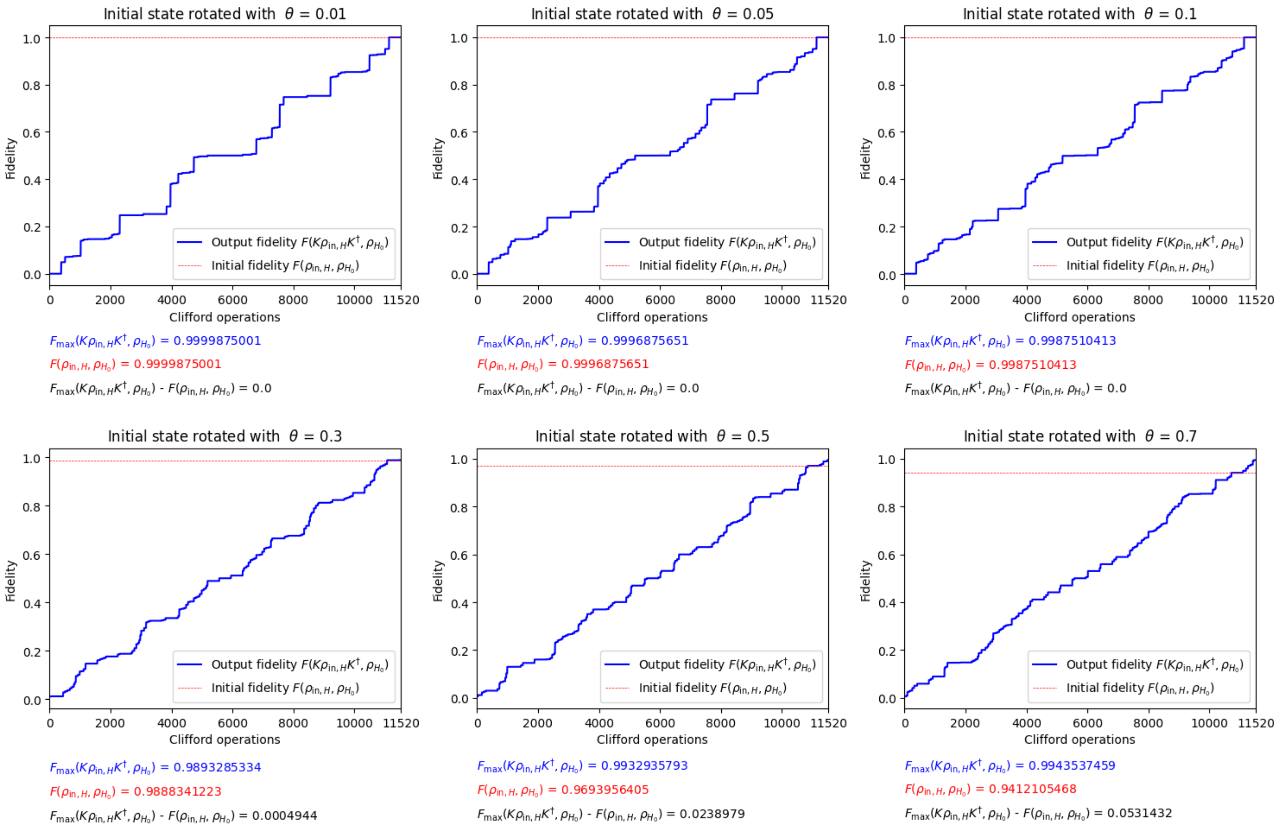


Figure 3.2: All the fidelities between the target H state ρ_{H_0} and output states that can be obtained by evolving the initial state according to the circuit in Figure 3.1 with each individual two-qubit Clifford operation in $\mathcal{C}_2'$. Note that these fidelities have been sorted. The initial state for each of these plots was a y-rotated H state and the six different subplots show the results for six different initial rotation angles. The maximally obtained fidelity, the initial fidelity, and the difference between these are written underneath the plots. All the unique output states that can be obtained for these rotation angles are additionally plotted in Appendix B.1.2, alongside other results that were obtained for the case of an input state that was a rotated H state.

In Appendix B, we present supplementary results that were obtained when investigating the MMS algorithm that can give additional insight into the function of the algorithm. For instance, in Appendix B.1.1 we show an analogous plot of all the sorted output fidelities that could be obtained for a rotated T state and additional results that were produced when the initial state was a rotated T state.

3.1.3 Description of multiple optimization rounds

The fidelity obtained by the circuit in Figure 3.1 can further be improved by implementing multiple optimization rounds of the evolution circuit where the output becomes the input for the next round. To illustrate how this extrapolation could be realized, the circuit for two rounds is shown in Figure 3.3. The algorithm can optimize the final output state from multiple rounds of evolution by investigating every sequence of Clifford operations and choosing the path that creates the best output.

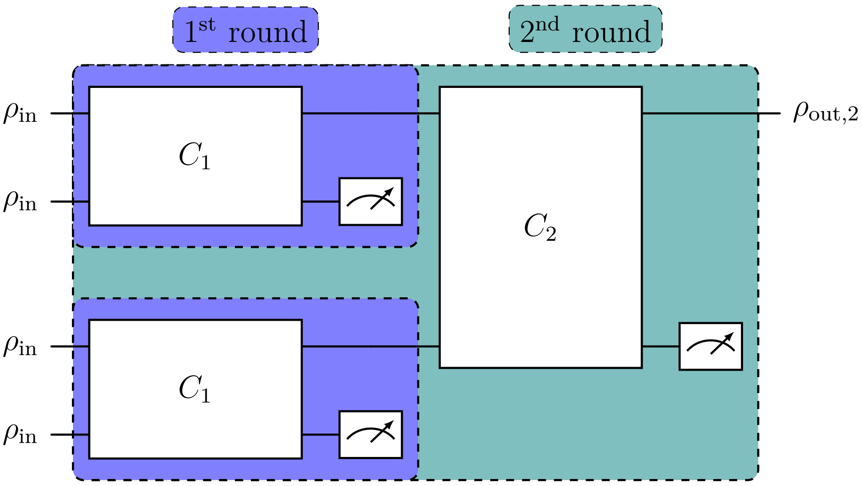


Figure 3.3: This is the general circuit for two rounds of evolution from a given input ρ_{in} with two Clifford operations C_1, $C_2 \in \mathcal{C}_2'$. Both of the two 1^{st} round evolution uses the same Clifford operation C_1 and they will therefore create the same output state from the first round. These two output states will then be evolved with another Clifford operation C_2, which could be the same or different as C_1. After a second post-selection, the final output will be the single state $\rho_{\text{out},2}$.

However, as we mentioned earlier, there will be 11520 individual Clifford operations in $\mathcal{C}_2'$, which implies that there may be equally many possible output states from the first round and each of these will in itself create the same number of outputs in the second round. Since it is not obvious which states will yield a good output state in the coming round, all of these Clifford operations need to be considered for each evolution. This quickly becomes a daunting computational task. However, a majority of the Clifford operations will transform the input to the same output state. This fact will impactfully reduce the number of inputs for the next round that necessitates examination, since there is no meaning in considering the same output more than once.

Additionally, two other symmetries of the output states can be used to further diminish the number of input states that need to be considered for the next round. These reductions are based on the same realization that numerous states can be made identical, and will thus result in the same information, which would make it unnecessary to investigate all of the output states. This simplification will be approached from the Bloch sphere representation, which is presented in Section 2.1.3, since this will make it easier to visualize the reduction. The first abbreviation comes from the realization that the output states will reside all over the Bloch sphere, which can be seen from the results in Figures B.3 and B.6, and this will result in a majority of the states that are not interesting for obtaining an output state that is close to the target. However, as is shown in Appendix A.3, every state on the Bloch sphere can be shifted into the positive octant by only using single-qubit Clifford operations, and it is thus within the allowed operations for this algorithm. Performing these transformations will shift numerous output states to the same positions inside of the positive octant and they will therefore be considered the same output state. Practically, this was implemented by simply taking the absolute value of each element in the Bloch vectors of the output state.

The second action that will be taken to diminish the number of output states that necessitate consideration is to order all the Bloch vector elements of the output states such that the largest element is shifted to the z-component and the smallest to the x-component. The justification for this abbreviation is that the order of the elements will not affect the output fidelity if the output state is inside of the positive octant, which is shown in Appendix A.4. The resulting output Bloch vectors will lie in the upper hexant of the positive octant, which is depicted in Figure A.3. This will further shift a majority of the states to the same position and thus further decreasing the number of output states that need to be considered for the next round. This will make the computations less taxing and ease the study for more optimization rounds. The decrease of distinctive output states from these symmetries has been visualized in Figure 3.4, where all unique output states for the different degrees of simplification have been plotted.

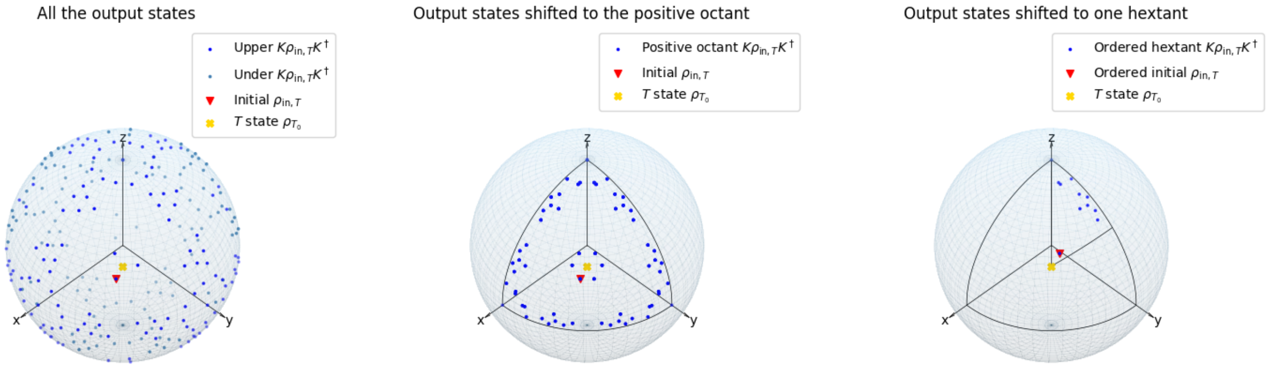


Figure 3.4: These three Bloch spheres showcase the reduction in the number of unique output states that can be achieved by first shifting the states into the positive octant and then further transforming them into a single hextant. The leftmost Bloch sphere shows all the unique output states from one round of optimization when the initial state is a T state that has been y-rotated with an angle of 0.2 radians. The upper points refer to those states that are lying above or on the plane $x + y + z = 0$, whereas the under states denote those states that are below the plane. In total, there are 246 unique output states for one round of optimization. The middle Bloch sphere portrays the resulting states from the same optimization round, but all the states have been shifted to the positive octant. Performing this shift results in 63 unique states. Finally, the rightmost Bloch sphere shows the states when they have been shifted to one particular hextant of the octant where the elements of the Bloch vector are ordered such that the z-element is the largest and the x-element is smallest. This reduces the number of unique output states for this case to only 11.

Utilizing these calculation abbreviations, MMS can further be extended to include more rounds of optimization by using the same logic as going from one round to two rounds. The circuits for more rounds of optimization are additionally shown in Appendix D. We will use up to four rounds of optimization to obtain the main results of this thesis.

As more rounds become implemented, there will be an increasing number of sequences of Clifford operators that yield the same output state. It is therefore important to not only catalog the optimal output fidelities, but also monitor the highest probability of obtaining the corresponding state. The algorithm will do this by first identifying the most advantageous fidelity and then search for what is the highest probability of obtaining that state. The success probability for the first round will simply be the previously mentioned normalization constant from Eq. (3.7), while the probability for higher rounds is calculated by multiplying the probabilities of obtaining the states that yielded the best fidelity.

Chapter 4

Results & discussion

In the coming sections, the most interesting results that were obtained by implementing the MMS algorithm will be presented and discussed. These results are the maximum fidelities and the corresponding probabilities that were obtained for the case of an input state that was initialized as a mixed magic state for both T- and H-type distillation.

4.1 Results

The results for the T-type distillation and H-type distillation were obtained independently and will therefore be presented in separate sections. The maximum fidelity and corresponding probability for these cases were obtained by applying the MMS algorithm to input states that were initiated with different initial error probabilities. These results will be presented with fidelity and probability curves, similar to the Bravyi & Kitaev results in Figures 2.8 and 2.7, but it is important to note that only discrete fidelity and probability values were obtained and the curves will therefore only be estimated. In Appendix E, we have summarized the initial error probabilities that were used to obtain the curves that are presented in this section, alongside other lists of used input parameters and tables. However, these approximations can still be used to assess the capabilities of the MMS algorithm and will be discussed in the coming section.

4.1.1 T states results

The first results from utilizing the MMS algorithm were obtained for the case of a T-type distillation and are shown in Figure 4.1. The input state was a mixed T state in the form of Eq. (3.5), were the target state was the T state in Eq. (3.2).

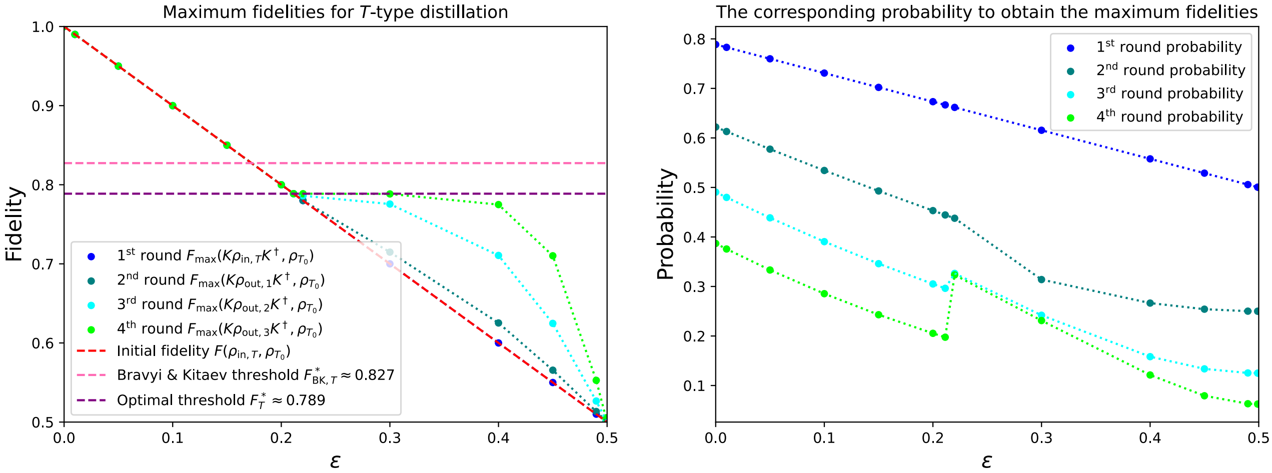


Figure 4.1: The left plot shows the maximum obtained fidelities for four rounds of optimization when the initial state is a mixed T state along side the initial fidelity curve as a red dashed line and the right plot shows the corresponding highest probabilities to obtain the states that give these fidelities. The threshold fidelity that was obtained from the Bravyi & Kitaev MSD and the theoretical optimal threshold fidelity is additionally plotted alongside the fidelity curves [1]. These plots were obtained for fourteen steps of initial error probabilities, where the points are the actual obtained fidelities, while the dotted lines connecting the points have been added for visualization purposes. In Appendix B.2.1, we have provided two examples of all the unique states that can be obtained for this case of an initial mixed T state for up to four rounds of optimization.

4.1.2 H states results

Similar to the previous result, the MMS algorithm was also utilized to investigate the potential of distilling a H state. The target state was chosen to be the H state from Eq. (3.1) and the input state was initialized as a mixed H state on the form of Eq. (3.4). First, the fidelity and probability curves were created for this distillation by choosing initial error probabilities in the range of $0 \leq \varepsilon < 0.5$. These curves are shown in Figure 4.2.

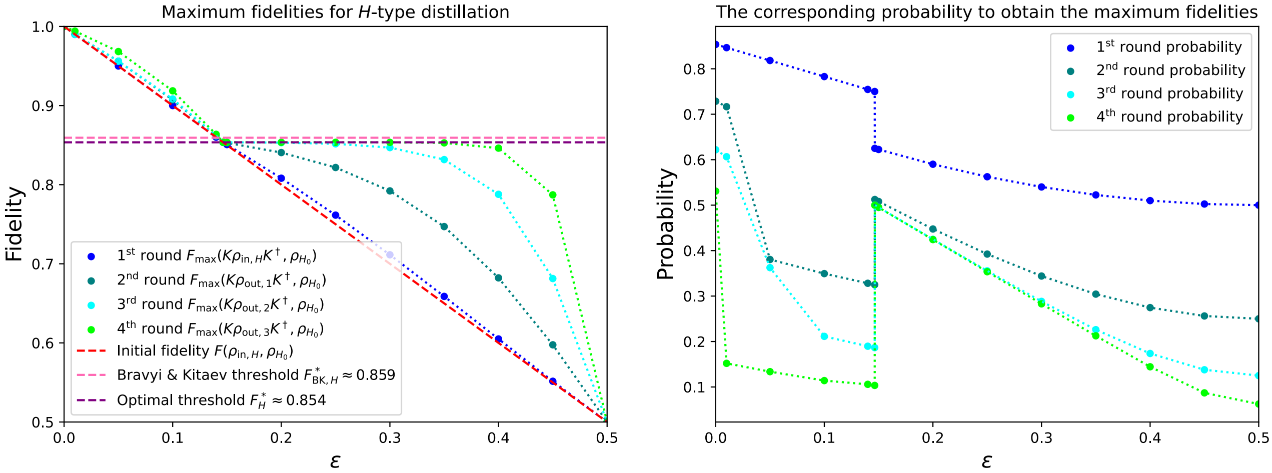


Figure 4.2: In the same vein as the results in Figure 4.1, the left figure depicts the maximally obtained fidelities for four rounds when the initial state was set to be a mixed H state, while the right figure displays the corresponding probabilities to obtain the maximum fidelities. Additionally, the threshold fidelity for the Bravyi & Kitaev MSD algorithm and theoretical optimal threshold fidelity is similarly plotted [1]. Fifteen different initial error probabilities were used to obtain these points. All the unique states that could be obtained after the shift into the same hextant at two different initial error probabilities for up to four rounds of optimization are displayed in Appendix B.2.2.

To further this investigation, the same calculations were performed for the more interesting case of only considering error probabilities that produce input states that are outside of the octahedron in Figure 2.6. This means only using initial error probabilities between zero and the theoretical threshold error probability for this case of H-type distillation, which is calculated in Appendix A.5. These plots are displayed in Figure 4.3.

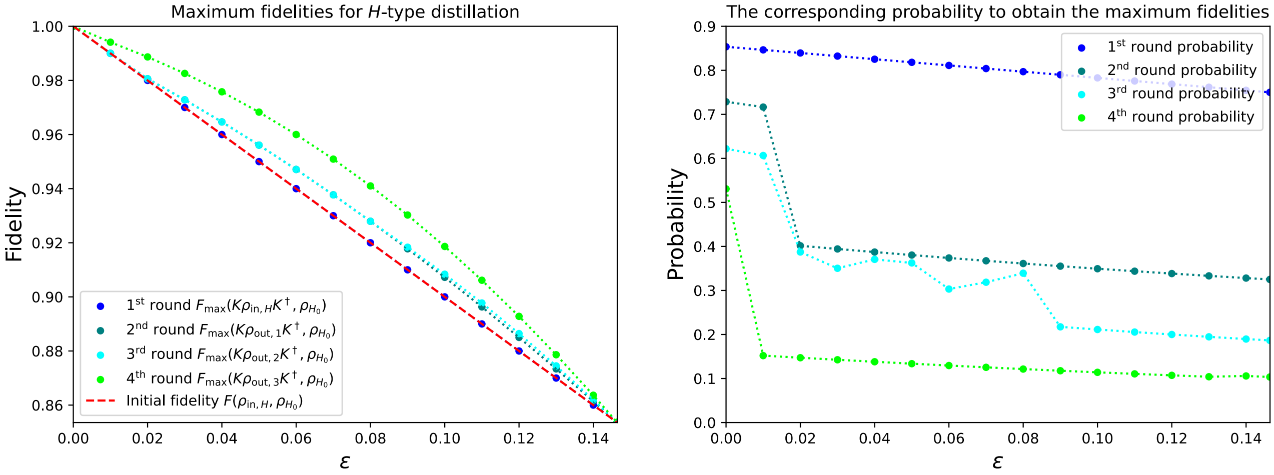


Figure 4.3: These plots were obtained in the same way as the curves in Figure 4.2, but the initial error probabilities have been chosen such that it only creates input states that are outside of the octahedron. Sixteen steps of initial error probabilities within this narrower range were used to obtain these plots.

To fully assess the potential of H-type distillation with the MMS algorithm, the same calculations as the previous plots were performed for even small initial error probabilities, and these results can be seen in Figure 4.4.

