Numerical Synthesis of Arbitrary Multi-Qubit Unitaries with low T-Count

by

Marc Grau Davis

B.A., University of California, Berkeley (2020)

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

Master of Science

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

February 2023

Author .....
Department of Electrical Engineering and Computer Science
January 20, 2023

Certified by .....
Dirk Robert Englund
Associate Professor of Electrical Engineering and Computer Science
Thesis Supervisor

Accepted by .....
Leslie A. Kolodziejski
Professor of Electrical Engineering and Computer Science
Chair, Department Committee on Graduate Students

Numerical Synthesis of Arbitrary Multi-Qubit Unitaries with low T-Count

by

Marc Grau Davis

Submitted to the Department of Electrical Engineering and Computer Science
on January 20, 2023, in partial fulfillment of the
requirements for the degree of
Master of Science

Abstract

Quantum gate synthesis based on numerical optimization produces efficient circuits for NISQ (Noisy Intermediate-Scale Quantum) computing by minimizing the number of two-qubit gates. The requirements for fault tolerant quantum computing are significantly different in that some single qubit gates require magic state distillation and gate teleportation, which are resource intensive. Here, We propose an approach to adapt numerical optimization to error corrected quantum circuits by using sequential two-pass multistart numerical optimizaton to reduce the number of R_Z gates that must be approximated with Clifford+T circuits. This technique allows NISQ synthesis based on numerical optimization to be applied to fault-tolerant circuits as well.

Thesis Supervisor: Dirk Robert Englund
Title: Associate Professor of Electrical Engineering and Computer Science

Acknowledgments

First I would like to thank my advisor, Professor Dirk Englund, for his assistance and suggestions with this project. I would also like to thank Dr. Hyeongrak Choi, a postdoctoral fellow who provided continuous mentorship and guidance, as well as the BQSKit team from LBNL, especially Ethan Smith and Ed Younis for their collaboration, friendship, and advice. Lastly, I would like to thank my friends and family, especially my mom, for helping to proofread this thesis, and my fiancée, Frances Grimaldo, for her constant support.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Quantum computing is an emerging technology that has the potential for polynomial to exponential speed-ups over classical computing technology. However, current quantum computers are not able to run quantum circuits with a sufficiently large number of qubits and gates to achieve a practical advantage over state-of-the-art classical computers. There has been a two-pronged effort to close this gap. One effort is to improve quantum hardware, such that the implemented quantity and quality of qubits can be increased, while another effort is to advance quantum software, such that the necessary quantity and quality of qubits can be reduced.

Quantum gate synthesis, a step in a quantum compilation stack, is the process of deriving a quantum circuit from a unitary matrix. Synthesis algorithms can be used to generate quantum circuit implementations of small unitaries. Resource-optimizing synthesis algorithms can be used as a powerful optimization step by extracting segments of a large quantum circuit and using a synthesis algorithm to discover more efficient implementations of those extracted segments. Because the target resource for defining efficiency of a quantum circuit is dependent on the context of the target quantum computing architecture, a variety of synthesis approaches are needed to target different scenarios.

One software approach to scaling quantum computation is the use of quantum error correction and fault-tolerance protocols. These techniques allow a quantity-for-quality trade-off in which a large number of low-quality qubits can be used to simulate

a smaller number of higher-quality qubits. Current synthesis techniques that target this scenario are lacking. Most implementations only target certain subsets of 1-qubit gates. In order to use quantum gate synthesis as an optimization technique for fault tolerant quantum computing, it is necessary to synthesize multi-qubit unitaries efficiently.

In this paper we describe a technique for quantum gate synthesis of arbitrary unitaries with low T-count. More specifically, we provide a post-processing step applicable to many numerical synthesis approaches which will minimize T-count when converting to a Clifford+T circuit. This technique enables synthesis of arbitrary multi-qubit unitaries in a form that is useful as an optimization step in fault-tolerant quantum computing contexts.