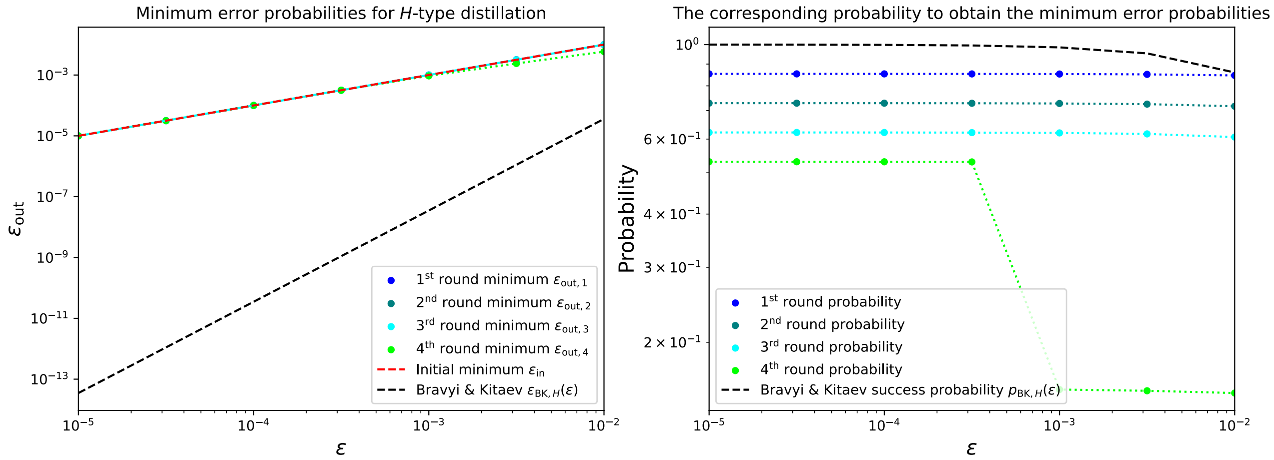


Figure 4.4: Continuing the trend from the previous figures, these plots have been obtained by considering seven initial error probabilities in a further diminished range that have now been reduced to values between 10^{-2} to 10^{-5}. These points were obtained in the same manner as Figures 4.2 and 4.3, but note that it is the minimum output error probability which is shown on the y-axis in the left figure. The relation between the fidelity and output error probability is derived in Appendix A.2 and can be summarized as $\varepsilon_{\mathrm{out}} = 1 - F(\rho_{\mathrm{out}}, \rho_{\mathrm{target}})$ for this case. The initial error probability in this scenario is chosen at such a minute scale that both plots utilize a logarithmic axis to present their content. In addition to the obtained error and success probabilities from the MMS algorithm, we have also plotted the derived Bravyi & Kitaev MSD error probability and success probability expression for the H state distillation, which is shown in Eqs. (2.27) and (2.28) [1] respectively.

4.2 Discussion

By investigating the plots in Figures 4.1 and 4.2, it becomes apparent that the fidelity curves have a change in behavior for a particular initial error probability. Additionally, the probability curves seem to be discontinuous around the same point. This shift will transpire when the error probability corresponds to an input state which is located on the octahedron, which is similar to the structure of the results that were obtained by Bravyi & Kitaev and are shown in Figures 2.7 and 2.8. These error probabilities will be called the threshold error probability ε^*, which corresponds to the previously mentioned optimal threshold fidelity, and they have been calculated in Appendix A.5 for both cases of H- and T-type distillation. The change in the general appearance of the fidelity curve and the discontinuity of the probability plots occurs because the algorithm will find different kinds of solutions for these scenarios. Therefore, the discussion of these scenarios has been divided into separate sections, where we will first consider the case when $\varepsilon > \varepsilon^*$ and then analyze when $\varepsilon < \varepsilon^*$.

4.2.1 Discussion for the case of $\varepsilon > \varepsilon^*$

We will start the discussion with the case when the initial error probability mixes the input state such that it will be inside of the octahedron from Figure 2.6, which is equivalent to saying that the input fidelity is below the optimal threshold F^*. Both the T-type and the H-type distillation will be considered and compared in this section.

When the initial error probability satisfies $\varepsilon > \varepsilon^*$, the input state will be located inside of the octahedron. It can be seen that both fidelity plots in Figures 4.1 and 4.2 have the same general structure for this range. It appears as though the maximum fidelity that can be achieved nears the optimal threshold fidelity with each additional optimization round, but cannot be increased beyond the threshold. More rounds seem to be required to shift the state closer to the hull of the octahedron and attain the threshold fidelity when the error probability increases. Furthermore, it can be observed that the fidelity will be independent of where on the edge, in the case of a H state, or the face, in the case of a T state, of the octahedron that the output state is located. This can be seen by considering that the elements of a state that is on the octahedron will always sum to one.

The probability curves in Figures 4.1 and 4.2 also seem to share a similar structure for this case. One significant feature is that the probabilities appear to converge for error probabilities close to ε^*, which can most clearly be seen for the third and fourth rounds of the probability curves in Figure 4.2. This behavior can be explained by considering how the algorithm is constructed. As was described at the end of Section 3.1.3, the algorithm will first search for the maximum fidelity for each round and then seek the largest probability to obtain the corresponding state. Considering that the fidelity is independent of where the output state lies on the octahedron and that the post-selection is performed by measuring the $|0\rangle$ state, the algorithm will be incentivized to shift the output to the $|0\rangle$ state to in order increase the probability. Since the success probability will approach one when the output state becomes closer to $|0\rangle$, the probabilities will be the same for higher evolution rounds.

By investigating which Clifford operations the algorithm settles on as the most optimal evolution of the input state in the case of H distillation, it becomes apparent that the algorithm consecutively applies the same operation for each round. This Clifford operation will be called C_{5760}, since it was the 5760^{th} number in the list of generated two-qubit Clifford operations. The full evolution that is described in Section 3.1 will then be noted with the Kraus operator K_{5760}.

This is a convention that will be used throughout the rest of the discussion and appendices. Recurringly evolving an input state that has been mixed with an initial error probability of 0.3 will give the chain of output states as

$$\underset{\rho_{\text{in},H}(\varepsilon=0.3)}{\begin{pmatrix}0\\0.28284\\0.28284\end{pmatrix}} \xrightarrow{K_{5760}} \underset{\rho_{\text{out},1}}{\begin{pmatrix}0\\0.07407\\0.52378\end{pmatrix}} \xrightarrow{K_{5760}} \underset{\rho_{\text{out},2}}{\begin{pmatrix}0\\0.00431\\0.82204\end{pmatrix}} \xrightarrow{K_{5760}} \underset{\rho_{\text{out},3}}{\begin{pmatrix}0\\0.00001\\0.98110\end{pmatrix}} \xrightarrow{K_{5760}} \underset{\rho_{\text{out},4}}{\begin{pmatrix}0\\0\\0.99982\end{pmatrix}}.$$

Note that these output states have been shifted into the same hextant as the input state. These evaluations clearly show that the algorithm truly does find an optimal output state that is close to the $|0\rangle$ state. Since the evolution will be the same for each round, it is straightforward to find an analytical expression for a general round. These calculations are performed in Appendix A.6 and the final analytical expression for the fidelity and the probability are plotted in Figure 4.5.

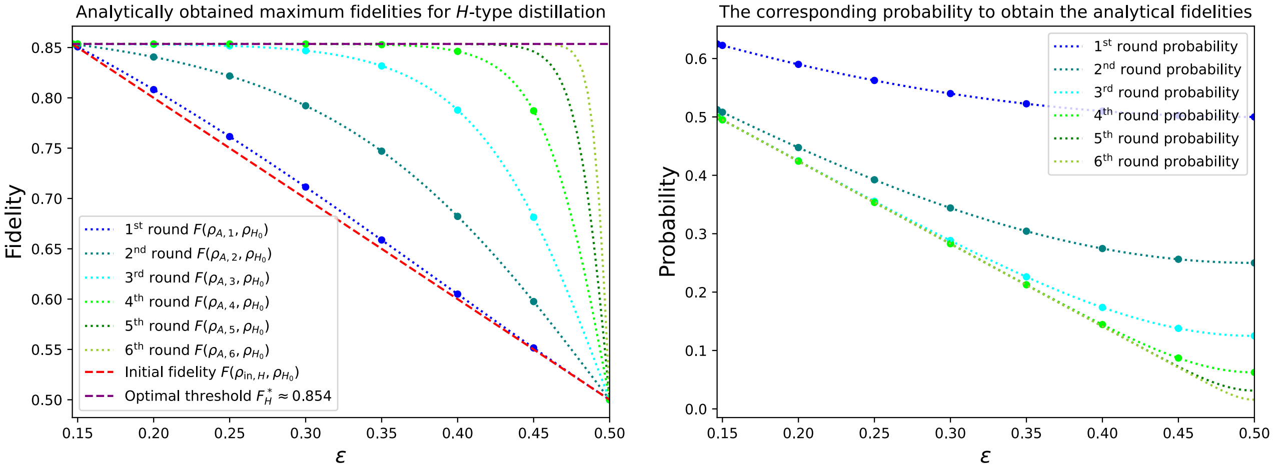


Figure 4.5: These figures depict six rounds of optimization that were obtained with the analytical expression for fidelity from Eq. (A.14) and probability from Eq. (A.11). The analytical results were obtained for one hundred steps of initial error probability between 0.1465 to 0.5 and are plotted with the dotted lines. Alongside the analytical curves, the previously obtained numerical results from Figure 4.2 are also shown with the dots.

In a similar fashion to the case of H-type distillation, the optimal sequence of Clifford operations that the algorithm found was also investigated for the T-type distillation when $\varepsilon > \varepsilon^*$. However, in contrast to the H-type distillation, these results show that the Clifford operations are not the same for every round and do not follow the same simple pattern. For example, when the initial T input state was mixed with an error probability of $\varepsilon = 0.3$, the optimal output states and the corresponding succession Clifford operations were:

$$1^{\text{st}} \text{ optimization round: } K_0\rho_{\text{in},T}K_0^\dagger = \begin{pmatrix}0.23094\\0.23094\\0.23094\end{pmatrix},$$

$$2^{\text{nd}} \text{ optimization round: } K_{5760}K_{128}\rho_{\text{in},T}K_{128}^\dagger K_{5760}^\dagger = \begin{pmatrix}0\\0.00860\\0.73556\end{pmatrix},$$

$$3^{\text{rd}} \text{ optimization round: } K_{5760}K_{5760}K_{128}\rho_{\text{in},T}K_{128}^\dagger K_{5760}^\dagger K_{5760}^\dagger = \begin{pmatrix}0\\0.00005\\0.95462\end{pmatrix},$$

$$4^{\text{th}} \text{ optimization round: } K_{5760}K_{5760}K_{5760}K_{128}\rho_{\text{in},T}K_{128}^\dagger K_{5760}^\dagger K_{5760}^\dagger K_{5760}^\dagger = \begin{pmatrix}0\\0\\0.99892\end{pmatrix}.$$

Despite the fact that the algorithm uses more than just one Clifford operation, there is a clear structure that can be seen from the above states and the two additional examples that are provided in Appendix B.2.1.1 for the cases when initial error probability are $\varepsilon = 0.22$ and $\varepsilon = 0.4$. First, it appears as though the algorithm was not able to increase the fidelity in the first round for all the different initial error probabilities. The most advantageous action that the algorithm could then perform is to apply a Clifford operation that leaves the input invariant. Since the 0th Clifford operation does not transform the input and it is the first operation that the algorithm investigates, it will simply consider it as the most beneficial operation. However, if the algorithm can increase the fidelity, it seems like it will first apply the 128th Clifford operation and then follow up with sequentially applying the 5760th Clifford operation. Furthermore, it can be noted that the algorithm still seems to accomplish the previously mentioned goal of producing an output state that is close to the $|0\rangle$ state, since this will increase the success probability and the fidelity is independent of where on the octahedron the output state is located. Since the sequence of Clifford operations does not follow the same simple pattern as the H state case, the analytical calculations for these curves will become more cumbersome and will therefore be omitted.

It can become apparent why the states that are initiated inside of the octahedron cannot be shifted to a position outside of the octahedron by considering the Gottesman-Knill theorem, which is described in Section 2.2.3. This theorem states that a classical computer can efficiently simulate algorithms that only have access to states inside of the octahedron in Figure 2.6 and is implemented by only using Clifford operations and measurements in the computational basis. Since some of the states outside of the octahedron can be distilled to a magic state, which would promote this set of operators to simulate universal quantum computing, it would create a contradiction if states inside of the octahedron could be transformed into a magic state with these operations.

4.2.2 Discussion for the case of $\varepsilon < \varepsilon^*$

Similar to the previous section, we will now discuss both the results from the H- and T-type distillation, but the focus will instead be on the case when the initial error probability is lower than the threshold error probability, or equivalent when the initial fidelity is higher than the optimal threshold F^*. This case is the more theoretically and practically interesting scenario, since there is now a possibility of obtaining a state that has a higher fidelity to the target magic state than the optimal threshold. In contrast to the previous case when $\varepsilon > \varepsilon^*$, the fidelity plots from Figures 4.1 and 4.2 do not have the same structure when $\varepsilon < \varepsilon^*$. The implications of these different appearances will be one of the main focus points of this discussion.

It can be seen from the fidelity plot in Figure 4.1 that the maximum fidelity curves align with the input fidelity line for this case. This implies that the optimal combination of Clifford operations that the algorithm found was a sequence of operations that leaves the input state invariant. Therefore, it will be impossible to increase the fidelity of the input state to the target T state and it therefore appears as though T-type distillation is unachievable with the MMS algorithm.

Conversely, the maximum fidelity curves in Figure 4.2 seem to be above the initial fidelity line. These resemble the general structure of the results that Bravyi & Kitaev obtained for H-type distillation, which are shown in Figure 2.8. The arc of the fidelity curves can more clearly be seen in Figure 4.3. This means that the algorithm was able to find a set of Clifford operations that could increase the fidelity of the input state to the target H state. The implication of this is that some H-type distillation could be possible by implying a similar procedure that was shown in Figure 2.9. It does appear as if it is feasible to improve the fidelity for all states that are mixed with initial error probabilities that are smaller than ε^*. This agrees with similar results from Ref. [32] where they show that the threshold error probability of their distillation procedure was equal to the theoretical optimal threshold.

To highlight the important feature that the MMS algorithm can change the Clifford operations to optimize the output fidelity depending on the input state, we also investigated the Clifford sequences the algorithm settled on when $\varepsilon < \varepsilon^*$ for H-type distillation. This is done similarly to the investigation of the optimal Clifford operations that were performed for the case when $\varepsilon < \varepsilon^*$ in the previous section. When the initial H state was mixed with an error probability of 0.12, the evolution procedure that the algorithm used was

$$1^{\text{st}} \text{ optimization round: } K_{256}\rho_{\text{in},H}K_{256}^{\dagger} = \begin{pmatrix} 0 \\ 0.53740 \\ 0.53740 \end{pmatrix},$$

$$2^{\text{nd}} \text{ optimization round: } K_{5841}K_{5760}\rho_{\text{in},H}K_{5760}^{\dagger}K_{5841}^{\dagger} = \begin{pmatrix} 0 \\ 0.42674 \\ 0.66223 \end{pmatrix},$$

$$3^{\text{rd}} \text{ optimization round: } K_{5841}K_{5841}K_{5760}\rho_{\text{in},H}K_{5760}^{\dagger}K_{5841}^{\dagger}K_{5841}^{\dagger} = \begin{pmatrix} 0 \\ 0.37099 \\ 0.72200 \end{pmatrix},$$

$$4^{\text{th}} \text{ optimization round: } K_{5841}K_{5841}K_{5776}K_{5760}\rho_{\text{in},H}K_{5760}^{\dagger}K_{5776}^{\dagger}K_{5841}^{\dagger}K_{5841}^{\dagger} = \begin{pmatrix} 0 \\ 0.54945 \\ 0.56155 \end{pmatrix},$$

which is quite different from the series of Clifford operations that the algorithm found for the case when the input state was initiated inside of the octahedron. Two supplementary examples of slightly different Clifford operation sequences are provided in Appendix B.2.2.1 for the cases when the initial H state is mixed with an error probability of 0.02 and 0.03. The previously discussed observation that the algorithm is incentivized to pick a Clifford series that produces a state that is closer to the $|0\rangle$, since this will generally increase the success probability, can similarly be observed for these examples. It can be seen that the output states seem to be generally higher up on the Bloch sphere.

Despite the promises of H-type distillation from Figure 4.3, the output error probability plot in Figure 4.4 shows that the minimum output error probability curves converge to the initial error probability line for moderately small error probabilities. This implies that the input state cannot be distilled beyond this point and that the output state cannot become arbitrarily close to the target H state. This is in contrast to the Bravyi & Kitaev MSD algorithm, which can produce an output state that is arbitrarily close to the target state. It is because of this observation that we have chosen to call this algorithm "synthesis" and not "distillation" in order to highlight this important point, that the MMS algorithm cannot produce a state with arbitrarily high fidelity to the target. The minimum output error probability could most likely be further reduced by increasing the number of optimization rounds. However, this would double the number of needed input states for each additional round, and since more efficient methods that require far fewer input states already exist, such as the MSD algorithm, this increase would render this algorithm practically uninteresting.

The fact that some H-type distillation seems to be possible, while the fidelity for the T state could not be increased by any means with the MMS algorithm, hints at the fact that H-type distillations are easier to accomplish. This coincides with the results from Bravyi & Kitaev, where their threshold fidelity for the H state distillations is closer to the optimal compared to the T state case. The implication of this is that MSD could distill H-type magic states from states that were to a greater extent faultier compared to the case of the T-type distillation. Other literature [28], [32] seems to report similar results that the H states have some inherent feature that eases the distillation process, but it is not evident why this is the case and our analysis has not allowed to unveil the truth behind this observation. A hypothesis that could explain why some H-type distillation is possible with the MMS algorithm is that there are simply more H states than T states. Since the MMS algorithm effectively compares the output states to all magic states of the same type by shifting the outputs into the same hextant, there is a larger possibility for the output to be closer to a H state than compared to a T state, thus increasing the distillation potential.

At first sight, it might appear as if MMS and MSD would accomplish the same objective since they are implemented with the same actions. However, it is evident from the results in Sections 4.1.1 and 4.1.2 that the distillation possibilities of MMS are far worse than MSD. An intuitive explanation for this is that MSD will have access to larger Clifford groups that can simultaneously act on more qubits. This becomes apparent by considering that MSD is implemented with either five or fifteen input states, in contrast to the two that are used in each module of MMS. Therefore, MSD would be able to transform its input states to a larger number of outputs, thus increasing the opportunity to find operations that will increase the fidelity to the target state, even in comparison to using four rounds of MMS that practically use sixteen input states. This reasoning might explain why MSD can obtain output states that are arbitrarily close to the target magic state, whereas MMS cannot.

An observation that can be made by looking at the probability plots for both the T and H results, which can appear strange, is that the probability is smaller than one for all initial error probabilities. This means that even when the initial state is precisely equal to the target state, the probability of obtaining the target state is less than one. This is a weakness in the algorithm which is caused by the fact that even though the optimal Clifford operations are those that leave the state invariant, the algorithm will still need to apply an operation, possibly identity, and post-select with regards to $|0\rangle$, which will give a probability that is less than one, unless the state is precisely equal to $|0\rangle$. However, we theorize that this issue could be solved by utilizing a classical feed-forward operation. This can be done by measuring the second qubit instead of post-selecting and then applying a single-qubit Clifford operation on the first qubit that depends on measurement outcome, thus effectively creating a control Clifford gate [2]. The classical feed-forward Clifford can be implemented with the already permitted actions of the algorithm and would thus be a valid extension to the method. Despite the potential improvements in success probability, classical feed-forward would most likely not increase output fidelity. This assumption stems from the fact that the applied single-qubit operation would effectively be included in the optimization over all the two-qubit Clifford operations in subsequent rounds.

Chapter 5

Conclusion & future studies

In this thesis, we investigated a version of the magic state distillation algorithm that was proposed by Bravyi & Kitaev that only uses two states as inputs for each round of optimization, which we call modular magic synthesis. The results show that the MMS algorithm does not distill any T states and could only distill H states up to a certain point. This implies that the MMS algorithm would not in general be suitable to obtain magic states, because other more efficient methods already exist. However, it could still be useful to study these kinds of versions of distillation algorithm that only uses a small number of Clifford operations, since it could give insight into what is required to distill a magic state.

A modification that could be made to the MMS algorithm, which was mentioned in Section 4.2.2, is allowing for classical feed-forward operations. This could be implemented by exchanging all or some of the post-selection measurements with classical feed-forward operations which would apply a single-qubit Clifford operations to the first qubit that depends on the measurement result of the second qubit. This could be interesting since it most likely would increase the success probability for the distillation procedure. However, we theorize this will most likely not increase output fidelity.

Similarly, it would be interesting to increase the number of implemented optimization rounds to investigate if the fidelity could be further increased for H-type distillation. As was also mentioned, this would not be interesting from a practical point of view, since the Bravyi & Kitaev algorithm requires fewer input qubits and can give better results. However, it could be of note from a theoretical perspective to study if the distillation could be enhanced by increasing the number of rounds compared to just the four rounds of MMS.

A third possible extension could be to perform an equivalent analysis for the continuous variable model of quantum computation, which has an analogous need for resourceful states to simulate a universal quantum computer [35]. The corresponding problem of acquiring magic states in the discrete quantum computer paradigm is to obtain non-Gaussian states by only applying Gaussian operators in the continuous variable model. This study could yield insight into the similarities and differences between these paradigms while deepening the knowledge of the individual models.

References

[1] S. Bravyi and A. Kitaev, “Universal quantum computation with ideal clifford gates and noisy ancillas”, *Physical Review A*, vol. 71, no. 2, Feb. 2005, ISSN: 1094-1622. DOI: 10.1103/physreva.71.022316. [Online]. Available: http://dx.doi.org/10.1103/PhysRevA.71.022316.

[2] M. A. Nielsen and I. L. Chuang, *Quantum computation and quantum information*, 10th anniversary ed. New York: Cambridge University Press, 2010, ISBN: 978-1-107-00217-3.

[3] J. J. Sakurai and J. Napolitano, *Modern Quantum Mechanics*, 3rd. Philadelphia: Cambridge University Press, 2021, ISBN: 978-1-108-47322-4. DOI: 10.1017/9781108587280.

[4] D. J. Griffiths and D. F. Schroeter, *Introduction to Quantum Mechanics*, 3rd. Oregon: Cambridge University Press, 2018, ISBN: 978-1-107-18963-8. DOI: 10.1017/9781316995433.

[5] R. P. Feynman, “Simulating physics with computers”, *International Journal of Theoretical Physics*, vol. 21, pp. 467–488, 1982. DOI: https://doi.org/10.1007/BF02650179.

[6] P. Benioff, “The computer as a physical system: A microscopic quantum mechanical hamiltonian model of computers as represented by turing machines”, *Journal of Statistical Physics*, vol. 22, pp. 563–591, 1980. DOI: https://doi.org/10.1007/BF01011339.

[7] P. W. Shor, “Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer”, *SIAM Journal on Computing*, vol. 26, no. 5, pp. 1484–1509, Oct. 1997, ISSN: 1095-7111. DOI: 10.1137/s0097539795293172. [Online]. Available: http://dx.doi.org/10.1137/S0097539795293172.

[8] P. W. Shor, *Introduction to quantum algorithms*, 2001. arXiv: quant-ph/0005003 [quant-ph].

[9] M. A. Broome, A. Fedrizzi, S. Rahimi-Keshari, *et al.*, “Photonic boson sampling in a tunable circuit”, *Science*, vol. 339, no. 6121, pp. 794–798, Feb. 2013, ISSN: 1095-9203. DOI: 10.1126/science.1231440. [Online]. Available: http://dx.doi.org/10.1126/science.1231440.

[10] L. K. Grover, *A fast quantum mechanical algorithm for database search*, 1996. arXiv: quant-ph/9605043 [quant-ph].

[11] R. Jozsa, *Searching in grover’s algorithm*, 1999. arXiv: quant-ph/9901021 [quant-ph].

[12] A. Abbas, A. Ambainis, B. Augustino, *et al.*, *Quantum optimization: Potential, challenges, and the path forward*, 2023. arXiv: `2312.02279 [quant-ph]`.

[13] A. Perdomo-Ortiz, M. Benedetti, J. Realpe-Gómez, and R. Biswas, “Opportunities and challenges for quantum-assisted machine learning in near-term quantum computers”, *Quantum Science and Technology*, vol. 3, no. 3, p. 030 502, Jun. 2018. DOI: `10.1088/2058-9565/aab859`. [Online]. Available: `https://dx.doi.org/10.1088/2058-9565/aab859`.

[14] M. Schuld and N. Killoran, “Is quantum advantage the right goal for quantum machine learning?”, *PRX Quantum*, vol. 3, p. 030 101, 3 Jul. 2022. DOI: `10.1103/PRXQuantum.3.030101`. [Online]. Available: `https://link.aps.org/doi/10.1103/PRXQuantum.3.030101`.

[15] A. M. Turing, “Systems of logic based on ordinals”, *Proceedings of the London Mathematical Society, Series 2*, vol. 45, pp. 161–228, 1939, ISSN: 1460-244X. DOI: `10.1112/plms/s2-45.1.161`. eprint: `https://onlinelibrary.wiley.com/doi/pdf/10.1112/plms/s2-45.1.161`. [Online]. Available: `https://onlinelibrary.wiley.com/doi/abs/10.1112/plms/s2-%2045.1.161`.

[16] A. Matuschak and M. A. Nielsen. “Quantum computing for the very curious”. San Francisco. (2019), [Online]. Available: `https://quantum.country/qcvc`.

[17] D. Deutsch, A. Barenco, and A. Ekert, “Universality in quantum computation”, *Proceedings of the Royal Society of London. Series A: Mathematical and Physical Sciences*, vol. 449, no. 1937, pp. 669–677, Jun. 1995, ISSN: 2053-9177. DOI: `10.1098/rspa.1995.0065`. [Online]. Available: `http://dx.doi.org/10.1098/rspa.1995.0065`.

[18] E. Chitambar and G. Gour, “Quantum resource theories”, *Reviews of Modern Physics*, vol. 91, no. 2, Apr. 2019, ISSN: 1539-0756. DOI: `10.1103/revmodphys.91.025001`. [Online]. Available: `http://dx.doi.org/10.1103/RevModPhys.91.025001`.

[19] J. Schwichtenberg, *Physics from Symmetry*, 2nd. Karlsruhe: Springer, 2018, ISBN: 978-3-319-66630-3. DOI: `https://doi.org/10.1017/978-3-319-66631-0`.

[20] G. Vidal, “Efficient classical simulation of slightly entangled quantum computations”, *Physical Review Letters*, vol. 91, no. 14, Oct. 2003, ISSN: 1079-7114. DOI: `10.1103/physrevlett.91.147902`. [Online]. Available: `http://dx.doi.org/10.1103/PhysRevLett.91.147902`.

[21] S. Aaronson, *Quantum computing, postselection, and probabilistic polynomial-time*, 2004. arXiv: `quant-ph/0412187 [quant-ph]`.

[22] J. Tolar, “On clifford groups in quantum computing”, *Journal of Physics: Conference Series*, vol. 1071, p. 012 022, Aug. 2018, ISSN: 1742-6596. DOI: `10.1088/1742-6596/1071/1/012022`. [Online]. Available: `http://dx.doi.org/10.1088/1742-6596/1071/1/012022`.

[23] A. R. Calderbank, E. M. Rains, P. W. Shor, and N. J. A. Sloane, *Quantum error correction via codes over gf(4)*, 1997. arXiv: quant-ph/9608006 [quant-ph].

[24] R. Koenig and J. A. Smolin, “How to efficiently select an arbitrary clifford group element”, *Journal of Mathematical Physics*, vol. 55, no. 12, Dec. 2014, ISSN: 1089-7658. DOI: 10.1063/1.4903507. [Online]. Available: http://dx.doi.org/10.1063/1.4903507.

[25] D. Gottesman, “Theory of fault-tolerant quantum computation”, *Physical Review A*, vol. 57, no. 1, pp. 127–137, Jan. 1998, ISSN: 1094-1622. DOI: 10.1103/physreva.57.127. [Online]. Available: http://dx.doi.org/10.1103/PhysRevA.57.127.

[26] S. Aaronson and D. Gottesman, “Improved simulation of stabilizer circuits”, *Physical Review A*, vol. 70, no. 5, Nov. 2004, ISSN: 1094-1622. DOI: 10.1103/physreva.70.052328. [Online]. Available: http://dx.doi.org/10.1103/PhysRevA.70.052328.

[27] M. C. de Oliveira and W. J. Munro, “Quantum computation with mesoscopic superposition states”, *Physical Review A*, vol. 61, no. 4, Mar. 2000, ISSN: 1094-1622. DOI: 10.1103/physreva.61.042309. [Online]. Available: http://dx.doi.org/10.1103/PhysRevA.61.042309.

[28] B. W. Reichardt, *Quantum universality by state distillation*, 2009. arXiv: quant-ph/0608085 [quant-ph].

[29] D. Litinski, “Magic state distillation: Not as costly as you think”, *Quantum*, vol. 3, p. 205, Dec. 2019, ISSN: 2521-327X. DOI: 10.22331/q-2019-12-02-205. [Online]. Available: http://dx.doi.org/10.22331/q-2019-12-02-205.

[30] C. H. Bennett, D. P. DiVincenzo, J. A. Smolin, and W. K. Wootters, “Mixed-state entanglement and quantum error correction”, *Physical Review A*, vol. 54, no. 5, pp. 3824–3851, Nov. 1996, ISSN: 1094-1622. DOI: 10.1103/physreva.54.3824. [Online]. Available: http://dx.doi.org/10.1103/PhysRevA.54.3824.

[31] R. Laflamme, C. Miquel, J. P. Paz, and W. H. Zurek, *Perfect quantum error correction code*, 1996. arXiv: quant-ph/9602019 [quant-ph].

[32] B. W. Reichardt, “Quantum universality from magic states distillation applied to css codes”, *Quantum Information Processing*, vol. 4, no. 3, pp. 251–264, Aug. 2005, ISSN: 1573-1332. DOI: 10.1007/s11128-005-7654-8. [Online]. Available: http://dx.doi.org/10.1007/s11128-005-7654-8.

[33] E. T. Campbell and D. E. Browne, “Bound states for magic state distillation in fault-tolerant quantum computation”, *Physical Review Letters*, vol. 104, no. 3, Jan. 2010, ISSN: 1079-7114. DOI: 10.1103/physrevlett.104.030503. [Online]. Available: http://dx.doi.org/10.1103/PhysRevLett.104.030503.

[34] A. Javadi-Abhari, M. Treinish, K. Krsulich, *et al.*, *Quantum computing with Qiskit*, 2024. DOI: 10.48550/arXiv.2405.08810. arXiv: 2405.08810 [quant-ph].

[35] A. Serafini, *Quantum Continuous Variables : A Primer of Theoretical Methods*. Boca Raton, FL: CRC Press, Taylor & Francis Group, 2017, ISBN: 978-1-4822-4634-6.

[36] B. M. Terhal and D. P. DiVincenzo, *Adaptive quantum computation, constant depth quantum circuits and arthur-merlin games*, 2004. arXiv: `quant-ph/0205133 [quant-ph]`.

Appendix A

Calculations and derivations

In this appendix, we have collected calculations and mathematical arguments that were used in the thesis. The appendix will be initiated with the derivation of the fidelity expression in Eq. (2.23), which will then be used in the next section where the relation between fidelity and error probability is made explicit. This is then followed by justifications of the two abbreviations that reduce the number of states that need to be considered when multiple rounds of the MMS algorithm are implemented. Afterward, we calculate the optimal threshold error probabilities and fidelities. The final section of this appendix covers the derivation of the analytical fidelity curve for H-type distillation when $\varepsilon > \varepsilon^*$.

A.1 Fidelity simplification

The fidelity has been used throughout this thesis to assess the quality of output states from the algorithms that have been covered. In this thesis, fidelity was defined as [1], [2]

$$F(\eta,\rho) \equiv \left(\mathrm{Tr}\left[\sqrt{\sqrt{\rho}\eta\sqrt{\rho}}\right]\right)^2, \tag{A.1}$$

were the two arbitrary density matrices ρ and η are in general an ensembles of pure states $|\psi_i\rangle\langle\psi_i|$ according to $\rho = \sum_i p_i |\psi_i\rangle\langle\psi_i|$ and $\eta = \sum_i e_i |\psi_i\rangle\langle\psi_i|$, which comes from Eq. (2.9). However, the fidelity expression in Eq. (A.1) can be simplified by noting that the density matrix of a pure state will be equal to its square, which was used in Eq. 2.11. Furthermore, if the density matrix ρ was a pure state, which means that it can simply be written as $\rho = |\psi_\rho\rangle\langle\psi_\rho|$ it can be noted that

$$\rho = \sqrt{\rho^2} = \sqrt{\rho}. \tag{A.2}$$

Using this relation and continuing to consider the case when ρ is a pure state, Eq. (A.1) can be simplified to

$$\begin{aligned}
F(\eta,\rho) &= \left(\mathrm{Tr}\left[\sqrt{\rho\eta\rho}\right]\right)^2 \\
&= \left(\mathrm{Tr}\left[\sqrt{|\psi_\rho\rangle\langle\psi_\rho|\,\eta\,|\psi_\rho\rangle\langle\psi_\rho|}\right]\right)^2 \\
&= \left(\sqrt{\langle\psi_\rho|\,\eta\,|\psi_\rho\rangle}\right)^2 \left(\mathrm{Tr}\left[\sqrt{|\psi_\rho\rangle\langle\psi_\rho|}\right]\right)^2 \\
&= \langle\psi_\rho|\,\eta\,|\psi_\rho\rangle \left(\mathrm{Tr}\left[|\psi_\rho\rangle\langle\psi_\rho|\right]\right)^2 \\
&= \mathrm{Tr}\left[\langle\psi_\rho|\,\eta\,|\psi_\rho\rangle\,|\psi_\rho\rangle\langle\psi_\rho|\right] \\
&= \mathrm{Tr}\left[\eta\rho^2\right] \\
&= \mathrm{Tr}\left[\eta\rho\right],
\end{aligned}$$

were both trace properties $\operatorname{Tr}[MN] = \operatorname{Tr}[NM]$ and $\operatorname{Tr}[cM] = c\operatorname{Tr}[M]$, for matrices M, N and constant c, have been used in the calculations. The fidelity can then be further rewritten by utilizing the expression of the density matrix from Eq. (2.12) and using the results that the Pauli matrices satisfy $\operatorname{Tr}[\sigma^i\sigma^j] = 0$ for $i \neq j$ and $\operatorname{Tr}[\sigma^i\sigma^i] = \operatorname{Tr}[I] = 2$ will simplify the fidelity to

$$\begin{aligned} F(\eta,\rho) &= \operatorname{Tr}\left[\frac{1}{2}\left(I + u_{x,\eta}\sigma^x + u_{y,\eta}\sigma^y + u_{z,\eta}\sigma^z\right)\frac{1}{2}\left(I + u_{x,\rho}\sigma^x + u_{y,\rho}\sigma^y + u_{z,\rho}\sigma^z\right)\right] \\ &= \frac{1}{4}\operatorname{Tr}\left[\left(1 + u_{x,\eta}u_{x,\rho} + u_{y,\eta}u_{y,\rho} + u_{z,\eta}u_{z,\rho}\right)I\right] \\ &= \frac{1}{2}\left(1 + u_{x,\eta}u_{x,\rho} + u_{y,\eta}u_{y,\rho} + u_{z,\eta}u_{z,\rho}\right) \\ &= \frac{1}{2}\left(1 + \vec{u}_\eta \cdot \vec{u}_\rho\right), \end{aligned} \tag{A.3}$$

where $\vec{u}_\eta$ and $\vec{u}_\rho$ are Bloch vectors, which are defined in Section 2.1.3.

A.2 Relation between fidelity and ε

The quantities fidelity and error probability are strongly correlated in the sense that they can both describe how close a mixed state is to a target state. Their correspondence can be made clear by considering the case of a general density matrix ρ_{mix} that is mixed by an error probability ε. After applying the dephasing transformation, which is described in Section 3.1.1, the mixed state will be in a superposition between the pure target state ρ_0 and its orthogonal counterpart ρ_1, according to

$$D\left(\rho_{\text{mix}}(\varepsilon)\right) = (1-\varepsilon)\,\rho_0 + \varepsilon\rho_1, \tag{A.4}$$

where the pure states will be written with the same general structure as Eq. (2.12)

$$\rho_0 = \frac{1}{2}\left(I + u_x\sigma^x + u_y\sigma^y + u_z\sigma^z\right), \tag{A.5}$$

which implies its orthogonal counterpart will be given as

$$\rho_1 = \frac{1}{2}\left(I - u_x\sigma^x - u_y\sigma^y - u_z\sigma^z\right). \tag{A.6}$$

Expanding Eq. (A.4) in terms of expression in Eqs. (A.5) and (A.6) results in

$$\begin{aligned} D\left(\rho_{\text{mix}}(\varepsilon)\right) &= \frac{(1-\varepsilon)}{2}\left(I + u_x\sigma^x + u_y\sigma^y + u_z\sigma^z\right) + \frac{\varepsilon}{2}\left(I - u_x\sigma^x - u_y\sigma^y - u_z\sigma^z\right) \\ &= \frac{1}{2}\left(I + (1-2\varepsilon)\left(u_x\sigma^x + u_y\sigma^y + u_z\sigma^z\right)\right). \end{aligned}$$

From this result, it can be seen that the Bloch vector for the mixed state $\vec{u}_{\text{mix}} = (u_x, u_y, u_z)$ will be related to the target state Bloch vector $\vec{u}_0$ according to

$$\vec{u}_{\text{mix}} = (1-2\varepsilon)\,\vec{u}_0. \tag{A.7}$$

By inserting Eq. (A.4) and the target state ρ_0 into the final expression of fidelity in Eq. (A.3), it is possible to obtain a direct relation between fidelity and error probability, given as

$$\begin{aligned} F\left(D\left(\rho_{\text{mix}}(\varepsilon)\right), \rho_0\right) &= \frac{1}{2}\left(1 + \vec{u}_{\text{mix}} \cdot \vec{u}_0\right) \\ &= \frac{1}{2}\left(I + (1-2\varepsilon)\underbrace{\vec{u}_0 \cdot \vec{u}_0}_{=1}\right) \\ &= 1-\varepsilon, \end{aligned}$$

where it has been used that the length of a pure state is equal to 1.

A.3 Shift into the positive octant

By shifting the output states that result from a round of optimization into the positive octant of the Bloch sphere, numerous states will be shifted to the same position. Since the MMS algorithm only investigated output states in terms of their fidelity to a target magic state, states occupying the same position will be regarded as identical. This will substantially reduce the number of output states that need to be considered after each round of optimization and therefore lower the computational burden of the simulation, which will allow for the implementation of multiple optimization rounds within a feasible time frame.

It is already stated in Ref. [1] that the Bloch vector can be shifted into the positive octant by only using Clifford operations, which makes this a reasonable abbreviation and should not diminish the results of this thesis. Furthermore, below is an example of a method presented to find a single qubit Clifford operation $C_1 \in \mathcal{C}_1'$ that will shift an arbitrary state into the positive octant, which offers evidence beyond just asserting the statement. As previously mentioned, the single-qubit Clifford group is generated by the Hadamard and phase gates, and the members of the group are summarized in Appendix E.

First, a general density matrix ρ on the representation from Eq. (2.12) can be given as

$$\rho_0 = \frac{1}{2}\left(I + u_x\sigma^x + u_y\sigma^y + u_z\sigma^z\right),$$

where the elements u_x, u_y and u_z will constitute a Bloch vector $\vec{u} = (u_x, u_y, u_z)$. The Bloch vector will subsequently reside in the positive octant of the Bloch sphere in the case when $u_x, u_y, u_z > 0$. Eight conceivable cases need to be considered to demonstrate that there will always be a single-qubit Clifford operation that can shift the state into the positive octant. The trivial and simplest case is when all the elements already are in the positive octant and there is no action needed. The complete opposite scenario will then be the corresponding orthogonal state to ρ_0, where all the elements of the Bloch vector are in the negative octant. A density matrix for this case was already defined in Eq. (A.6) to be

$$\rho_1 = \frac{1}{2}\left(I - u_x\sigma^x - u_y\sigma^y - u_z\sigma^z\right).$$

To find the Clifford operation that will beneficially shift the elements of this state into the positive octant, it can be useful to investigate how the Pauli matrices will transform under various different members of the single-qubit Clifford group. Three Clifford operators that will be of particular importance to this study are the Pauli operators, which will act on themselves according to the recapitalization

$$\sigma^p\sigma^p\sigma^{p\dagger} = \sigma^p \quad \text{and} \quad \sigma^p\sigma^q\sigma^{p\dagger} = -\sigma^q, \quad \text{where } p, q \in \{x, y, z\}. \tag{A.8}$$

Another important operator for this study is the Hadamard operator. It will act on the Pauli matrices according to the fifth line in Table E.1 of Appendix E, which are

$$H\sigma^x H^\dagger = \sigma^z, \quad H\sigma^y H^\dagger = -\sigma^y, \quad H\sigma^z H^\dagger = \sigma^x. \tag{A.9}$$

By using relations in Eqs. (A.8) and (A.9), it is possible to show that ρ_1 can be transformed into ρ_0 by first applying a Pauli-Y matrix and then a Hadamard, which will look like

$$\begin{aligned} H\sigma^y\rho_1\left(H\sigma^y\right)^\dagger &= H\left(\frac{1}{2}\left(\sigma^y I\sigma^y - u_x\sigma^y\sigma^x\sigma^y - u_y\sigma^y\sigma^y\sigma^y - u_z\sigma^y\sigma^z\sigma^y\right)\right)H \\ &= \frac{1}{2}\left(HIH + u_xH\sigma^xH - u_yH\sigma^yH + u_zH\sigma^xH\right) \\ &= \frac{1}{2}\left(I + u_x\sigma^z + u_y\sigma^y + u_z\sigma^x\right). \end{aligned}$$

Additionally, it is useful to note that this calculation shows that if a general Bloch vector only has negative elements, it can always be rotated into the positive octant. It can also be noted that because H, $\sigma^y \in \mathcal{C}_1'$, then also $H\sigma^y \in \mathcal{C}_1'$, which comes from the definitions of a group [19].