Chapter 2

Background

2.1 Quantum Logic Gates as Unitary Matrices

A qubit is an idealized two-level quantum system. It is a system with an observable Z that has two eigenstates with two different eigenvalues.

$$Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} |0\rangle \equiv \begin{pmatrix} 1 \\ 0 \end{pmatrix} |1\rangle \equiv \begin{pmatrix} 0 \\ 1 \end{pmatrix} \tag{2.1}$$
$$\langle 0|Z|0\rangle = +1 \, \langle 1|Z|1\rangle = -1$$

Physical systems are not this simple, but it is the goal of quantum hardware development to emulate an ideal qubit as closely as possible. Deviance from the ideal qubit behavior can be modeled as noise applied to an ideal two-level system. This allows us to work out the theory of quantum computation by working with the mathematics of two levels systems. Under these assumptions, quantum computing operations are described mathematically in terms of unitary matrices. The state of n qubits is a statevector consisting of 2^n complex coeficients of the 2^n eigenvectors, corresponding to the potential bit strings that could arise from measuring each qubit along its standard observable as described in equation (2.1). An operation on n qubits can be described as a $2^n \times 2^n$ unitary matrix. Because unitaries and statevectors scale exponentially with the number of qubits involved, it is only possible to use

this formulation to mathematically simulate a small number of qubits at a time on a classical computer. If it were possible to efficiently simulate quantum computers in this way, they would not be able to achieve a computational advantage over classical computers. For these reasons, we will need a way to describe quantum computation without writing down the entire statevectors and unitary matrices involved.

The task of developing hardware quantum computers is to develop quantum systems that emulate ideal two-level systems as closely as possible, and to develop quantum gates to manipulate these two-level systems. Developing a quantum gate is to develop a time-dependent control Hamiltonian with a number of degrees of freedom (DOF) proportional to the number of qubits times the number of time samples equation (2.2).

$$DOF \propto n * (time\ samples) \tag{2.2}$$

As the number of qubits and complexity of the desired operation increase, this problem quickly becomes intractable, especially when considering the difficulties of tuning hardware to minimize noise-causing non-ideal behavior. Instead, it is useful to focus on implementing a small set of quantum gates, known as the gateset, which can be strung together to form quantum circuits, which can implement unitaries not in the gateset. This can be represented mathematically by performing a matrix product between two unitaries representing gates in sequence, and by performing a Kronecker product between two gates in parallel.

Even within this restricted alphabet of controls, compositing the optimal sequence of gates is a critical challenge. In the worst case, it takes $O(4^n)$ gates to implement an n-qubit unitary [18, 20, 23, 27] which nullifies any potential quantum advantage. Therefore, efficient algorithm design that does not rely on general unitaries is crucial for obtaining a computational advantage. Unnecessarily long quantum circuits pose an increased burden on other parts of the quantum pipeline, so quantum circuit optimization techniques are a core component of the quantum software stack. These optimization processes involve scanning a quantum circuit for opportunities to reduce the count of certain gates, thereby making the overall circuit more efficient. Different

Technology	1-Qubit Gate Fidelity	2-Qubit Gate Fidelity	Topology	Source
Superconducting	99.85%	99.66%	restricted	[22, 45]
Trapped Ion	99.5%	97.5%	unrestricted	[55]
Silicon	99%	90%	restricted	[45]
Photonic	99.84%	99.69%	restricted	[48]

Table 2.1: *A summary of various types of quantum hardware. Gate fidelities are compared. Most physical qubits are restricted to nearest-neighbor two-qubit gates according to the topology of the device architecture. Trapped ion qubits are capable of unrestricted two-qubit gates using individual qubit addressing [10, 55].*

approaches to quantum computing evoke different bottlenecks, such that the choice of which gate to minimize is dependent on the details of the quantum computer, such as whether or not quantum error correction is being used.

2.2 Quantum Computing Hardware

There are various physical systems used as qubits, with tradeoffs in gate fidelity, qubit lifespan, qubit connectivity (topology), and the physical gate set. A comparison of the gate fidelity and topology restrictions of several different types of physical qubits is presented in table 2.1.

A superconducting qubit is an LC circuit with a Josephson junction implementing an anharmonic oscillator. The energy levels of a simple harmonic oscillator are evenly spaced, but the use of a Josephson junction shifts the energy levels, making the spacing uneven, and allowing the first two energy levels to be targeted. These two energy levels become the basis states of our qubit. The ground state of the oscillator is $|0\rangle$ and the first excited state is $|1\rangle$ [22, 30]. At low temperature of ~ 100 mK, the thermal energy is lower than the excitation energy of the oscillators so that the qubit will naturally decay to the ground state. Microwave pulses can rotate the qubit states on the Bloch sphere to enact quantum gates. An example of a superconducting qubit chip is shown in figure 2-1.

S_X gates are performed on superconducting qubits with microwave pulses, and R_Z gates with arbitrary angles are implemented by adjusting the phase on future

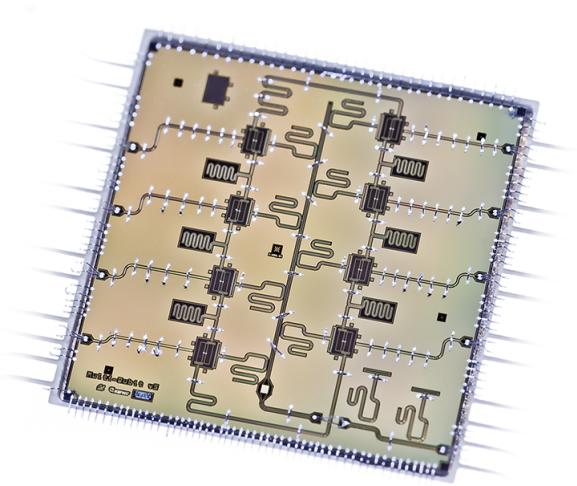

Figure 2-1: *A chip containing superconducting qubits and microwave buses [32].*

microwave pulses. This is described as performing a "virtual Z gate" [22, 30]. There are several methods of performing two-qubit gates on superconducting hardware. There are various approaches to implementing CZ gates, which can be converted into $CNOT$ gates with single qubit operations [9, 33, 54]. Another popular approach is the $iSWAP$ gate, which performs the unitary operation shown in equation (2.3), and has several physical implementations [21, 42]. For any multi-qubit gate acting on superconducting qubits, a physical connection, such as a microwave bus, must exist between qubits to facilitate interaction. If two qubits are required to interact for the purposes of a quantum circuit, but no physical connection exists, then qubit states must be moved around using $SWAP$ gates, which can be implemented with 3 $CNOT$ gates, or 1 $iSWAP$ gate and phase correction. The arrangement of qubits and connections between qubits depends on the superconducting chip architecture.

$$iSWAP = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & i & 0 \\ 0 & i & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{2.3}$$

A different approach is to used the spin states of trapped ions as the qubit. One

spin state becomes $|0\rangle$ and the orthogonal state becomes $|1\rangle$. Gates are applied to trapped ions in the form of laser pulses. There are several approaches to performing gates that can be transformed into the $CNOT$ through single-qubit operations. Alternative approaches entangle many qubits simultaneously [10, 51, 55]. The ability to entangle multiple qubits at the same time is an advantage over superconducting qubits. Other approaches instead aim to prepare specific entangled states which can then be used for operations on qubits [2, 52].

While superconducting qubit and trapped ion technology are currently the most advanced in terms of scalability and low gate error rates, there are other physical systems being studied for their potential use in quantum computers. Optical quantum computing benefits from high speed and room temperature operation but is limited by short photon lifetime [31, 36, 48]. Neutral atom quantum computing has potential in scalability but suffers from lower gate fidelities compared to trapped ion technology [11].

2.3 Noisy Intermediate-Scale Quantum Computing

The short-term strategy for making use of quantum computing is called NISQ (Noisy Intermediate-Scale Quantum). NISQ computing involves running quantum gates directly on hardware, and relying on the hardware to have a sufficiently high fidelity to execute the desired quantum circuit with a reasonable error rate. Most quantum hardware implementations can implement any single-qubit gate by performing $R_Z(\theta)$ rotations with an arbitrary angle θ, combined with S_X, gates. All quantum hardware implementations must also implement at least one multi-qubit gate. The $CNOT$ is a popular choice because it is convenient for theorists to work with but also practical to implement in hardware. Other popular choices include $iSWAP$, which has advantages over the $CNOT$ in the ease of implementation with superconducting qubits [21, 42], and the Mølmer-Sørenson gate, which is commonly used with trapped-ion qubits [51]. Multi-qubit gates require hardware designed to facilitate qubit-qubit interactions while minimizing unintentional interactions, making

two-qubit gates noisier than one-qubit gates. Additionally, some types of qubits require physical connections for each pair of qubits that will be able to interact, such that hardware topology becomes a factor to consider in software quantum circuit design. For these reasons, it is a priority to minimize the number of multi-qubit gates in a quantum circuit when targeting NISQ computing. A common gateset for NISQ computing is S_X, $R_Z(\theta)$, and $CNOT$. The unitaries for these gates are shown in equation (2.4). This is a continuous gateset because the θ in $R_Z(\theta)$ can be any real number (although in practice there are hardware limitations on the precision).