The other six cases that additionally need to be considered are the three scenarios when only one part of the Bloch vector is negative and the three scenarios when two parts of the Bloch vector are negative. For example, it can straightforwardly be shown that the density matrix

$$\rho_2 = \frac{1}{2}\left(I - u_x\sigma^x + u_y\sigma^y + u_z\sigma^z\right)$$

can be rotated to the form of ρ_1 by applying a σ^x,

$$\begin{aligned} \sigma^x\rho_2\left(\sigma^x\right)^\dagger &= \frac{1}{2}\left(\sigma^x I\sigma^x - u_x\sigma^x\sigma^x\sigma^x + u_y\sigma^x\sigma^y\sigma^x + u_z\sigma^x\sigma^z\sigma^x\right) \\ &= \frac{1}{2}\left(I - u_x\sigma^x - u_y\sigma^y - u_z\sigma^z\right) \\ &= \rho_1, \end{aligned}$$

where Eq. (A.8) have been used. According to the calculations that were discussed for the previous case, by sequential application of the operators σ^y and H, the state ρ_2 would be shifted into the positive octant. The same calculations can also be performed for the other two cases when the y or the z components of the ρ_2 Bloch vector are negative.

Finally, the three cases when two of the constituent elements of the Bloch vector are negative can similarly be investigated by studying the example density matrix

$$\rho_3 = \frac{1}{2}\left(I + u_x\sigma^x - u_y\sigma^y - u_z\sigma^z\right).$$

Remembering Eq. (A.8), it can be seen that ρ_3 can be shifted into the positive octant by simply applying a σ^x, which would look like

$$\begin{aligned} \sigma^x\rho_3\sigma^x &= \frac{1}{2}\left(\sigma^x I\sigma^x + u_x\sigma^x\sigma^x\sigma^x - u_y\sigma^x\sigma^y\sigma^x - u_z\sigma^x\sigma^z\sigma^x\right) \\ &= \frac{1}{2}\left(I + u_x\sigma^x + u_y\sigma^y + u_z\sigma^z\right). \end{aligned}$$

Similar to the previous scenario, the same calculations can be carried out for the case that the y or the z component is the positive element.

This concludes the method of finding a single-qubit Clifford operation that shifts a state to the positive octant and it should show that all possible positions that a Bloch vector can have around the Bloch sphere can be rotated into the positive octant. It can additionally be noted that these operations do not uniquely shift the states into the positive octant, and there are generally more Clifford operations that can accomplish the same goal. These calculations have also been summarized in Table A.1 and are illustrated in Figure A.1.

Negative element	Action
u_p	$H\sigma^y\sigma^p$
u_p, u_q	σ^r
u_p, u_q, u_r	$\sigma^y H$

Table A.1: This is a summarization of the different non-trivial cases for negative components of the Bloch vector and what action could be taken to shift it into the positive octant. The u:s are the components of the Bloch vector where $p, q, r \in \{x, y, z\}$.

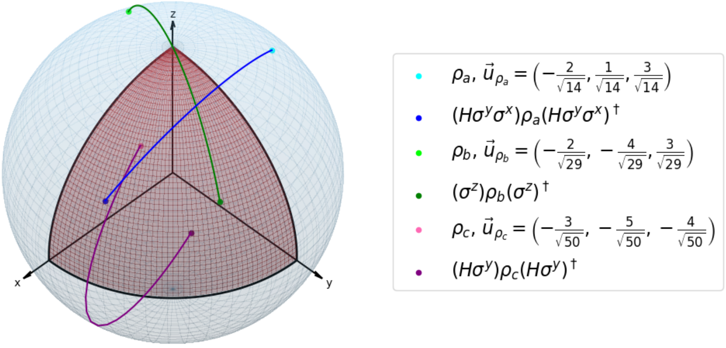
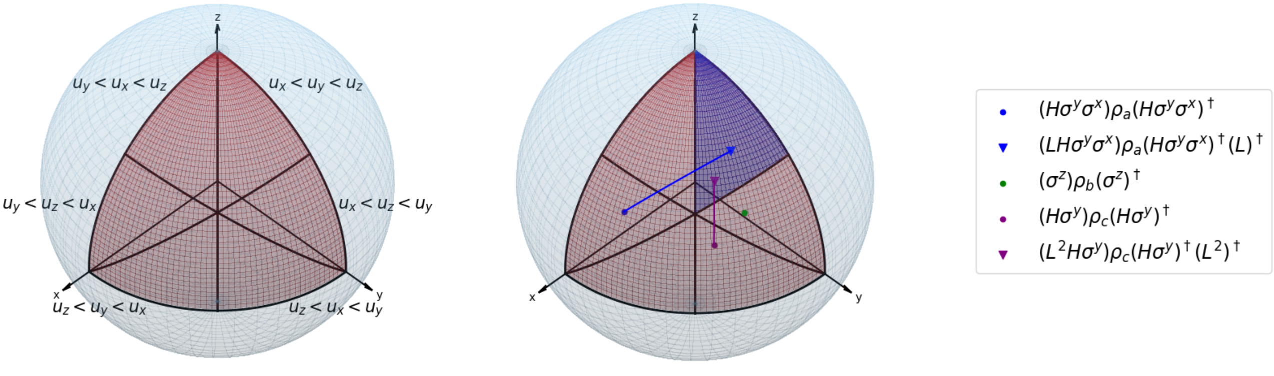


Figure A.1: A plot of the Bloch sphere with three capriciously chosen points, where the positive octant is marked with the red area. By using the method that is described above of finding an adequate single-qubit Clifford operation, these points have been transformed accordingly such that they will be inside of the positive octant.

A.4 Similarity among the hextants

Besides shifting the Bloch vectors into the positive octant of the Bloch sphere, by also transforming the states into one hextant of the positive octant, a majority of the states will additionally be transformed to the same position of the Bloch sphere. This will thus further reduce the number of states that need to be considered after each optimization round. One octant can be subdivided into six hextants, where the relative magnitude ordering among the elements in a Bloch vector determines which hextant it will be located in. The hextant are shown in Figure A.2a with the corresponding ordering of the sizes of the Bloch vector elements.



(a) **(b)**

Figure A.2: In both Figures (a) and (b), the positive octant is noted with a red area and is divided into its hextant. Figure (a) highlights which hextant will correspond to what ordering of sizes among the Bloch vector elements in a general vector $\vec{u} = (u_x, u_y, u_z)$. In Figure (b), the ordered hextant that is mainly considered in this thesis is additionally emphasized with the blue area and shows how the ρ_a and ρ_c states from Figure A.1 can further be transformed with a single-qubit Clifford operation such that they will reside in that hextant.

Although single-qubit Clifford operations can be used to shift any arbitrary Bloch vector to the positive octant, single-qubit Clifford cannot shift all states to a specific hextant. This can be seen by studying the transformation in Appendix E, which will make it apparent that there are only two single-qubit Clifford operators, disregarding identity, which will transform all Pauli matrices to positive Pauli matrices. These operators are the operators that are the same as L and L^2 up to a global phase, where they will rotate states around the L state in the positive octant with angles $\frac{\pi}{3}$ and $\frac{2\pi}{3}$ respectively [28]. The operators will act on the Pauli matrices according to

$$\begin{aligned} L\sigma^x L^\dagger &= \sigma^z, & L\sigma^y L^\dagger &= \sigma^x, & L\sigma^z L^\dagger &= \sigma^y, \\ L^2\sigma^x (L^2)^\dagger &= \sigma^y, & L^2\sigma^y (L^2)^\dagger &= \sigma^z, & L^2\sigma^z (L^2)^\dagger &= \sigma^x. \end{aligned}$$

It can then be noted from these transformations that they can only shift state to a specific hextant when the state is initially located inside a hextant that is $n\frac{\pi}{3}$ from the target hextant. For example, in Figure A.2a, it can be seen that both ρ_a and ρ_c can be shifted to the $u_x < u_y < u_z$ hextant. Meanwhile, ρ_b cannot be rotated to that hextant, since it becomes located inside of a hextant that is directly adjacent to the target hextant.

Although not all states can be shifted to the same hextant by only using single-qubit Clifford operations, it does not matter which hextant of the positive octant a state is located in for the purposes of this thesis. The reason for this is that states are only perceived as the fidelity between it and a target magic state. This can be seen by looking at how fidelity is calculated with Eq. (A.3). For example, since all the elements of the T state in the positive octant are equal, the fidelity will be the same between the T state and all the states with different permutations of its Bloch vector elements. This can explicitly be calculated by using the two differently transformed ρ_a states which are shown in Figure A.2b. This could look like

$$\begin{aligned} \rho_{a1} &= \left(H\sigma^y\sigma^x\right)\rho_a\left(H\sigma^y\sigma^x\right)^\dagger = \frac{1}{2}\left(I + \frac{3}{\sqrt{14}}\sigma^x + \frac{1}{\sqrt{14}}\sigma^y + \frac{2}{\sqrt{14}}\sigma^z\right), \\ F(\rho_{a1},\rho_T) &= \frac{1}{2}\left(1 + \frac{1}{\sqrt{3}}\left(\frac{3}{\sqrt{14}} + \frac{1}{\sqrt{14}} + \frac{2}{\sqrt{14}}\right)\right) \approx 0.9629, \\ \rho_{a2} &= \left(LH\sigma^y\sigma^x\right)\rho_a\left(H\sigma^y\sigma^x\right)^\dagger(L)^\dagger = \frac{1}{2}\left(I + \frac{1}{\sqrt{14}}\sigma^x + \frac{2}{\sqrt{14}}\sigma^y + \frac{3}{\sqrt{14}}\sigma^z\right), \\ F(\rho_{a2},\rho_T) &= \frac{1}{2}\left(1 + \frac{1}{\sqrt{3}}\left(\frac{1}{\sqrt{14}} + \frac{2}{\sqrt{14}} + \frac{3}{\sqrt{14}}\right)\right) \approx 0.9629. \end{aligned}$$

A similar argument can be made for the fidelity between states in the positive octant with different ordered elements and the H states. The fidelity between a state and one of the H states on the circumference of the positive octant will be the same as the fidelity between that state with different ordered elements and another H state on the circumference. This effectively means that the ordering of the elements in a state does not matter when only the fidelity is evaluated. This can similarly be seen by considering two transformed ρ_c states in A.2b, which will yield

$$\begin{aligned} \rho_{c,1} &= \left(H\sigma^y\right)\rho_c\left(H\sigma^y\right)^\dagger = \frac{1}{2}\left(I + \frac{4}{\sqrt{50}}\sigma^x + \frac{5}{\sqrt{50}}\sigma^y + \frac{3}{\sqrt{50}}\sigma^z\right), \\ F(\rho_{c,1},\rho_{H,2}) &= \frac{1}{2}\left(1 + \frac{1}{\sqrt{2}}\cdot\frac{4}{\sqrt{50}} + \frac{1}{\sqrt{2}}\cdot\frac{5}{\sqrt{50}} + 0\cdot\frac{3}{\sqrt{50}}\right) = 0.95, \\ \rho_{c,2} &= \left(L^2H\sigma^y\right)\rho_c\left(H\sigma^y\right)^\dagger\left(L^2\right)^\dagger = \frac{1}{2}\left(I + \frac{3}{\sqrt{50}}\sigma^x + \frac{4}{\sqrt{50}}\sigma^y + \frac{5}{\sqrt{50}}\sigma^z\right), \\ F(\rho_{c,2},\rho_{H,1}) &= \frac{1}{2}\left(1 + 0\cdot\frac{3}{\sqrt{50}} + \frac{1}{\sqrt{2}}\cdot\frac{4}{\sqrt{50}} + \frac{1}{\sqrt{2}}\cdot\frac{5}{\sqrt{50}}\right) = 0.95, \end{aligned}$$

The relations of the fidelities between magic states and states with different permutations of the constituent elements are represented in Figure A.3.

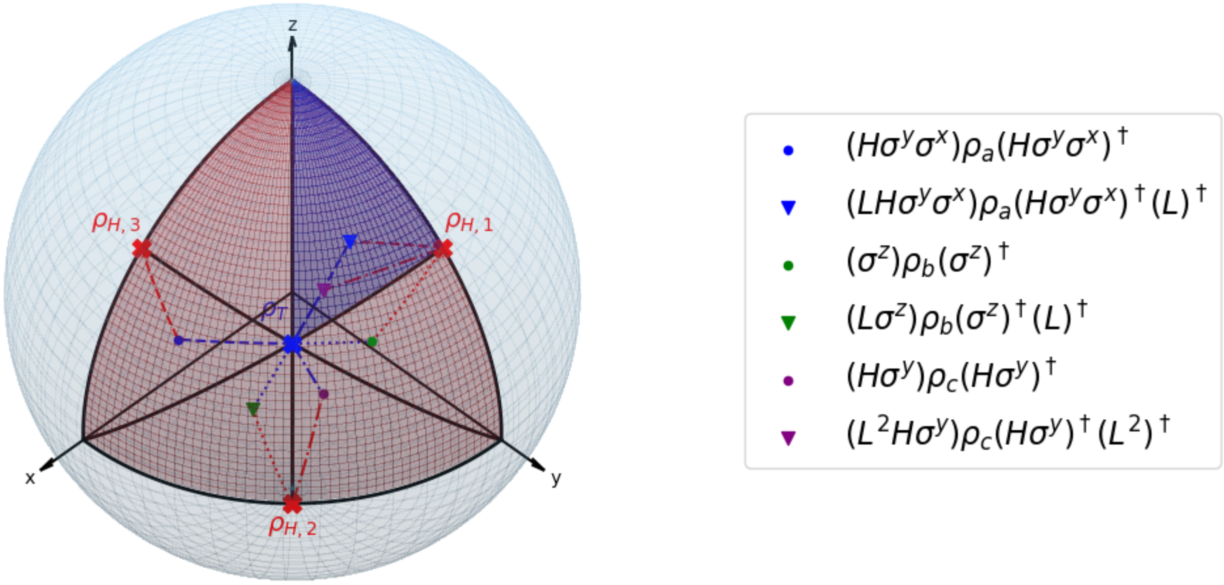


Figure A.3: Continuing from the states in Figure A.2b, this plot emphasizes that the fidelity between a magic state and states that only differ by a transformation that switches the ordering among the elements will be the same. This is done by drawing lines between the arbitrary states in the positive octant and the magic states, which represents that the states are located at the same distance from the target magic states.

A.5 Determining the threshold error probability and fidelity

The threshold error probability ε^* marks the shift from an input state that is inside of the octahedron to a state outside of the octahedron hull. ε^* can be determined from the general expressions of the initial mixed inputs in Eqs. (3.4) and (3.5) by solving for what ε gives a Bloch vector that has elements that sum to one, since this defines a state which will be located on the octahedron. Starting by expanding the expression for the initial mixed H state from Eq. (3.4) will give

$$\begin{aligned}\rho_{\text{in},H} &= \frac{1-\varepsilon}{2}\left(I + \frac{1}{\sqrt{2}}\left(\sigma^y + \sigma^z\right)\right) + \frac{\varepsilon}{2}\left(I - \frac{1}{\sqrt{2}}\left(\sigma^y + \sigma^z\right)\right) \\ &= \frac{1}{2}\left(I + \frac{1-2\varepsilon}{\sqrt{2}}\sigma^y + \frac{1-2\varepsilon}{\sqrt{2}}\sigma^z\right),\end{aligned}$$

where the analogous Bloch vector to this state will be given as

$$\vec{u}_{\text{in},H} = \frac{1-2\varepsilon}{\sqrt{2}}\begin{pmatrix}0\\1\\1\end{pmatrix}. \tag{A.10}$$

The ε that will place this state on the octahedron can then be found by solving

$$\frac{1-2\varepsilon_H^*}{\sqrt{2}} + \frac{1-2\varepsilon_H^*}{\sqrt{2}} = 1 \implies \varepsilon_H^* = \frac{1}{2}\left(1 - \frac{1}{\sqrt{2}}\right) \approx 0.146447.$$

Similar calculations can be performed to find which ε will place the initial mixed T state from Eq. (3.5) on the octahedron. Expanding the input state will give

$$\begin{aligned}\rho_{\mathrm{in},T} &= \frac{(1-\varepsilon)}{2}\left(I + \frac{1}{\sqrt{3}}\left(\sigma^x + \sigma^y + \sigma^z\right)\right) + \frac{\varepsilon}{2}\left(I - \frac{1}{\sqrt{3}}\left(\sigma^x + \sigma^y + \sigma^z\right)\right) \\ &= \frac{1}{2}\left(I + \frac{1-2\varepsilon}{\sqrt{3}}\sigma^x + \frac{1-2\varepsilon}{\sqrt{3}}\sigma^y + \frac{1-2\varepsilon}{\sqrt{3}}\sigma^z\right),\end{aligned}$$

which will have the corresponding Bloch vector

$$\vec{u}_{\mathrm{in},T} = \frac{1-2\varepsilon}{\sqrt{3}}\begin{pmatrix}1\\1\\1\end{pmatrix}.$$

Similarly, the ε which will position the state on the octahedron can be solved for as

$$\frac{1-2\varepsilon_T^*}{\sqrt{3}} + \frac{1-2\varepsilon_T^*}{\sqrt{3}} + \frac{1-2\varepsilon_T^*}{\sqrt{3}} = 1 \implies \varepsilon_T^* = \frac{1}{2}\left(1 - \frac{1}{\sqrt{3}}\right) \approx 0.211325.$$

Since a state cannot be transformed from inside the octahedron to outside by only using Clifford operations, the largest possible fidelity for this case will be the corresponding threshold fidelity for a state that is located on the octahedron. By using the same argument that the Bloch vector elements of such a state will sum to one, the optimal threshold fidelity can be calculated for the H and T states as

$$\begin{aligned}F^*(\rho^*_{\mathrm{out},H}, \rho_{H_0}) &= \frac{1}{2}\left(1 + \vec{u}_{\mathrm{out},H}\cdot\vec{u}_{H_0}\right) = \frac{1}{2}\left(1 + \frac{1}{\sqrt{2}}\left(u_{\mathrm{out},T,y} + u_{\mathrm{out},T,z}\right)\right) \\ &= \frac{1}{2}\left(1 + \frac{1}{\sqrt{2}}\right) \approx 0.853553,\end{aligned}$$

$$\begin{aligned}F^*(\rho^*_{\mathrm{out},T}, \rho_{T_0}) &= \frac{1}{2}\left(1 + \vec{u}^*_{\mathrm{out},T}\cdot\vec{u}_{T_0}\right) = \frac{1}{2}\left(1 + \frac{1}{\sqrt{3}}\left(u^*_{\mathrm{out},T,x} + u^*_{\mathrm{out},T,y} + u^*_{\mathrm{out},T,z}\right)\right) \\ &= \frac{1}{2}\left(1 + \frac{1}{\sqrt{3}}\right) \approx 0.788675.\end{aligned}$$

A.6 Analytical solution for H distillation when $\varepsilon > \varepsilon^*$

As was discussed in Section 4.2.1, the optimal circuit of Clifford operations the algorithm finds for H-type distillation appears to be the same for all cases when the initial state is located inside of the octahedron. The only gate that the algorithms used for these scenarios is the Clifford gate that can be decomposed as the circuit in Figure A.4, which has the number 5760 in the generated list of Clifford operations and will therefore be called C_{5760}.

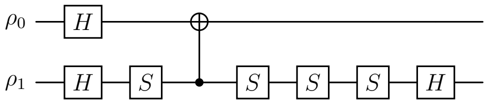


Figure A.4: Decomposition of the Clifford operation C_{5760} which is continuously used by the algorithm to optimize the fidelity when the input state is initialized inside of the octahedron for the case of H-type distillation.