$$CNOT = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \tag{2.4}$$

$$S_X = \begin{pmatrix} \frac{1}{2}+\frac{1}{2}i & \frac{1}{2}-\frac{1}{2}i \\ \frac{1}{2}-\frac{1}{2}i & \frac{1}{2}+\frac{1}{2}i \end{pmatrix} \; R_Z(\theta) = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\theta} \end{pmatrix}$$

2.4 Quantum Error Correction

Because ideal two-level quantum systems do not physically exist, hardware quantum computers are inherently noisy. This noise can build up over the course of a long quantum circuit, ultimately resulting in qubits that give random results when measured, rather than the output of an algorithm. The long-term strategy for achieving large-scale quantum computing is to use quantum error correction, which allows a quantity-for-quality tradeoff in which a large but noisy quantum computer can simulate a smaller but less noisy quantum computer.

To mitigate the buildup of errors, quantum error correction is a technique that can protect the quantum state of a qubit, separate it from potential errors, and then correct those errors. This works by using multiple physical qubits acting as a single logical qubit. One joint state of the system is assigned to be the logical $|0\rangle$ state of the system, and another is assigned to be logical $|1\rangle$, often denoted as $|0_L\rangle$ and $|1_L\rangle$. As long as the superposition of the overall logical qubit system is a linear

combination of those two states, the system is acting as an ideal individual qubit. If the superposition contains other states, this is an error to be corrected. By making stabilizer measurements, the superposition will collapse to a state where the error is known and a correction can be applied, while leaving the protected qubit state unmodified.

The simplest error-correcting codes are only capable of correcting errors with individual hardware qubits, so that two hardware qubits affected by an error at the same time could result in an incorrect state for the overall logical qubit. Larger error-correcting codes (such as simply applying the same error-correcting code recursively, which is known as "concatenating" error correcting codes) can allow higher error rates to be corrected.

2.5 Shor's 9-Qubit Error Correction Code

In this section, we introduce the Shor code to explain quantum error correction [37,49]. The $|0_L\rangle$ and $|1_L\rangle$ states for this code are shown in

$$\begin{aligned} |0_L\rangle &\equiv \frac{(|000\rangle + |111\rangle)(|000\rangle + |111\rangle)(|000\rangle + |111\rangle)}{2\sqrt{2}} \\ |1_L\rangle &\equiv \frac{(|000\rangle - |111\rangle)(|000\rangle - |111\rangle)(|000\rangle - |111\rangle)}{2\sqrt{2}} \end{aligned} \tag{2.5}$$

This code is robust against both a bit-flip error ($|0\rangle \mapsto |1\rangle$) and a phase flip error ($|+\rangle \mapsto |-\rangle$) to any of the 9 qubits. Let us start with our logical qubit in the state $\alpha\,|0_L\rangle + \beta\,|1_L\rangle$. Suppose that the first qubit suffers an error in the form of an arbitrary R_X rotation such that $|0\rangle \mapsto a\,|0\rangle + b\,|1\rangle$ and $|1\rangle \mapsto b\,|0\rangle + a\,|1\rangle$ where $a, b \in \mathbb{R}_{\geq 0}$ and $a^2 + b^2 = 1$. The new state of our logical qubit system is shown in equation (2.6).

$$\begin{gathered} \alpha\frac{(a\,|000\rangle + a\,|111\rangle + b\,|100\rangle + b\,|011\rangle)(|000\rangle + |111\rangle)(|000\rangle + |111\rangle)}{2\sqrt{2}} \\ + \\ b\frac{(a\,|000\rangle - a\,|111\rangle + b\,|100\rangle - b\,|011\rangle)(|000\rangle - |111\rangle)(|000\rangle - |111\rangle)}{2\sqrt{2}} \end{gathered} \tag{2.6}$$

We will detect this error by performing the measurement corresponding to the set of operators, known as syndromes, shown in equation (2.7) [37]. By measuring the first 3 qubits in this way, we will arrive at one of the states shown in equation (2.8) and will know which state by using the outcome of the measurement. This measurement, known as a stabilizer measurement, is performed by interacting the qubits in our error correcting code with an ancilla qubit in $|+\rangle$ state and measuring it. Two measurements using this method will allow us to arrive at one of the four syndrome results. Note that P_2 and P_3 correspond to errors on the second and third qubit, which we have assumed are error-free so far, so the probability of measuring these values is 0.

$$\begin{aligned} P_0 &\equiv |000\rangle\langle 000| + |111\rangle\langle 111| \\ P_1 &\equiv |100\rangle\langle 100| + |011\rangle\langle 011| \\ P_2 &\equiv |010\rangle\langle 010| + |101\rangle\langle 101| \\ P_3 &\equiv |001\rangle\langle 001| + |110\rangle\langle 110| \end{aligned} \tag{2.7}$$

$$\begin{gathered} \text{Measurement Result } P_0 \text{ with } p = a^2 \\ \alpha\frac{(|000\rangle + |111\rangle)(|000\rangle + |111\rangle)(|000\rangle + |111\rangle)}{2\sqrt{2}} \\ + \\ \beta\frac{(|000\rangle - |111\rangle)(|000\rangle - |111\rangle)(|000\rangle - |111\rangle)}{2\sqrt{2}} \end{gathered} \tag{2.8}$$

$$\begin{gathered} \text{Measurement Result } P_1 \text{ with } p = b^2 \\ \alpha\frac{(|100\rangle + |011\rangle)(|000\rangle + |111\rangle)(|000\rangle + |111\rangle)}{2\sqrt{2}} \\ + \\ \beta\frac{(|100\rangle - |011\rangle)(|000\rangle - |111\rangle)(|000\rangle - |111\rangle)}{2\sqrt{2}} \end{gathered}$$

If the measurement outcome corresponds to P_0, we have already corrected the error by projecting the state into the logical standard basis through stabilizer measurement,

and no further action is needed. If the measurement outcome corresponds to P_1, we have detected that a bit flip error has occurred on the first qubit, and the state can be put back into the logical standard basis by simply applying an X gate to the first qubit.

Now consider the case where the error takes the form of an arbitrary R_Z rotation such that $|0\rangle \mapsto |0\rangle$ and $|1\rangle \mapsto e^{i\theta}|1\rangle$. The state $\alpha|0_L\rangle + \beta|1_L\rangle$ will become the state shown in equation (2.9).

$$\begin{gathered} \alpha\frac{(|000\rangle + e^{i\theta}|111\rangle)(|000\rangle + |111\rangle)(|000\rangle + |111\rangle)}{2\sqrt{2}} \\ + \\ \beta\frac{(|000\rangle - e^{i\theta}|111\rangle)(|000\rangle - |111\rangle)(|000\rangle - |111\rangle)}{2\sqrt{2}} \end{gathered} \tag{2.9}$$

The path to correcting this error becomes more clear if we define the states $|+_3\rangle = \frac{1}{\sqrt{2}}(|000\rangle + |111\rangle)$ and $|-_3\rangle = \frac{1}{\sqrt{2}}(|000\rangle - |111\rangle)$. States in the form $\frac{1}{\sqrt{2}}(|000\rangle \pm e^{i\theta}|111\rangle)$ expand to $\frac{1}{2}(|+_3\rangle + |-_3\rangle \pm e^{i\theta}|+_3\rangle \mp e^{i\theta}|-_3\rangle)$. Rewriting equation (2.9) using these definitions, we arrive at equation (2.10).

$$\begin{gathered} \alpha\frac{(1+e^{i\theta})|+_3\rangle + (1-e^{i\theta})|-_3\rangle}{2}|+_3\rangle|+_3\rangle \\ + \\ \beta\frac{(1+e^{i\theta})|-_3\rangle + (1-e^{i\theta})|+_3\rangle}{2}|-_3\rangle|-_3\rangle \end{gathered} \tag{2.10}$$

These states now resemble equation (2.6), and the path to error correction is clear. The syndromes we must measure are shown in equation (2.11).

$$\begin{aligned} P_0 &\equiv |+_3\,+_3\,+_3\rangle\langle+_3\,+_3\,+_3| + |-_3\,-_3\,-_3\rangle\langle-_3\,-_3\,-_3| \\ P_1 &\equiv |-_3\,+_3\,+_3\rangle\langle-_3\,+_3\,+_3| + |+_3\,-_3\,-_3\rangle\langle+_3\,-_3\,-_3| \\ P_2 &\equiv |+_3\,-_3\,+_3\rangle\langle+_3\,-_3\,+_3| + |-_3\,+_3\,-_3\rangle\langle-_3\,+_3\,-_3| \\ P_3 &\equiv |+_3\,+_3\,-_3\rangle\langle+_3\,+_3\,-_3| + |-_3\,-_3\,+_3\rangle\langle-_3\,-_3\,+_3| \end{aligned} \tag{2.11}$$