The operation in Figure A.4 can equivalently be written as

$$C_{5760} = (I \otimes H)\left(I \otimes S^3\right) CNOT_{10}\,(I \otimes S)\,(H \otimes H)\,.$$

By using the calculations from Eqs. (2.19) and (2.20), the expression of C_{5760} can further be simplified as

$$\begin{aligned} C_{5760} &= (I \otimes H)\left(I \otimes S^3\right)(I \otimes |0\rangle\langle 0| + \sigma^x \otimes |1\rangle\langle 1|)\,(I \otimes S)\,(H \otimes H) \\ &= H \otimes \left(HS^3\,|0\rangle\langle 0|\,SH\right) + \sigma^x H \otimes \left(HS^3\,|1\rangle\langle 1|\,SH\right) \\ &= H \otimes |+\rangle\langle +| + \sigma^x H \otimes |-\rangle\langle -|\,, \end{aligned}$$

which will have the corresponding Hermitian conjugate

$$C^\dagger_{5760} = H \otimes |+\rangle\langle +| + H\sigma^x \otimes |-\rangle\langle -|\,.$$

The input state is a mixed H state that has been acted upon by the dephasing operation, which means that the state can in general be written as

$$\rho_{\text{in}} = \frac{1}{2}\left(I + c_y\sigma^y + c_z\sigma^z\right).$$

After acting on two of these general input states with C_{5760} and post-selecting, the unnormalized output state will be

$$\tilde{\rho}_{\text{out}} = (I \otimes \langle 0|)\,C_{5760}\,(\rho_{\text{in}} \otimes \rho_{\text{in}})\,C^\dagger_{5760}\,(I \otimes |0\rangle)\,,$$

which can be calculated by first considering the separate operators

$$\begin{aligned} (I \otimes \langle 0|)\,C_{5760} &= H \otimes \langle 0|+\rangle\,\langle +| + \sigma^x H \otimes \langle 0|-\rangle\,\langle -| \\ &= H \otimes \frac{1}{\sqrt{2}}\,\langle +| + \sigma^x H \otimes \frac{1}{\sqrt{2}}\,\langle -|\,, \end{aligned}$$

$$\begin{aligned} C^\dagger_{5760}\,(I \otimes |0\rangle) &= H \otimes |+\rangle\,\langle +|0\rangle + H\sigma^x \otimes |-\rangle\,\langle -|0\rangle \\ &= H \otimes |+\rangle\,\frac{1}{\sqrt{2}} + H\sigma^x \otimes |-\rangle\,\frac{1}{\sqrt{2}}. \end{aligned}$$

Using these partial calculations will then give

$$\begin{aligned} \tilde{\rho}_{\text{out}} &= \frac{1}{2}\left(H \otimes \langle +| + \sigma^x H \otimes \langle -|\right)(\rho_{\text{in}} \otimes \rho_{\text{in}})\left(H \otimes |+\rangle + H\sigma^x \otimes |-\rangle\right) \\ &= \frac{1}{2}\left(H \otimes \langle +| + \sigma^x H \otimes \langle -|\right)\left(\rho_{\text{in}} H \otimes \rho_{\text{in}}\,|+\rangle + \rho_{\text{in}} H\sigma^x \otimes \rho_{\text{in}}\,|-\rangle\right) \\ &= \frac{1}{2}\left(H\rho_{\text{in}} H \otimes \langle +|\,\rho_{\text{in}}\,|+\rangle + H\rho_{\text{in}} H\sigma^x \otimes \langle +|\,\rho_{\text{in}}\,|-\rangle\right. \\ &\quad \left. + \sigma^x H\rho_{\text{in}} H \otimes \langle -|\,\rho_{\text{in}}\,|+\rangle + \sigma^x H\rho_{\text{in}} H\sigma^x \otimes \langle -|\,\rho_{\text{in}}\,|-\rangle\right) \end{aligned}$$

The scalar products can be calculated by considering how the Pauli matrices act on the basis states $|+\rangle$ and $|-\rangle$

$$\begin{aligned} \sigma^x\,|+\rangle &= |+\rangle\,, \quad \sigma^y\,|+\rangle = -i\,|-\rangle\,, \quad \sigma^z\,|+\rangle = |-\rangle\,, \\ \sigma^x\,|-\rangle &= -\,|-\rangle\,, \quad \sigma^y\,|-\rangle = i\,|+\rangle\,, \quad \sigma^z\,|-\rangle = |+\rangle\,. \end{aligned}$$

Using these relations will give the scalar products as

$$\langle +|\,\rho_{\text{in}}\,|+\rangle = \langle +|\,\frac{1}{2}\left(I\,|+\rangle + c_y\sigma^y\,|+\rangle + c_z\sigma^z\,|+\rangle\right) = \langle +|\,\frac{1}{2}\left(|+\rangle - c_y i\,|-\rangle + c_z\,|-\rangle\right) = \frac{1}{2},$$

$$\langle +|\,\rho_{\text{in}}\,|-\rangle = \langle +|\,\frac{1}{2}\left(I\,|-\rangle + c_y\sigma^y\,|-\rangle + c_z\sigma^z\,|-\rangle\right) = \langle +|\,\frac{1}{2}\left(|-\rangle + c_y i\,|+\rangle + c_z\,|+\rangle\right) = \frac{1}{2}\left(c_y i + c_z\right),$$

$$\langle -|\,\rho_{\text{in}}\,|+\rangle = \langle -|\,\frac{1}{2}\left(I\,|+\rangle + c_y\sigma^y\,|+\rangle + c_z\sigma^z\,|+\rangle\right) = \langle -|\,\frac{1}{2}\left(|+\rangle - c_y i\,|-\rangle + c_z\,|-\rangle\right) = \frac{1}{2}\left(-c_y i + c_z\right),$$

$$\langle -|\,\rho_{\text{in}}\,|-\rangle = \langle -|\,\frac{1}{2}\left(I\,|-\rangle + c_y\sigma^y\,|-\rangle + c_z\sigma^z\,|-\rangle\right) = \langle -|\,\frac{1}{2}\left(|-\rangle + c_y i\,|+\rangle + c_z\,|+\rangle\right) = \frac{1}{2}.$$

By using the relations from Eqs. (A.8) and (A.9), the states can be calculated as

$$\begin{aligned} H\rho_{\text{in}}H &= \frac{1}{2}\left(HIH + c_y H\sigma^y H + c_z H\sigma^z H\right) \\ &= \frac{1}{2}\left(I - c_y\sigma^y + c_z\sigma^x\right), \end{aligned}$$

$$\begin{aligned} H\rho_{\text{in}}H\sigma^x &= \frac{1}{2}\left(HIH\sigma^x + c_y H\sigma^y H\sigma^x + c_z H\sigma^z H\sigma^x\right) \\ &= \frac{1}{2}\left(\sigma^x - c_y\sigma^y\sigma^x + c_z\sigma^x\sigma^x\right) \\ &= \frac{1}{2}\left(\sigma^x + c_y i\sigma^z + c_z I\right), \end{aligned}$$

$$\begin{aligned} \sigma^x H\rho_{\text{in}}H &= \frac{1}{2}\left(\sigma^x HIH + c_y\sigma^x H\sigma^y H + c_z\sigma^x H\sigma^z H\right) \\ &= \frac{1}{2}\left(\sigma^x - c_y\sigma^x\sigma^y + c_z\sigma^x\sigma^x\right) \\ &= \frac{1}{2}\left(\sigma^x - c_y i\sigma^z + c_z I\right), \end{aligned}$$

$$\begin{aligned} \sigma^x H\rho_{\text{in}}H\sigma^x &= \frac{1}{2}\left(\sigma^x HIH\sigma^x + c_y\sigma^x H\sigma^y H\sigma^x + c_z\sigma^x H\sigma^z H\sigma^x\right) \\ &= \frac{1}{2}\left(\sigma^x\sigma^x - c_y\sigma^x\sigma^y\sigma^x + c_z\sigma^x\sigma^x\sigma^x\right) \\ &= \frac{1}{2}\left(I + c_y\sigma^y + c_z\sigma^x\right). \end{aligned}$$

Combining these results will then give the unnormalized state

$$\begin{aligned} \tilde{\rho}_{\text{out}} = \frac{1}{2}\Bigg(&\frac{1}{4}\left(I - c_y\sigma^y + c_z\sigma^x\right) + \frac{1}{4}\left(c_y i + c_z\right)\left(\sigma^x + c_y i\sigma^z + c_z I\right) \\ &+ \frac{1}{4}\left(-c_y i + c_z\right)\left(\sigma^x - c_y i\sigma^z + c_z I\right) + \frac{1}{4}\left(I + c_y\sigma^y + c_z\sigma^x\right)\Bigg) \\ = \frac{1}{2}&\left(\frac{1 + c_z^2}{2}I + c_z\sigma^x - \frac{c_y^2}{2}\sigma^z\right). \end{aligned}$$

The probability of obtaining this state will be given as the trace of the unnormalized state

$$\mathrm{Tr}\left[\tilde{\rho}_{\text{out}}\right] = \frac{1+c_z^2}{2}, \tag{A.11}$$

which can also be used as a normalization factor to give the output

$$\rho_{\text{out}} = \frac{\tilde{\rho}_{\text{out}}}{\mathrm{Tr}\left[\tilde{\rho}_{\text{out}}\right]} = \frac{1}{2}\left(I + \frac{2c_z}{1+c_z^2}\sigma^x - \frac{c_y^2}{1+c_z^2}\sigma^z\right).$$

This output can then be shifted to the same hextant of the Bloch sphere as the input ρ_{in}, which is the blue hextant from Figure A.2b, by using the reasoning described in Sections A.3 and A.4. This will give the output as

$$\rho_{\text{out}} = \frac{1}{2}\left(I + \frac{c_y^2}{1+c_z^2}\sigma^y + \frac{2c_z}{1+c_z^2}\sigma^z\right). \tag{A.12}$$

This output state will then become the input for the next round of optimization. Since the optimal circuit that the algorithm found for this case only consisted of applying C_{5760} numerous times, the shifted output of the next round will have the same form as Eq. (A.12). Therefore, it will be possible to write the analytical density matrix of round $i+1$ as

$$\rho_{A,i+1} = \frac{1}{2}\left(I + c_{y,i+1}\sigma^y + c_{z,i+1}\sigma^z\right) = \frac{1}{2}\left(I + \frac{c_{y,i}^2}{1+c_{z,i}^2}\sigma^y + \frac{2c_{z,i}}{1+c_{z,i}^2}\sigma^z\right). \tag{A.13}$$

This density matrix will then have the corresponding Bloch vector $\vec{u}_{A,i+1} = \left(0, \frac{c_{y,i}^2}{1+c_{z,i}^2}, \frac{2c_{z,i}}{1+c_{z,i}^2}\right)$. Using the expression for the fidelity from Eq. (A.3) will give the general formula for the fidelity as

$$\begin{aligned} F\left(\rho_{A,i+1}, \rho_{H_0}\right) &= \frac{1}{2}\left(1 + \vec{u}_{A,i+1}\cdot\vec{u}_{H_0}\right) \\ &= \frac{1}{2}\left(1 + \frac{1}{\sqrt{2}\left(1+c_{z,i}^2\right)}\left(c_{y,i}^2 + 2c_{z,i}\right)\right) \end{aligned} \tag{A.14}$$

For this case when the initial input state is a mixed H state on the form Eq. (3.4), the initial constants $c_{y,0}$ and $c_{z,0}$ can be determined by considering the Bloch vector from Eq. (A.10), which implies that the constant will be

$$c_{y,0} = c_{z,0} = \frac{1-2\varepsilon}{\sqrt{2}}.$$

Appendix B

Supplementary results

We will in this appendix present supplementary results that were obtained throughout the thesis work. These results have been used in the thesis to further clarify arguments and additionally visualize statements. However, they were not strictly necessary for the discussion of the thesis and have therefore been delegated to this appendix. Nevertheless, they can still offer some valuable insights to further enhance the comprehension of the thesis material.

This appendix is divided into two sections, where we first present supplementary results for the case when the input state was a rotated magic state, and this is then followed by additional results that were obtained when a mixed magic state was considered as input. Both T and H states were used as inputs to obtain these results and each of the coming sections will present these scenarios separately.

B.1 Rotation results

As was mentioned previously, it has already been proven that a pure input state, which in this case is a rotated input state, can always be transformed into the target state. However, these results have still been important for this thesis as an initial analysis of the algorithm's function and an illustration of the possible output states for multiple rounds of optimization. This section will first present supplementary results for when a rotated T state was considered as input, and this is then followed by corresponding results for a rotated H state.

B.1.1 T state rotation results

In this section, we will present three different kinds of supplementary rotation results for the case when we tried to obtain a T state with the MMS algorithm. First in Figure B.1, we show all the possible fidelities that can be obtained for one round of optimization for six different initial rotation angles, which is the T state equivalent to the results in Figure 3.2. Subsequently, the plots in Figure B.2 show the maximum fidelities that the MMS algorithm found for ten different initial rotations and two rounds of optimization. In Figures B.1 and B.4, we have plotted all the unique states that could be obtained on a Bloch sphere for one and two rounds of optimization, respectively.

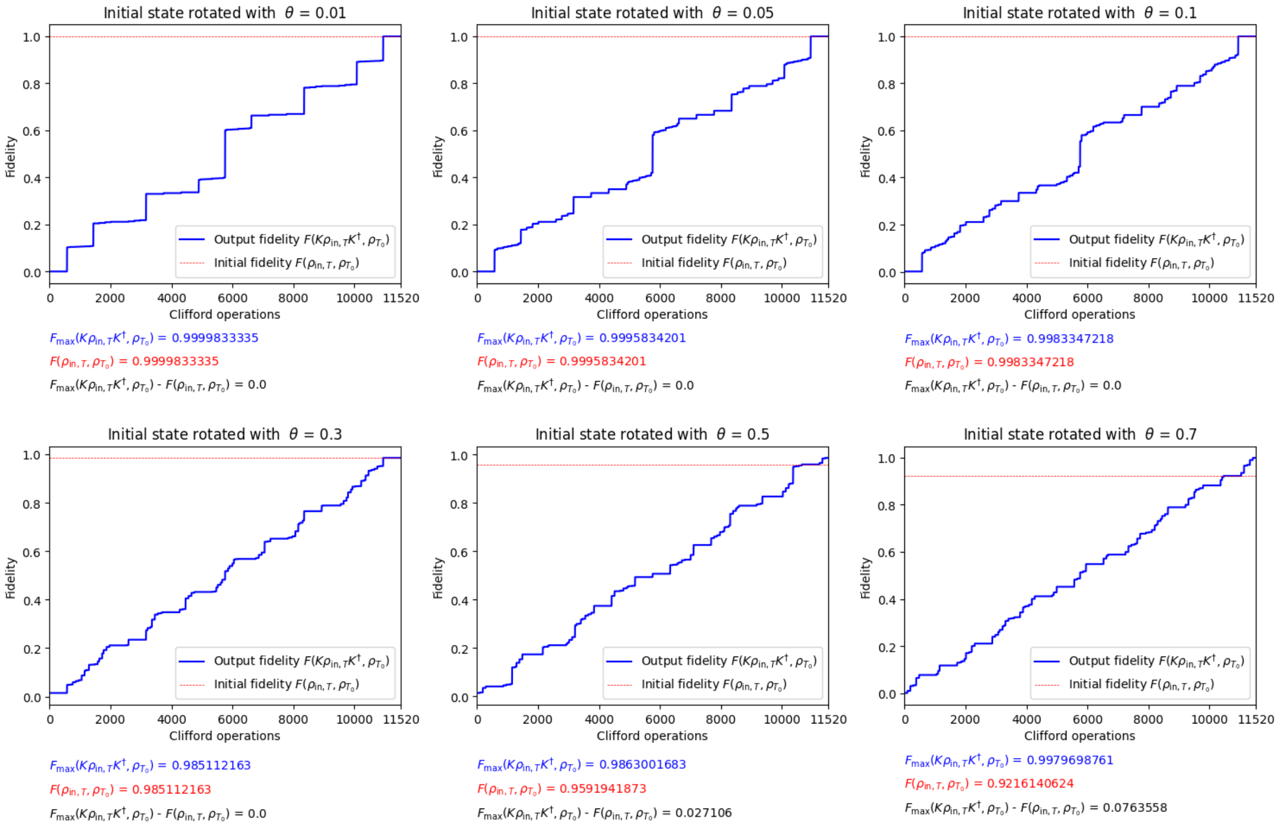


Figure B.1: Similar to Figure 3.2, these plots show the fidelities between the target T state and the output states that can be obtained by evolving the initial state according to the circuit in Figure 3.1 with each two qubit Clifford operation from $\mathcal{C}_2'$. The initial state will for this case be a y-rotated T state and the six different subplots show the results for six different initial rotation angles. In each sub-figure, the maximally obtained fidelity for that case is written out, alongside the initial fidelity and the difference between these values. The resulting unique output state that corresponds to these fidelities is plotted in Figure B.3.

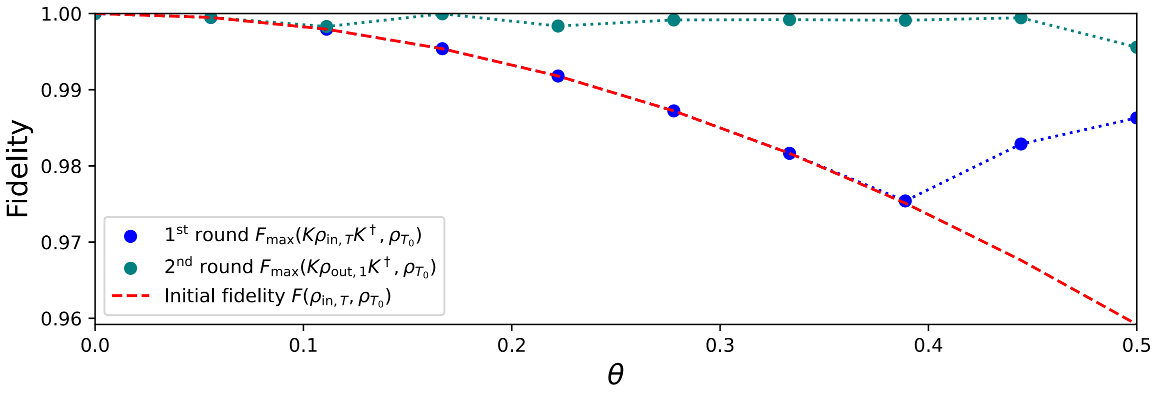
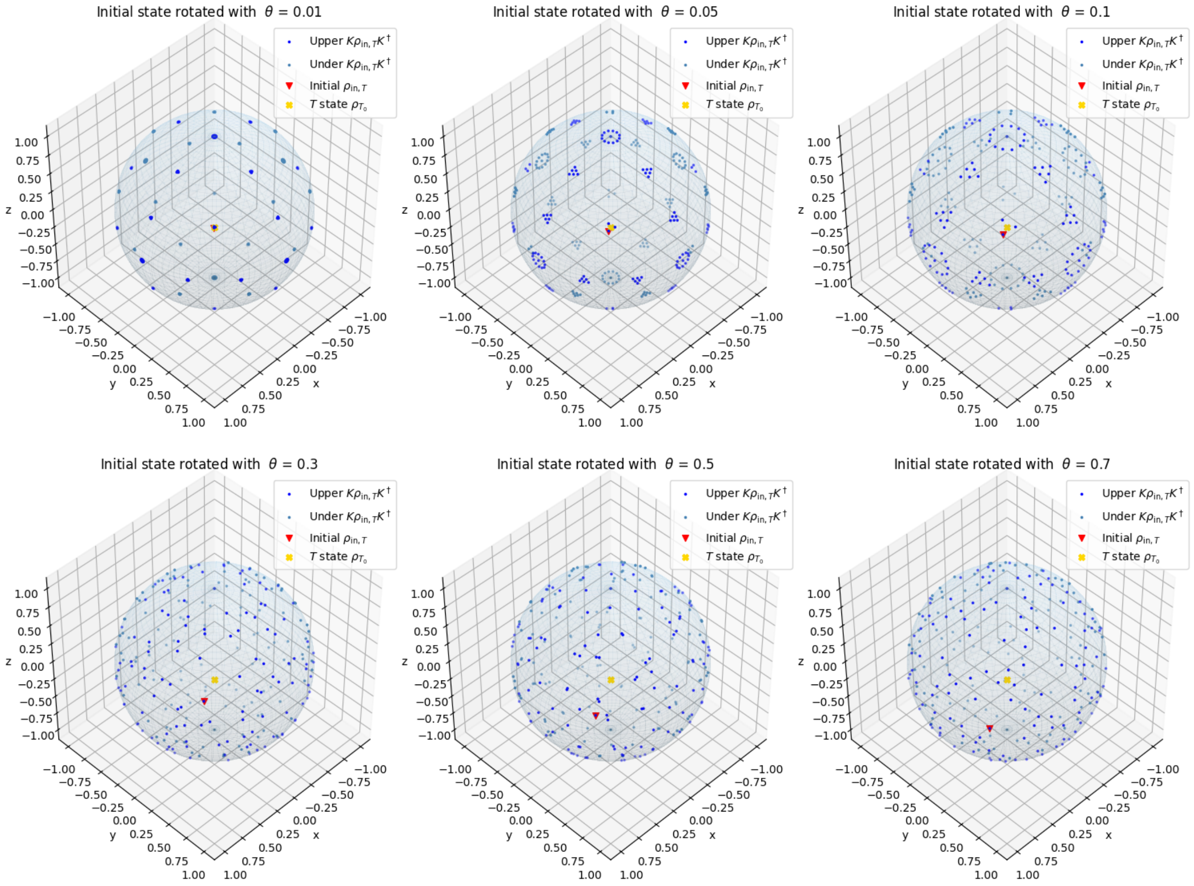


Figure B.2: This plot shows the obtained maximum fidelities when two rounds of optimizations were applied to an input state that was a y-rotated T state and the target state was the ρ_{T_0} state from Eq. (3.2). Similar to the results in Chapter 4, the points in the plot are the actual obtained values, while the dotted line has been added for visualization purposes. Ten different initial rotations were used to obtain these results between 0 and 0.5, and they are summarized in Appendix E.2.2. Three examples of all the unique states that could be obtained for two rounds of optimization when the input state was a T state that was individually rotated with three different initial rotation angles are additionally shown in Figure B.4.



Initial state rotated with θ = 0.01

Initial state rotated with θ = 0.05

Initial state rotated with θ = 0.1

Initial state rotated with θ = 0.3

Initial state rotated with θ = 0.5

Initial state rotated with θ = 0.7

Figure B.3: All the states that can be obtained by evolving the initial y-rotated T state with each two-qubit Clifford operation plotted on the Bloch spheres. These subplots show the states for six different initial rotation angles that correspond to those used in Figure B.1. The upper and under points are defined in the same manner as the states in the leftmost Bloch sphere in Figure 3.4.

1st round rotated with $\theta = 0.05$

2nd round rotated with $\theta = 0.05$

1st round rotated with $\theta = 0.25$

2nd round rotated with $\theta = 0.25$

1st round rotated with $\theta = 0.5$

2nd round rotated with $\theta = 0.5$

Figure B.4: All the unique states that can be obtained for up to two rounds of optimization and three different initial angles that y rotates the input T states have been plotted on Bloch spheres. Each column is a different round of optimization and each row corresponds to a different rotation angle. The states have been shifted to the same hextant as previous examples.

B.1.2 H state rotation results

In a similar vein to the previous section, we will here present additional results that were obtained when a rotated H state was considered as input to the MMS algorithm. Figure B.5 shows the maximum fidelities that were obtained for two rounds of optimization, while Figures B.6 and B.7 displays all the unique states that can be obtained for different initial rotation angles.

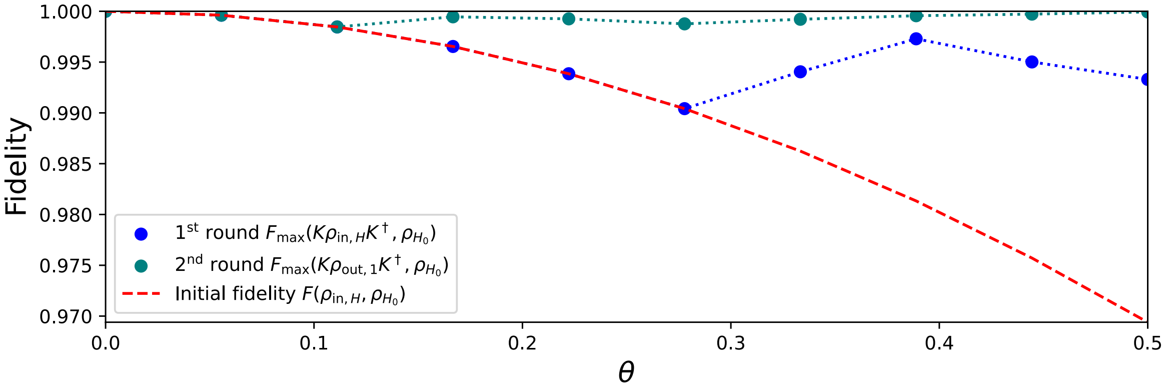


Figure B.5: Similar to the results for the T state in Figure B.2, this plot shows the maximum fidelity for two rounds of optimization with a rotated initial state, but the initial state is now a H state and the target is ρ_{H_0} from Eq. (3.1). These points were obtained for the same initial rotations as in Figure B.2. Similarly, examples are given in Figure B.7 of all the unique states that can be obtained for two rounds of optimization and three different initial rotation angles.

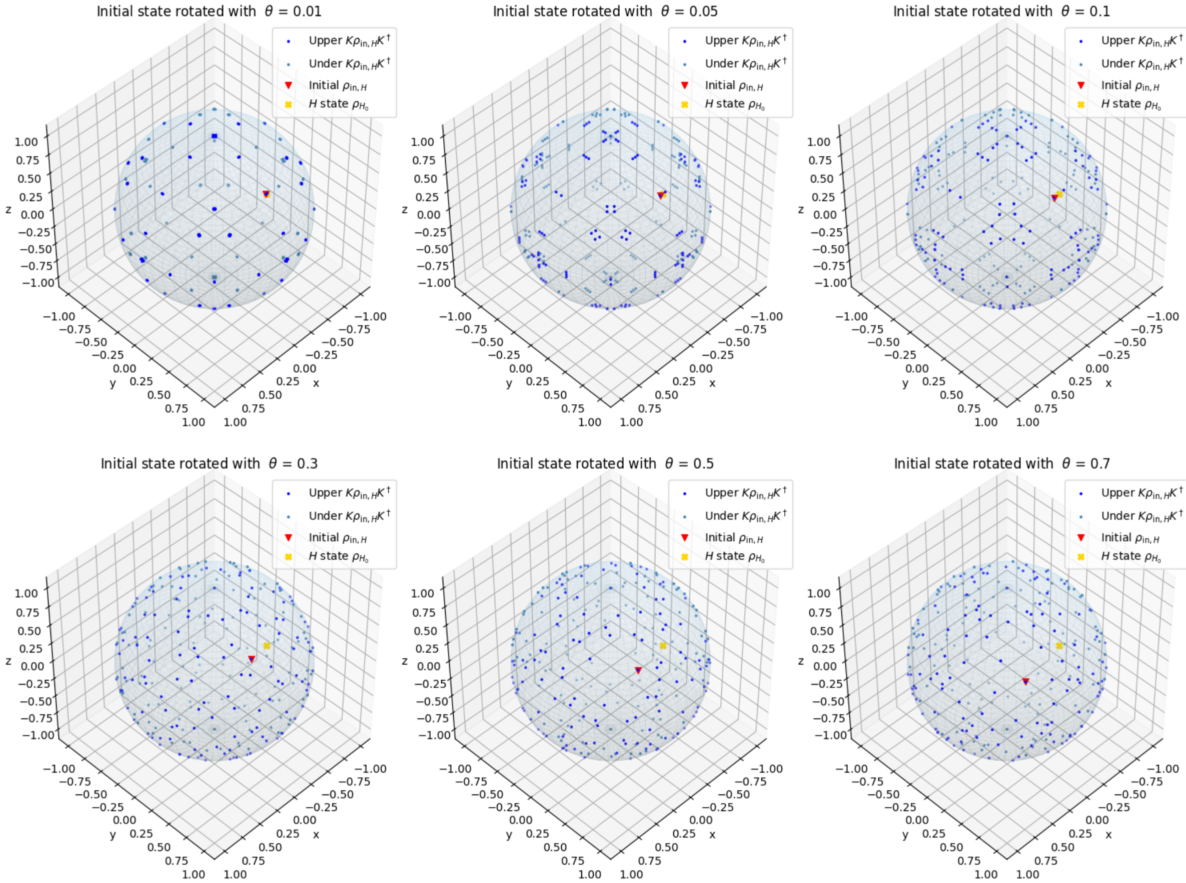


Figure B.6: In the same vein as the results in Figure B.3, these plots show all the states that can be obtained by evolving the initial y-rotated H state with each two-qubit Clifford gates. The same rotation angles for the initial state as in Figure B.3 have been considered.

Figure B.7: As for the T state case in Figure B.4, these plots show the obtainable unique states for up to two rounds of optimization and three different rotation angles of the initial state. The initial state is however for this case a rotated H state.

B.2 Mixed input state results

The results that were obtained for the case of a mixed input state were the main results of this thesis. In the coming section, we will present extra results that visualize the distillation process by plotting the possible output state and additional examples of sequences of Clifford gates that give the maximum obtained fidelity.

B.2.1 Mixed T state results

Here are the unique output states that can obtained for up to four rounds of optimization with two different mixed input T states.

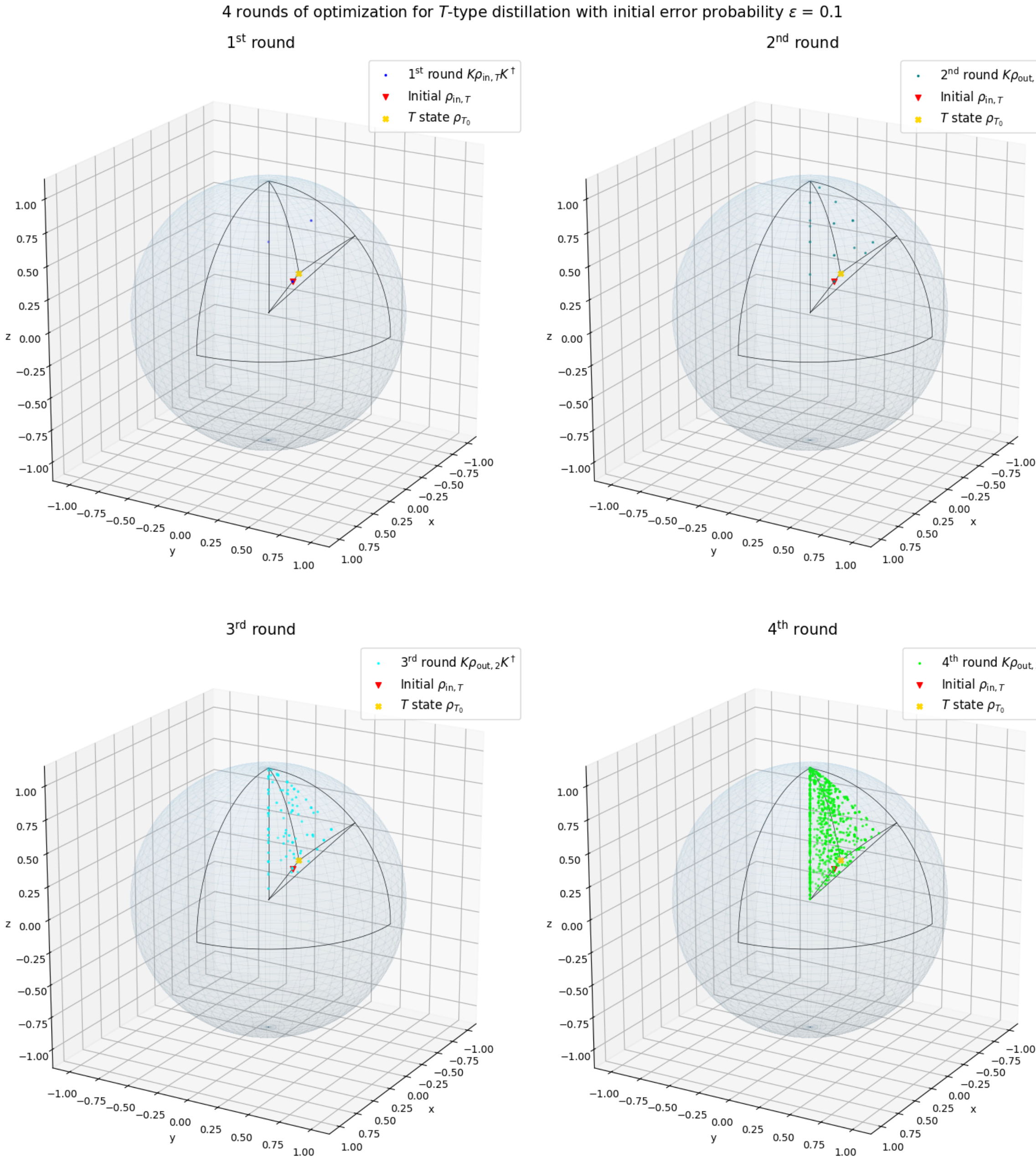


Figure B.8: All the unique mixed states that can be obtained for one to four rounds of optimization are individually plotted on four different Bloch spheres. The input state that was considered when obtaining these states was a T state that where mixed with an initial error probability of $\varepsilon = 0.1$ and was initiated to be in the same form as the input state from Eq. (3.5). It is also important to note that all the unique states from each of the optimization rounds have been shifted into the hextant which is highlighted in Figure A.3.

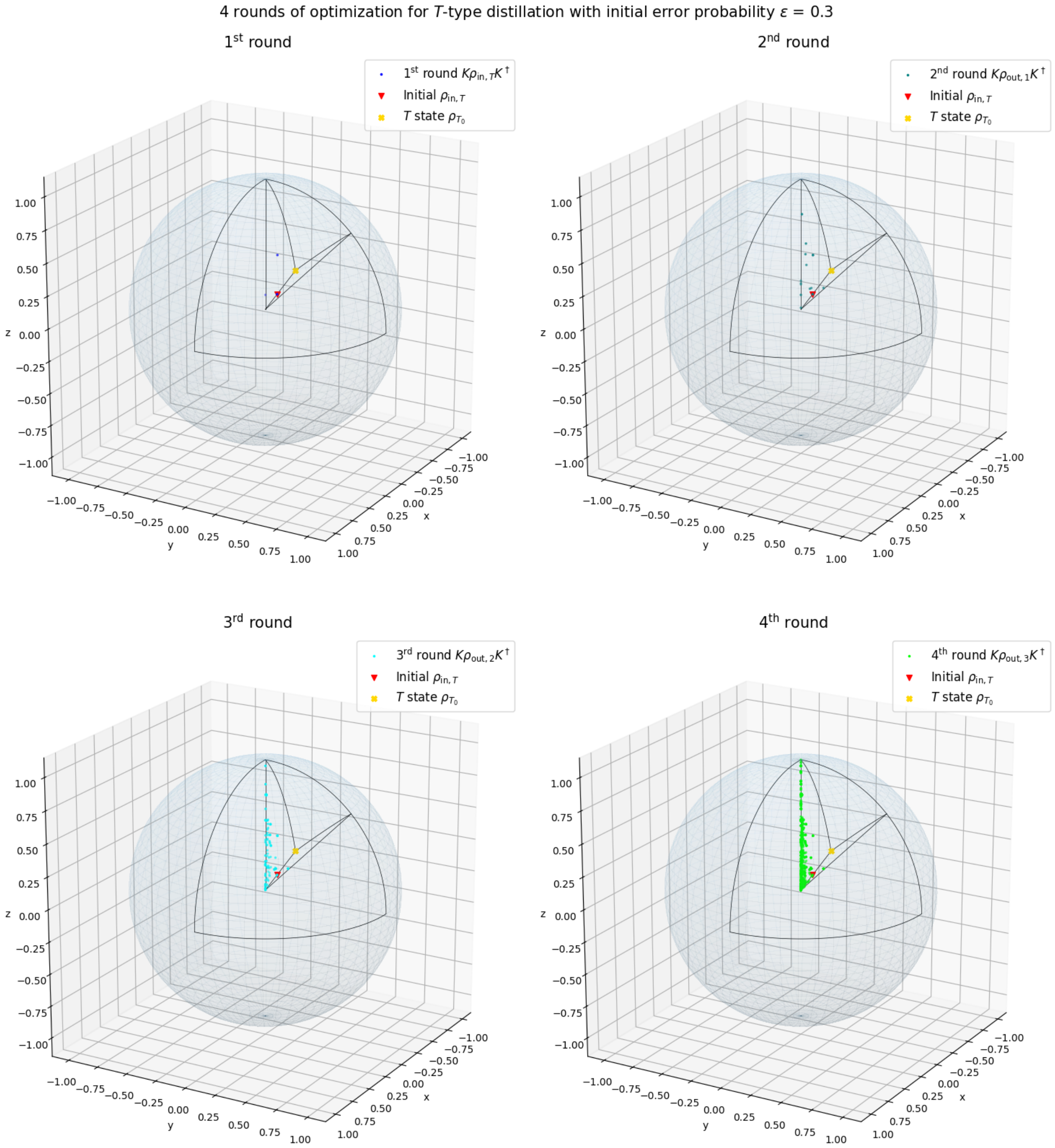


Figure B.9: Similar to Figure B.8, these figures show all the unique output states plotted on the Bloch sphere that can be obtained for four rounds of optimization, but these plots were obtained when the initial state is mixed with $\varepsilon = 0.3$. The input state was similarly on the form of Eq. (3.5) and all the states have been shifted into the same hextant.

B.2.1.1 Clifford sequences for T-type distillation when $\varepsilon > \varepsilon^*$

In Section 4.2.1, we gave an example of a series of Clifford gates that the algorithm found for the T-type distillation when the initial error probability was higher than the threshold. However, in contrast to the corresponding results that were obtained for H-type distillation, the algorithm does not simply use the same gate for all the rounds and all inputs in this regime. Rather the algorithm uses a few different Clifford gates and the sequences might vary for different mixed states when the input T state is located inside of the octahedron. Here are two additional examples of Clifford gate sequences that provided the maximum fidelity for two different input states that were mixed with either $\varepsilon = 0.22$ or $\varepsilon = 0.4$.

Sequences of optimal Clifford operations when $\varepsilon = 0.22$:

$$1^{\text{st}} \text{ optimization round: } K_0 \rho_{\text{in},T} K_0^\dagger = \begin{pmatrix} 0.32332 \\ 0.32332 \\ 0.32332 \end{pmatrix},$$

$$2^{\text{nd}} \text{ optimization round: } K_0 K_0 \rho_{\text{in},T} K_0^\dagger K_0^\dagger = \begin{pmatrix} 0.32332 \\ 0.32332 \\ 0.32332 \end{pmatrix},$$

$$3^{\text{rd}} \text{ optimization round: } K_{5760} K_{5760} K_{128} \rho_{\text{in},T} K_{128}^\dagger K_{5760}^\dagger K_{5760}^\dagger = \begin{pmatrix} 0 \\ 0.00040 \\ 0.99069 \end{pmatrix},$$

$$4^{\text{th}} \text{ optimization round: } K_{5760} K_{5760} K_{5760} K_{128} \rho_{\text{in},T} K_{128}^\dagger K_{5760}^\dagger K_{5760}^\dagger K_{5760}^\dagger = \begin{pmatrix} 0 \\ 0 \\ 0.99996 \end{pmatrix}.$$

Sequences of optimal Clifford operations when $\varepsilon = 0.4$:

$$1^{\text{st}} \text{ optimization round: } K_0 \rho_{\text{in},T} K_0^\dagger = \begin{pmatrix} 0.11547 \\ 0.11547 \\ 0.11547 \end{pmatrix},$$

$$2^{\text{nd}} \text{ optimization round: } K_{5760} K_{128} \rho_{\text{in},T} K_{128}^\dagger K_{5760}^\dagger = \begin{pmatrix} 0 \\ 0.00066 \\ 0.43330 \end{pmatrix},$$

$$3^{\text{rd}} \text{ optimization round: } K_{5760} K_{5760} K_{128} \rho_{\text{in},T} K_{128}^\dagger K_{5760}^\dagger K_{5760}^\dagger = \begin{pmatrix} 0 \\ 0 \\ 0.72961 \end{pmatrix},$$

$$4^{\text{th}} \text{ optimization round: } K_{5760} K_{5760} K_{5760} K_{128} \rho_{\text{in},T} K_{128}^\dagger K_{5760}^\dagger K_{5760}^\dagger K_{5760}^\dagger = \begin{pmatrix} 0 \\ 0 \\ 0.95229 \end{pmatrix}.$$

B.2.2 Mixed H state results

Similar to the previous section, we will here show examples of all the unique states that can be obtained when considering two different initial error probabilities and up to four rounds of optimization for H-type distillation.