This yields the measurement results shown in equation (2.12). Again, syndromes P_2 and P_3 correspond to errors on qubits other than qubit 1.

$$\begin{aligned}
&\text{Measurement Result } P_0 \text{ with } p = \frac{|1+e^{i\theta}|^2}{4} \\
&\alpha\,|+_3\,+_3\,+_3\rangle + \beta\,|-_3\,-_3\,-_3\rangle \\
&\text{Measurement Result } P_1 \text{ with } p = \frac{|1-e^{i\theta}|^2}{4} \\
&\alpha\,|-_3\,+_3\,+_3\rangle + \beta\,|+_3\,-_3\,-_3\rangle
\end{aligned} \tag{2.12}$$

If the measurement outcome corresponds to P_0, the error has already been corrected. If the measurement outcome corresponds to P_1, applying Z gates to all of the first three qubits should be sufficient to switch $|+_3\rangle$ to $|-_3\rangle$ and visa-versa, thus correcting the phase flip error.

Because of the symmetry of the problem, errors in these forms affecting any individual qubit can be corrected. Because any single qubit quantum gate can be described as a sequence of R_X and R_Z gates, any arbitrary error affecting a single qubit can be corrected by applying a combination of these two error correction protocols. This error correcting code provides fault tolerance as long as the probability of errors affecting multiple qubits simultaneously is low [37, 49].

2.6 Transversal Gates on Error Corrected Qubits

A quantum logic gate applied to a logical qubit must perform the operation we would expect on the $|0\rangle$ and $|1\rangle$ states of an ideal hardware qubit to the $|0_L\rangle$ and $|1_L\rangle$ states of the larger logical qubit system. To perform these logical gates fault-tolerantly, we must ensure that these gates cannot introduce errors that could not be corrected. More specifically, the circuit that performs a logical quantum gate must not involve gates that interact more than one hardware qubit within the same logical qubit as noisy multi-qubit gates could introduce simultaneous errors which cannot be corrected. Performing a two-qubit gate between hardware qubits of separate logical qubits is acceptable, because only one error per logical qubit may be introduced. Gates performed this way are known as "transversal" gates, and the set of gates that can be performed transversally is the set of Clifford gates [37], which consists of the

gates $CNOT$, H, S, and all gates that can be composed of those three (including the Pauli gates X, Y,Z, as well as the S_X and $iSWAP$ gates mentioned previously). However, circuits consisting of just Clifford gates can be simulated in $O(n \log n)$ time on a classical computer, and therefore at least one non-Clifford gate is needed to achieve a quantum advantage [37]. The unitaries for $CNOT$ and S_X are shown in equation (2.4), and the unitaries for X, Y, Z, H, and S gates are shown in equation (2.13).

$$X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} Y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix} Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$$
$$H = \begin{pmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{pmatrix} S = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\frac{\pi}{2}} \end{pmatrix} \tag{2.13}$$

2.7 Fault-Tolerant Quantum Computing

To achieve fault-tolerant quantum computing, quantum error-correcting codes and transversal gates can be used to create logical qubits and perform Clifford gates. For universal quantum computing, we need at least one non-Clifford gate to be able to approximate any unitary. In NISQ quantum computing, the R_Z gate often fills this role. However, to be suitable for fault-tolerant quantum computing, there must be a way to perform this gate on a logical qubit without introducing error to multiple physical qubits at a time. And because this gate is not a Clifford gate, it will not be possible to perform transversally.

Two common choices for a non-Clifford gate for fault-tolerant quantum computing are the Toffoli gate and the T gate. The Toffoli gate is a three-qubit gate that can be described as the "Controlled-Controlled-Not", or a quantum "And" gate. The T gate is a one-qubit gate that is a Z rotation of $\frac{\pi}{4}$. This makes it the square root of the S gate. The Toffoli gate is important because it forms a universal gateset all by itself. Therefore, any gateset that can implement a Toffoli is universal. Seven T gates, combined with Clifford gates, can implement a Toffoli, making the Clifford+T gateset universal. We will focus on the T gate because it is simpler to work with for

our purposes as it is a one-qubit gate. The unitaries for the Toffoli and T gates are shown in equation (2.14).