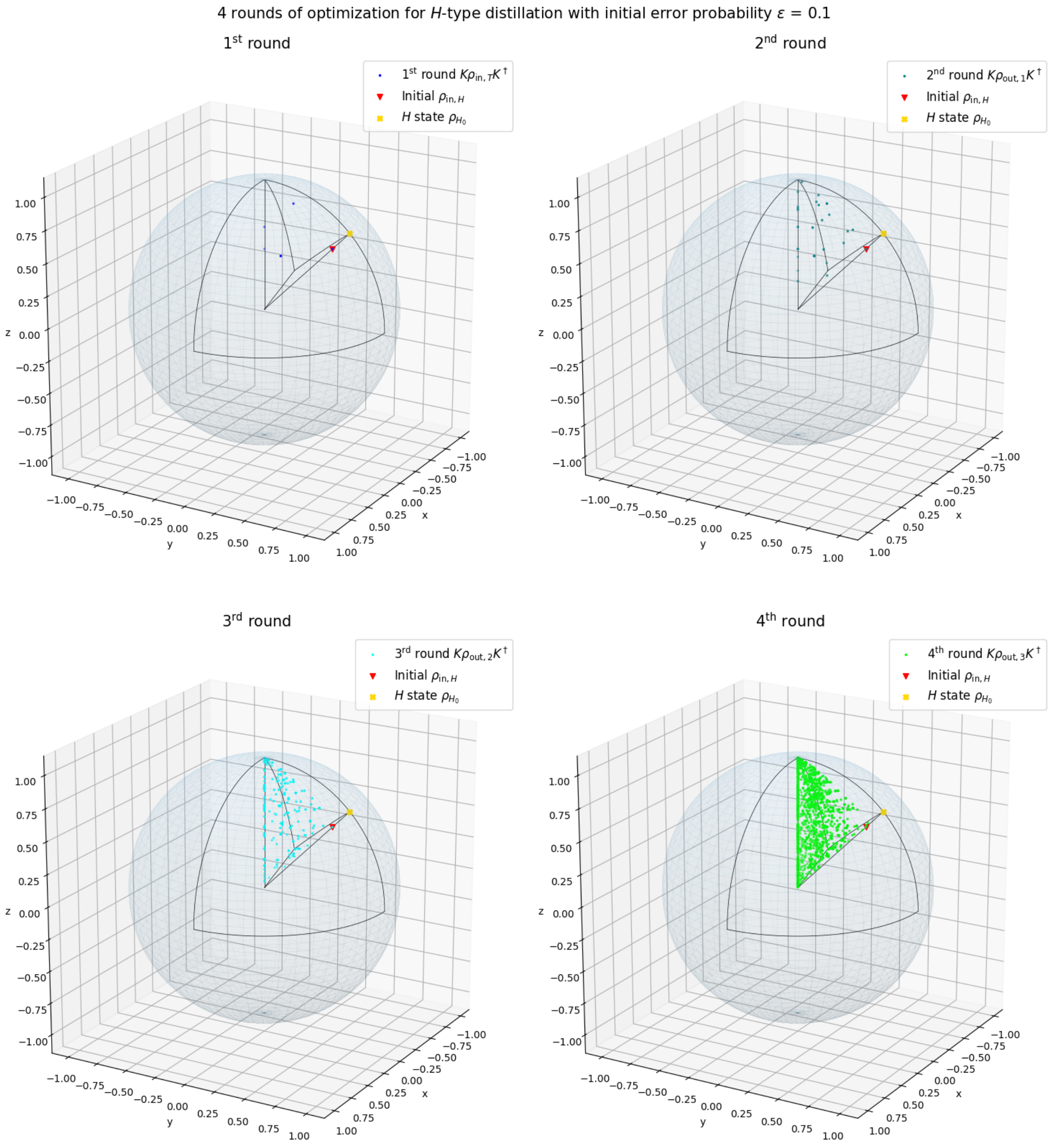


Figure B.10: In these four plots, we show all the unique output states for one to four rounds of optimization when the initial H state was mixed with the initial error portability of 0.1. Similar to many of the previous results, these states have been transformed to the same ordered hextant

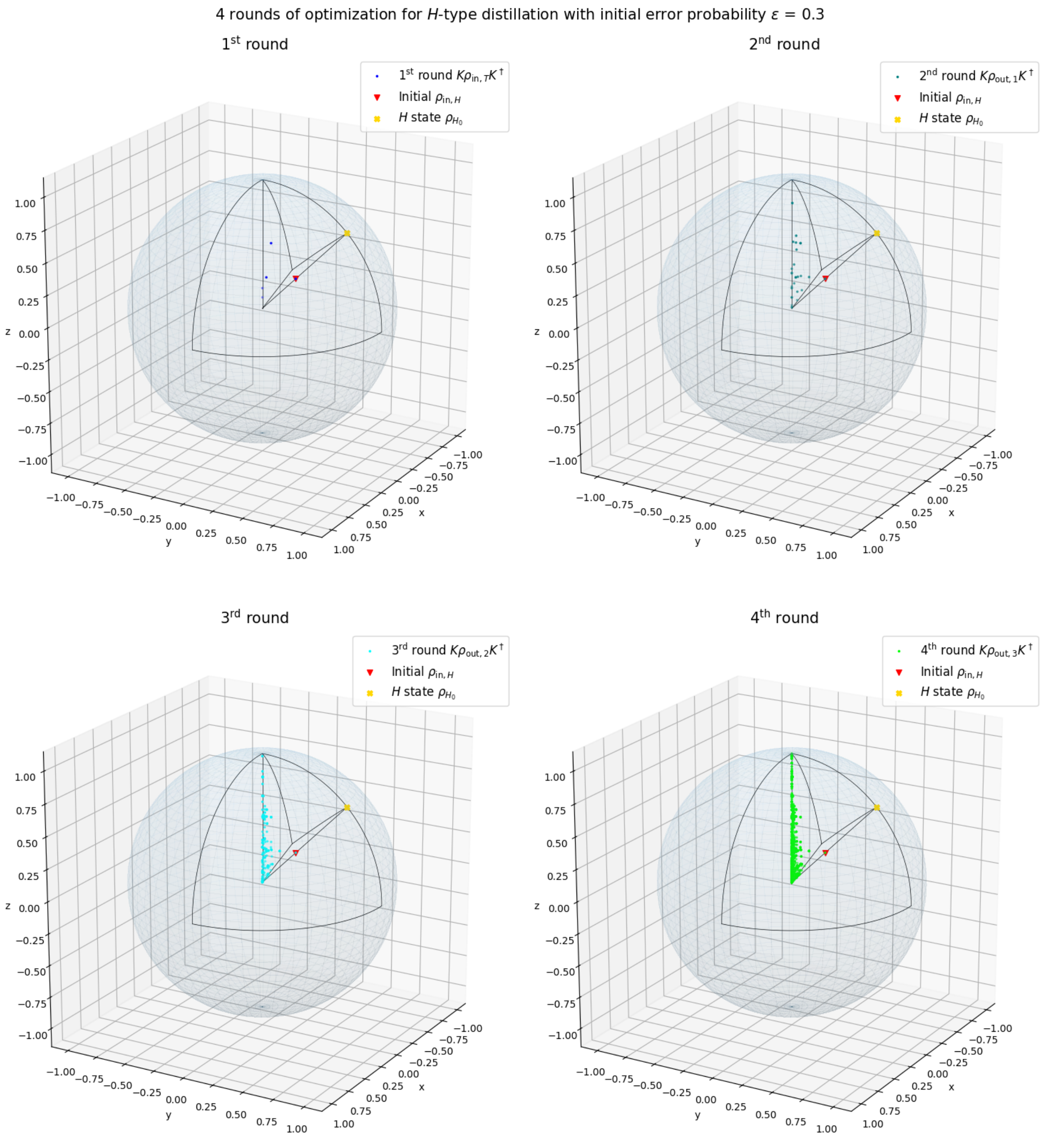


Figure B.11: Theses plots additionally show the output states for up to four rounds of optimization for H-type distillation, similar to Figure B.10, but the initial error probability is now equal to 0.3.

B.2.2.1 Clifford sequences for H-type distillation when $\varepsilon < \varepsilon^*$

Here are two additional chains of Clifford operations that provided the maximum fidelities for two different initial H states that were mixed with an initial error probability that was smaller than the threshold error probability.

Sequences of optimal Clifford operations when $\varepsilon = 0.02$:

$$1^{\text{st}} \text{ optimization round: } K_{256}\rho_{\text{in},H}K_{256}^{\dagger} = \begin{pmatrix} 0 \\ 0.67882 \\ 0.67882 \end{pmatrix},$$

$$2^{\text{nd}} \text{ optimization round: } K_{5841}K_{5760}\rho_{\text{in},H}K_{5760}^{\dagger}K_{5841}^{\dagger} = \begin{pmatrix} 0 \\ 0.57379 \\ 0.78559 \end{pmatrix},$$

$$3^{\text{rd}} \text{ optimization round: } K_{256}K_{5841}K_{5760}\rho_{\text{in},H}K_{5760}^{\dagger}K_{5841}^{\dagger}K_{256}^{\dagger} = \begin{pmatrix} 0 \\ 0.57379 \\ 0.78559 \end{pmatrix},$$

$$4^{\text{th}} \text{ optimization round: } K_{5841}K_{5841}K_{5776}K_{5760}\rho_{\text{in},H}K_{5760}^{\dagger}K_{5776}^{\dagger}K_{5841}^{\dagger}K_{5841}^{\dagger} = \begin{pmatrix} 0 \\ 0.65495 \\ 0.72725 \end{pmatrix}.$$

Sequences of optimal Clifford operations when $\varepsilon = 0.03$:

$$1^{\text{st}} \text{ optimization round: } K_{256}\rho_{\text{in},H}K_{256}^{\dagger} = \begin{pmatrix} 0 \\ 0.66468 \\ 0.66468 \end{pmatrix},$$

$$2^{\text{nd}} \text{ optimization round: } K_{5841}K_{5760}\rho_{\text{in},H}K_{5760}^{\dagger}K_{5841}^{\dagger} = \begin{pmatrix} 0 \\ 0.56024 \\ 0.77714 \end{pmatrix},$$

$$3^{\text{rd}} \text{ optimization round: } K_{256}K_{5841}K_{5760}\rho_{\text{in},H}K_{5760}^{\dagger}K_{5841}^{\dagger}K_{256}^{\dagger} = \begin{pmatrix} 0 \\ 0.56024 \\ 0.77714 \end{pmatrix},$$

$$4^{\text{th}} \text{ optimization round: } K_{5841}K_{5841}K_{5776}K_{5760}\rho_{\text{in},H}K_{5760}^{\dagger}K_{5776}^{\dagger}K_{5841}^{\dagger}K_{5841}^{\dagger} = \begin{pmatrix} 0 \\ 0.65069 \\ 0.71427 \end{pmatrix}.$$

Appendix C

Tracing out modular magic synthesis

In addition to studying the modular magic synthesis algorithm, which was presented in Chapter 3, we also investigated a deterministic version of MMS that traced out the second qubit of each round instead of post-selecting. To differentiate these algorithms, this variant of the MMS algorithm will henceforth be called tracing out modular magic synthesis (TMMS). Despite this change, TMMS will still share all other implementation characteristics with MMS; TMMS will use two input states for each round, the states will only be evolved with two-qubit Clifford operations and the operations used might vary between rounds such that the fidelity of the output state to the target becomes optimized.

This chapter will be initiated with a description of how the TMMS algorithm was implemented, which is done similarly to the introduction of MMS in Chapter 3. This description is then followed by a presentation and a corresponding discussion of the obtained results, where both rotated and mixed input states were considered.

C.1 Implementation of the TMMS algorithm

In this section, we will present how the TMMS algorithm was implemented. This will begin with a description of one round of the algorithm, which is then followed by an outline of how multiple rounds were implemented.

Similar to MMS, the input for one round of optimization of the TMMS algorithm is two of the same states ρ_{in}, which are initiated by applying a dephasing operation $D\left(\rho_{\text{in}}\right)$ to each state. The input states are then evolved with a Clifford operation $C \in \mathcal{C}_2'$, which will yield a system that is the same as in Eq. (3.6). In contrast to the MMS algorithm, the second qubit will then be traced out in a similar fashion to the calculations in Eq. (2.14), which will yield the output state of the optimization round as

$$\begin{aligned}\rho_{\text{out}} &= \mathrm{Tr}_2\left[D\left(\rho_{\text{in}}\right) \otimes D\left(\rho_{\text{in}}\right)\right] \\ &= \left(I \otimes \langle 0|\right) C\left(D\left(\rho_{\text{in}}\right) \otimes D\left(\rho_{\text{in}}\right)\right) C^{\dagger}\left(I \otimes |0\rangle\right) + \left(I \otimes \langle 1|\right) C\left(D\left(\rho_{\text{in}}\right) \otimes D\left(\rho_{\text{in}}\right)\right) C^{\dagger}\left(I \otimes |1\rangle\right)\end{aligned}$$

These calculations can be visualized as the circuit in Figure C.1. An important effect of tracing out the second qubit is that the TMMS algorithm will be deterministic, in contrast to MMS which is probabilistic.

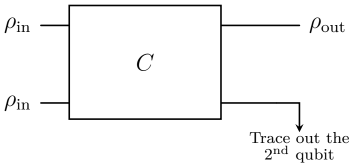


Figure C.1: This circuit depicts one round of optimization of the TMMS algorithm, where $C \in \mathcal{C}_2'$

To investigate the full capabilities of the TMMS algorithm, more rounds of optimization need to be considered, where the output state will become the input for the next round. Two rounds of optimizations are depicted in Figure C.2 to give an impression of how the circuit can be inferred for more rounds. Similar to the study of MMS, we investigated the TMMS algorithm up to four rounds of optimization, where the circuits for three and four rounds are depicted in Appendix D.2.

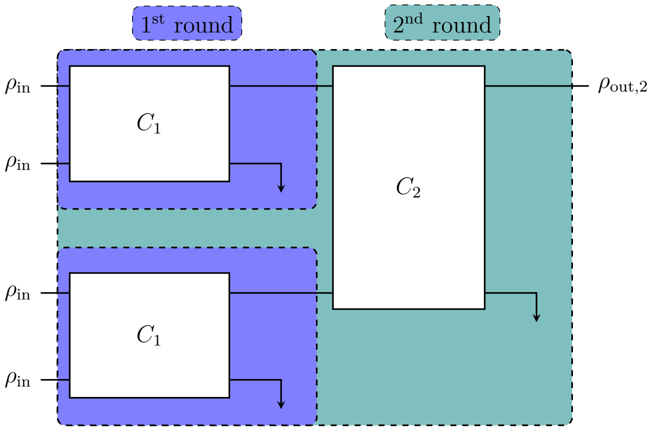


Figure C.2: The corresponding circuit for two rounds of optimization of the TMMS algorithm, where C_1, $C_2 \in \mathcal{C}_2'$

The implementation of more rounds of the TMMS algorithm was, analogously to the case of the MMS algorithm, aided by the abbreviations that are presented in Appendices A.3 and A.4 to reduce the number of output states that needed to be considered after each round.

C.2 Results & discussion of the TMMS algorithm

Analogously to Chapter 4, we will in this section present the results that we obtained by implementing the TMMS algorithm, and provide a discussion regarding the findings. Similar to the results of the MMS algorithm in Section 4.1, the acquired maximum fidelity curves are only approximations of the true continuous curves, since they have been obtained with individual input states that give corresponding discrete output values. Also similar to the previous results, the values of the initial rotation angles and error probabilities that were used are explicitly written in Appendix E.2.

C.2.1 TMMS results

We studied the TMMS algorithm by using both rotated and mixed input states, and these results will be presented and discussed in this section. We only considered the case of H-type distillation, since that showed the most promise from the previous results of the MMS algorithm that are shown in Chapter 4.

First, we applied the TMMS algorithm to several input states that were rotated with different initial angles, similar to the studies that gave the results in Figures B.2 and B.5. This investigation was performed for three rounds of optimization and the fidelity curves we obtained are shown in Figure C.3.

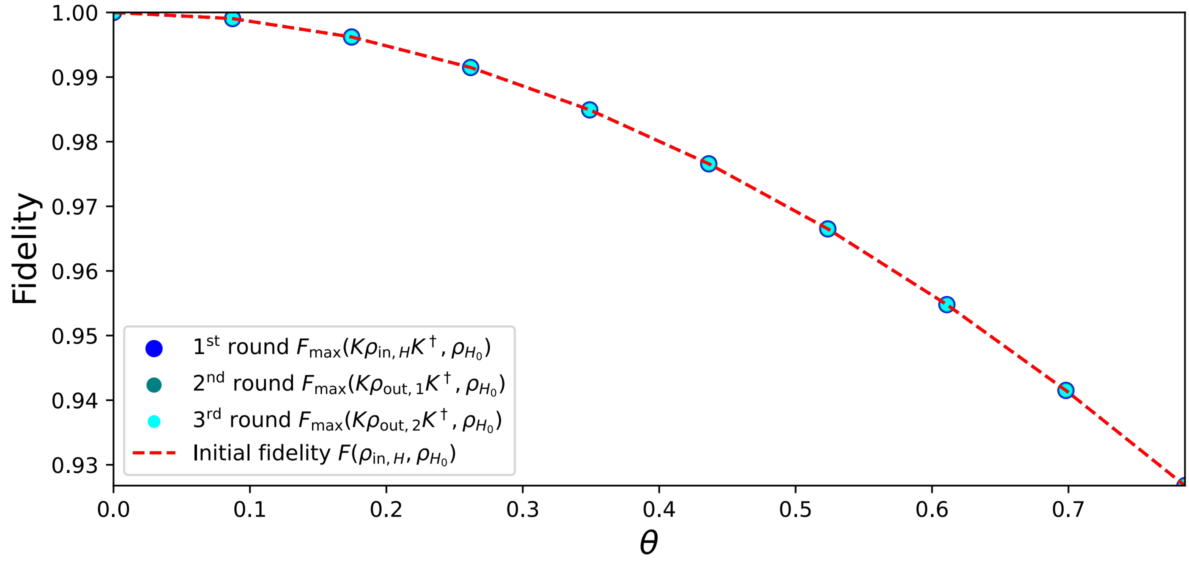


Figure C.3: This plot shows the obtained maximum fidelities for ten initial rotation angles and three rounds of optimization, which were implemented with the circuit that is shown in Figure D.3. In addition to the dots that mark the obtained fidelity values for the three rounds, the fidelity between the target T state and the input is also shown with the red dashed line.

In addition to the maximum fidelity that we obtained for the case of a rotated input state, we also investigated the scenario when the initial input was a mixed state on the form of that state in Eq. (3.4). We obtained maximum fidelities for four rounds of optimization and they are displayed in Figure C.4.

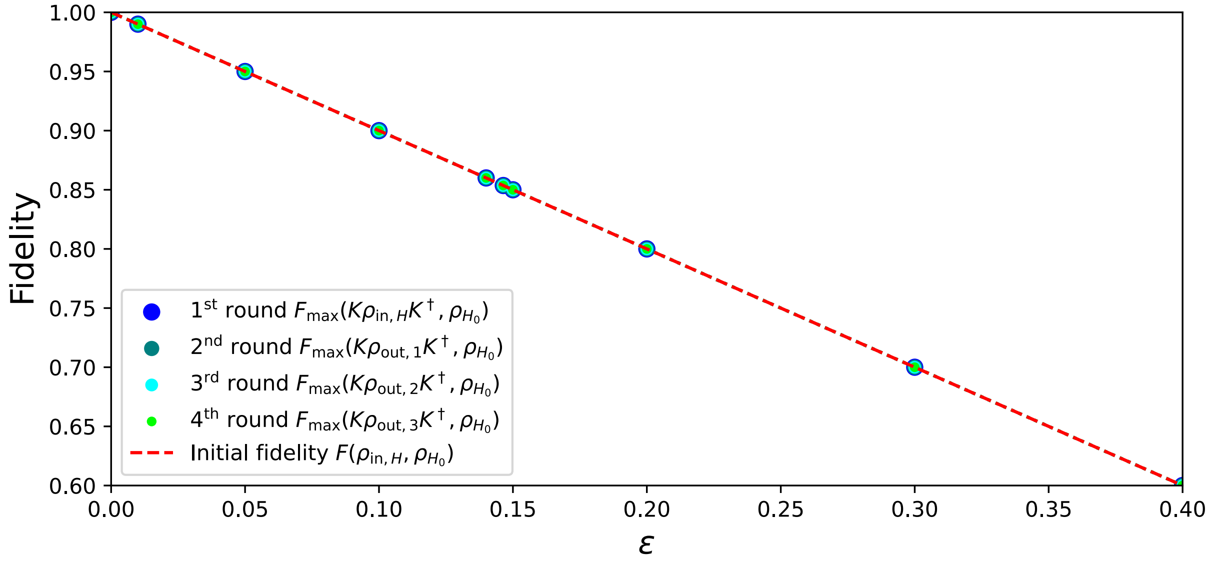


Figure C.4: In this plot, we show the maximum fidelities that were obtained for four rounds of the TMMS algorithm for different initial error probabilities, similar to the results in Figures 4.1 and 4.2, with the corresponding initial fidelity. The ten initial error probabilities that were used when we obtained these results are shown in Appendix E.2.1.

C.2.2 TMMS discussion

The discussion of the previously shown results that we obtained by implementing the TMMS algorithm will now be given. These fidelity plots will be studied in comparison to the findings of Bravyi & Kitaev and the results that were acquired by implementing the MMS algorithm, which is presented and discussed in Chapter 4 and the Appendix B. We will start this study by comparing the results that were obtained when a rotated input state was used in the TMMS algorithm, which is shown in Figure C.3, and the obtained maximum fidelity curves from Figures B.5. This is then followed by an investigation of the case when a mixed input was considered for the TMMS algorithm in Figure C.4 and the derived output fidelity curves from Bravyi & Kitaev, which are shown in Figure 2.7, and the maximum fidelity curves from the implementation of the MMS algorithm that are shown in Figure 4.2.

One of the most important differences that can be seen by comparing the rotation results from Figures C.3 and B.5, is the MMS algorithm was able to find a fidelity that was higher than the initial fidelity for most initial rotations, but the TMMS algorithm could not find a better fidelity for any input angle. Additionally, the MMS algorithm was able to increase the fidelity for smaller initial rotations when more rounds were implemented, while the maximum fidelity that the TMMS algorithm could produce where always equal to the initial fidelity for all optimization rounds.

Similar to the rotation results, the maximum fidelity curves in Figure C.4 appear to analogously be the same as the input fidelity curve for all initial error probabilities. This is in contrast to the derived output fidelity curves from Bravyi & Kitaev and the maximum fidelity curves from the implementation of the MMS algorithm, where these results show that the respective algorithm was able to produce an output state that was closer to the target compared to the input.

The fact that the TMMS algorithm was not able to increase fidelity for either a rotated input or a mixed input, while the implementation of this algorithm is similar to that of the MMS algorithm which could increase the fidelity for some input states, can be explained by considering how these algorithms differ. The distinctive characteristic of these algorithms is that in TMMS we apply a partial trace to the second qubit, while we instead perform a post-selection measurement in MMS. These results highlight the powerful benefits of using post-selection, which effectively corresponds to artificially picking the desired outcome independent of the probability of obtaining that result, while partial trace is equivalent to measuring a qubit and then discarding the outcome [21],[36]. A clear example that highlights the uncertainty created by performing a partial trace is that the remaining output state most often will be a mixed state, as can for example be seen from the results of the calculations in Eq. (2.14), while the remaining state can be a pure state after post-selection, as is the case of Eq. (2.16). Furthermore, since no optimization round of the TMMS could improve the fidelity, we conclude that more rounds most likely would also be insufficient to obtain an output state with increased fidelity compared to the input state.

Appendix D

Extended quantum circuits

In this appendix, we present the complete versions of the circuits used when implementing and studying the MMS and TMMS algorithms.

D.1 Circuits for MMS

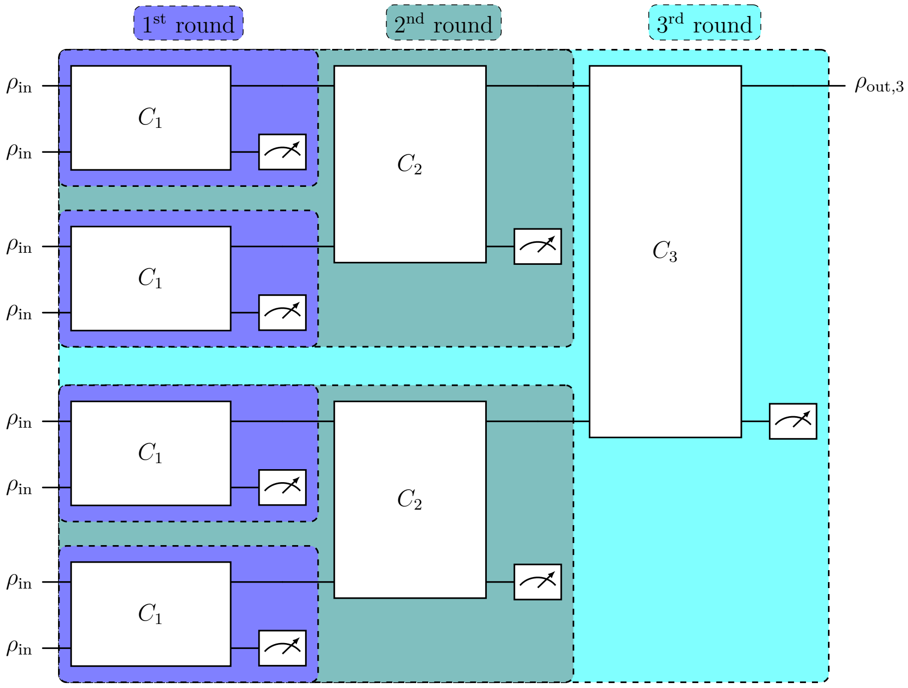


Figure D.1: This is the circuit for three rounds of optimization of the MMS algorithm, which requires eight input states and where C_1, C_2, $C_3 \in \mathcal{C}'_2$.

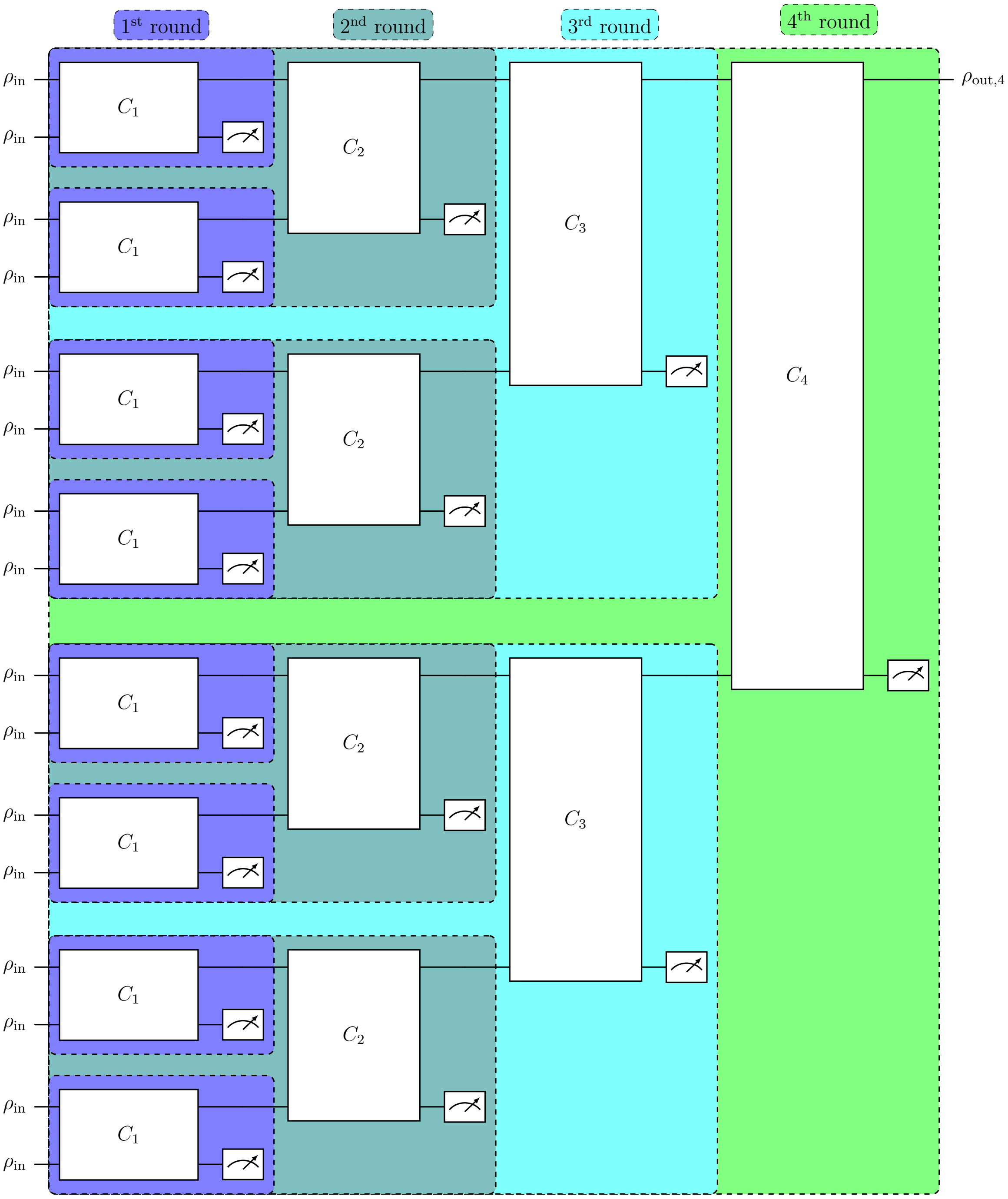


Figure D.2: This is the complete circuit that was used when we implemented four rounds of optimization of the MMS algorithm. The circuit was for example used to obtain the results in Figures 4.1 and 4.2. Sixteen input states are needed for this optimization, where C_1, C_2, C_3, $C_4 \in \mathcal{C}'_2$.

D.2 Circuits for TMMS

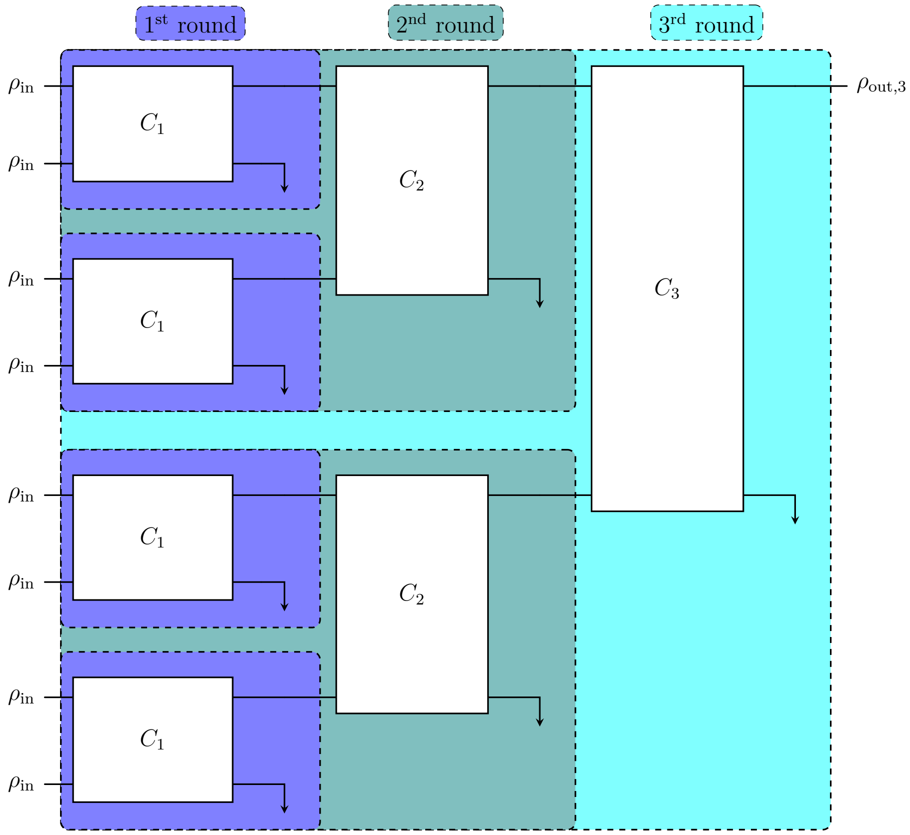


Figure D.3: Correspondingly to the circuit in Figure D.1, this is the quantum circuit which illustrates how three rounds of the TMMS algorithm are implemented, where C_1, C_2, $C_3 \in \mathcal{C}_2'$.

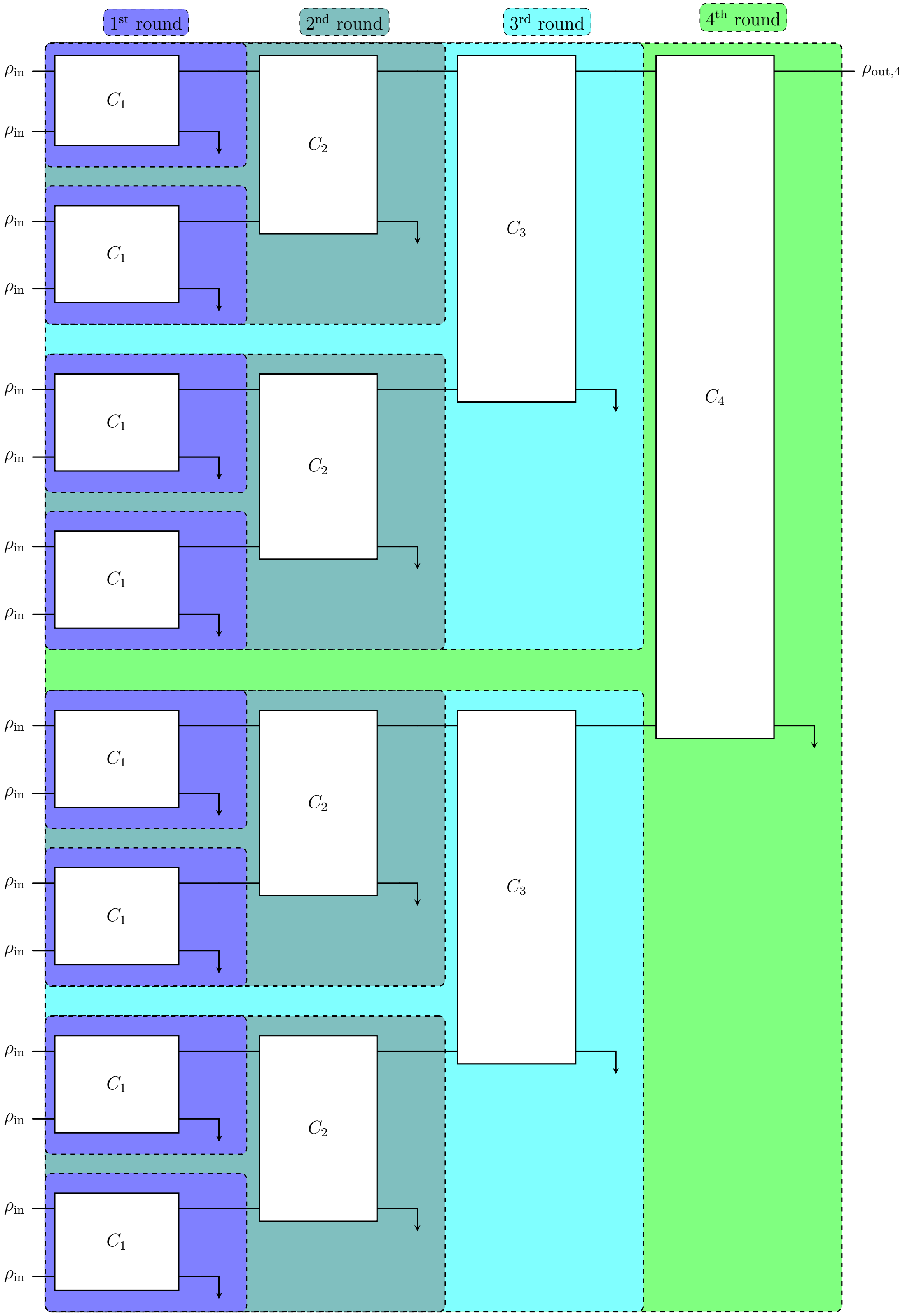


Figure D.4: The circuit in this figure shows how the full four rounds of the TMMS algorithm is implemented, where C_1, C_2, C_3, $C_4 \in \mathcal{C}_2'$.

Appendix E

Tables and lists

We summarize in this appendix the table of single-qubit Clifford operations and lists of the initial error probabilities and rotations angles that were used to obtain the presented results.

E.1 Single-qubit Clifford operations

In Section 2.2.1, we mention that the single-qubit Clifford quotient group can be generated by the Hadamard and the phase gate. Furthermore, in Section 2.1.5, we saw that this group will consist of twenty-four operators. An example of the members of this group is summarized in Table E.1 up to a global phase.

$C \in \mathcal{C}_1'$	$C\sigma^x C^\dagger$	$C\sigma^y C^\dagger$	$C\sigma^z C^\dagger$	$C \in \mathcal{C}_1'$	$C\sigma^x C^\dagger$	$C\sigma^y C^\dagger$	$C\sigma^z C^\dagger$
I	σ^x	σ^y	σ^z	HS	$-\sigma^y$	$-\sigma^z$	σ^x
σ^x	σ^x	$-\sigma^y$	$-\sigma^z$	$HS\sigma^x$	$-\sigma^y$	σ^z	$-\sigma^x$
σ^y	$-\sigma^x$	σ^y	$-\sigma^z$	$HS\sigma^y$	σ^y	$-\sigma^z$	$-\sigma^x$
σ^z	$-\sigma^x$	$-\sigma^y$	σ^z	$HS\sigma^z(L^2)$	σ^y	σ^z	σ^x
H	σ^z	$-\sigma^y$	σ^x	$SH(L)$	σ^z	σ^x	σ^y
$H\sigma^x$	σ^z	σ^y	$-\sigma^x$	$SH\sigma^x(L\sigma^x)$	σ^z	$-\sigma^x$	$-\sigma^y$
$H\sigma^y$	$-\sigma^z$	$-\sigma^y$	$-\sigma^x$	$SH\sigma^y(L\sigma^y)$	$-\sigma^z$	σ^x	$-\sigma^y$
$H\sigma^z$	$-\sigma^z$	σ^y	σ^x	$SH\sigma^z(L\sigma^z)$	$-\sigma^z$	$-\sigma^x$	σ^y
S	σ^y	$-\sigma^x$	σ^z	$SHS(LS)$	σ^x	$-\sigma^z$	σ^y
$S\sigma^x$	σ^y	$-\sigma^x$	$-\sigma^z$	$SHS\sigma^x(LS\sigma^x)$	σ^x	σ^z	$-\sigma^y$
$S\sigma^y$	$-\sigma^y$	$-\sigma^x$	$-\sigma^z$	$SHS\sigma^y(LS\sigma^y)$	$-\sigma^x$	$-\sigma^z$	$-\sigma^y$
$S\sigma^z$	$-\sigma^y$	σ^x	σ^z	$SHS\sigma^z(LS\sigma^z)$	$-\sigma^x$	σ^z	σ^y

Table E.1: Summarization of one example of all the single-qubit Clifford quotient operators and how they act on the Pauli matrices. These operators were generated by first creating all the possible single qubit tableaux and then utilizing the Qiskit [34] Python function which converted the tableaux to matrices. Note that this list of Clifford operators is arbitrarily chosen and there are numerous other possible single-qubit Clifford operators. However, these other operators will only differ by a global phase and will therefore act on the Pauli matrices in the same way.

E.2 Steps for initial ε and θ

Some of the results in Chapter 4 and Appendices B and C.2.1 were obtained by varying the error probability or rotation angle of the initial state over a specified range to acquire a curve that approximately describes how the maximum fidelity depends on the input state. To offer more detail on how these plots were obtained, the individual input error probability or rotational angle values that were employed are summarized in this section.

E.2.1 Initial ε

These are the initial error probabilities that were used to obtain the individual output fidelities and corresponding success probabilities that hint at the general structure of the curves in Figures 4.1, 4.2, 4.3, 4.4 and C.4

$$\varepsilon_{\mathrm{in},4.1} \in \{0.0, 0.01, 0.05, 0.1, 0.15, 0.2, \varepsilon_T^*, 0.22, 0.3, 0.4, 0.45, 0.49, 0.499, 0.49999\}\,, \tag{E.1}$$

$$\varepsilon_{\mathrm{in},4.2} \in \{0.0, 0.01, 0.05, 0.1, 0.14, 0.1464, 0.1465, 0.15, 0.2, 0.3, 0.4, 0.45, 0.49999\}\,, \tag{E.2}$$

$$\varepsilon_{\mathrm{in},4.3} \in \{0.0, 0.01, 0.02, 0.03, 0.04, 0.05, 0.06, 0.07, 0.08, 0.09, 0.1, 0.11, 0.12, 0.13, 0.14, 0.1464\}\,, \tag{E.3}$$

$$\varepsilon_{\mathrm{in},4.4} \in \left\{10^{-5}, 10^{-4.5}, 10^{-4}, 10^{-3.5}, 10^{-3}, 10^{-2.5}, 10^{-2}\right\}, \tag{E.4}$$

$$\varepsilon_{\mathrm{in},C.4} \in \{0.0, 0.01, 0.05, 0.1, 0.14, 0.1464, 0.15, 0.2, 0.3, 0.4\}\,. \tag{E.5}$$

E.2.2 Initial θ

Similarly, these are the initial rotation angles that were used to obtain the curves in Figures B.2, B.5, and C.3

$$\theta_{\mathrm{in,B.2}},\, \theta_{\mathrm{in,B.5}} \in \left\{ \frac{1}{2} \cdot \frac{u}{9} \,\middle|\, u = 0, 1, \ldots, 9 \right\}, \tag{E.6}$$

$$\theta_{\mathrm{in},C.3} \in \left\{ \frac{\pi}{4} \cdot \frac{u}{9} \,\middle|\, u = 0, 1, \ldots, 9 \right\}. \tag{E.7}$$

DEPARTMENT OF SOME SUBJECT OR TECHNOLOGY
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden
www.chalmers.se

CHALMERS
UNIVERSITY OF TECHNOLOGY