$$\text{Toffoli} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix} T = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\frac{\pi}{4}} \end{pmatrix} \tag{2.14}$$

While a T gate cannot be performed transversally, there are several approaches to perform a T gate on an encoded logical qubit fault-tolerantly. One option is to use quantum gate teleportation. By preparing an ancilla qubit in the superposition state $|+\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$, and interacting it with the target qubit, it is possible to move the state of the target qubit, replacing the state on the ancilla qubit. This process is called quantum *state* teleportation. If a gate is performed on the ancilla qubit before the teleportation procedure, then the teleported state will have a gate applied to it. The applied gate may not be the same as the gate applied to the ancilla, but for gates that commute with the control side of the $CNOT$, including the T gate, it will be the same. This procedure is known as quantum *gate* teleportation and can be performed fault-tolerantly. This reduces the problem of performing the T gate on an arbitrary encoded state, to the problem of performing the T gate on a specific state: $\frac{1}{\sqrt{2}}(|0_L\rangle + |1_L\rangle)$. Or equivalently, to prepare the state that results from performing T on this superposition state, which is $\frac{1}{\sqrt{2}}(|0_L\rangle + e^{i\frac{\pi}{4}}|1_L\rangle)$. The circuit for applying a T gate by using quantum gate teleportation is depicted in figure 2-2. Unfortunately, it is not possible to fault-tolerantly generate this state with only Clifford gates. However, this limitation can be bypassed with magic state distillation [5, 28]. Magic state distillation consumes multiple erroneous copies of the target state (which can be prepared non-transversally) to produce a smaller number

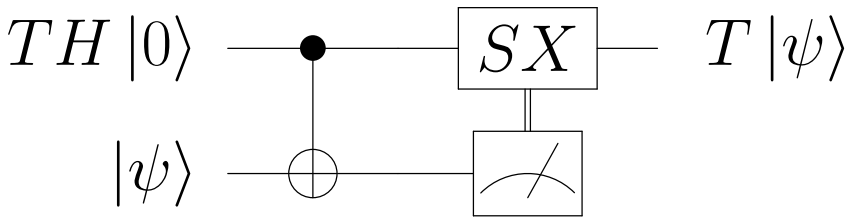

Figure 2-2: *The circuit for applying a T gate via quantum gate teleportation. When used for fault-tolerant quantum computing, the qubits in this circuit represent logical qubits that consist of many physical qubits. The state $TH\,|0\rangle$ must be prepared by a fault-tolerant method, such as magic state distillation [5, 28].*

of higher-fidelity copies of the target state. Another option is to find a code that has a tranversal non-Clifford gate through "gauge fixing". The 3D gauge color code is the representative example of gauge fixing [4]. From the Bravyi-Koenig bound [6], this requires qubits to be arranged in a dimension equal or larger than three.

We now have a gateset for universal fault-tolerant quantum computing. This is the set of Clifford gates, combined with the T gate. A $T^\dagger$ gate can be performed by applying T gate followed by $S^\dagger$, so for practical purposes it is often included as well. While Clifford gates can be performed transversally, the process for performing T gates involves multiple ancilla, a quantum circuit with many Clifford gates, measurement, and feed-forward to apply a gate dependent on the result of a measurement in the same quantum circuit. The cost of performing a T gate is often orders of magnitude greater than that of performing a Clifford gate, so it is a priority to minimize the number of T gates in a quantum circuit for fault-tolerant quantum computing. This is a discrete gateset, meaning that there is a finite number of basic gates.

2.8 Quantum Gate Synthesis

Quantum gate synthesis is the process of generating a quantum circuit to implement a desired unitary matrix. The field began with the Solovay-Kitaev theorem, which gave constructive proof that any unitary matrix could be approximated by a quantum circuit to within an error threshold ϵ with only $O\left(\log^{3.97}\frac{1}{\epsilon}\right)$ quantum gates [14]. Implementations of the Solovay-Kitaev algorithm exist, but produce long and repetitive

circuits, even for 1-qubit unitaries. The field has progressed since then, with a variety of different approaches depending on the type of unitary and the desired quality and efficiency metrics for the synthesized quantum circuit.

2.9 NISQ Quantum Gate Synthesis

For synthesis approaches targeting NISQ circuits, the target gateset is usually S_X, $R_Z(\theta)$, and $CNOT$, with the primary goal of minimizing the number of $CNOT$s. However, NISQ circuits have the advantage of a continuous gateset, which allows for approaches that use numerical optimization [12,19,25,34,35,40,50]. These approaches are inherently approximate; relying on a threshold and a matrix distance function to signal approximate matrix equivalence. Measures such as the diamond norm [26] and the trace distance [17, 37] have been used, but a popular measure for modern synthesis techniques is a cost function based on the Hilbert-Schmidt norm [12, 16, 25, 34,35,40,50,58], such as the function shown in equation (2.15). The general structure of synthesis by numerical optimization is to choose an ansatz circuit and define an optimization problem such that the matrix distance to the target unitary is to be minimized by adjusting continuous paramters of the ansatz circuit. The primary difference between numerical optimization approaches is the design of ansatz circuits.

$$D(U_1, U_2) = \sqrt{1 - \frac{|\mathrm{Tr}(U_1^\dagger U_2)|}{2^N}} \tag{2.15}$$

Some types of quantum computing hardware, such as Rydberg atoms [11] and superconducting qubits [22, 30] have a restriction on which pairs of qubits can be involved in a two-qubit gate. This limitation is a factor of NISQ-targeted synthesis techniques [12, 25, 34, 40, 50, 58] as well as other layers of quantum software, such as error correcting code design [57].

Because the size of a unitary scales exponentially with the number of qubits in the corresponding circuit, quantum gate synthesis does not scale to large circuits. It can be useful for discovering a circuit for small unitaries, but it is also of interest as

a circuit optimization step. Most circuit optimization techniques involve applying a fixed set of rules to cancel out gates, but resynthesis can discover optimization options that would be difficult to find using traditional rule-based approaches [1,35,38,41,56, 58].

Numerical optimization is not the only technique for NISQ gate synthesis. The KAK decomposition [24, 53] can be used to compute optimal implementations of any 2-qubit unitary with $CNOT$ and R_Z gates. There are quantum gate synthesis algorithms based on recursively applying matrix decomposition techniques, such as the Cosine-Sine Decomposition (CSD) or Quantum Shannon Decomposition (QSD), and finishing with the KAK decomposition [15,47]. Generally the techniques based on decomposition have faster runtime, but produce circuits that are much longer when compared to numerical optimization techniques.

2.10 Fault-Tolerant Quantum Gate Synthesis

Synthesis algorithms targeting fault-tolerant quantum circuits often use different approaches from those targeting NISQ circuits. Because the Clifford+T gateset does not contain any continuous gates, numerical optimization cannot be applied. Furthermore, not every gate can be exactly implemented using Clifford+T gates. In cases where an exact implementation does not exist, instead an approximation must be used. This separates the problem of synthesizing general unitaries as Clifford+T circuits into two steps: finding a sufficient approximate unitary that can be implemented exactly as a Clifford+T circuit, and finding the implementation of that unitary.

Unitaries in the ring $\mathbb{Z}[i, \frac{1}{\sqrt{2}}]$ can be exactly implemented with the Clifford+T gateset [18,27], and there are algorithms that can exactly synthesize a unitary in this ring [1, 3, 18, 27, 39, 44]. Certain well-known unitaries, such as the Toffoli, have been thoroughly studied and optimized [1, 44].

Any unitary can be approximated to arbitrary precision with Clifford+T quantum circuts [14,20], but doing so efficiently for multi-qubit unitaries is an open problem. There are several approaches to approximating arbitrary unitaries with the

Clifford+T gateset, the most famous of which being the Solovay-Kitaev method [14] which can be used to approximate any unitary, but in practice produces infeasibly long circuits even for the single-qubit case. Actual implementations of approximate Clifford+T synthesis algorithms usually focus on approximating $R_Z(\theta)$ for various values of θ [3, 44]. The process of approximating multi-qubit unitaries with Clifford+T circuits usually involves approximating the circuit with the Clifford+R_Z gateset and then approximating R_Z gates using a Clifford+T synthesis method [7].

Chapter 3

Optimization Algorithm

3.1 Converting Clifford+R_Z Circuits to Clifford+T

A conventional approach to converting a numerically synthesized circuit in the Clifford+R_Z format to a Clifford+T circuit is to use a synthesis algorithm to convert each R_Z gate to a Clifford+T circuit [7]. During numerical synthesis, the values of parameters are generated with no consideration of Clifford+T cost, and often settle on values which result in high T-count circuits after this conversion process.

We improve this approach with a circuit post-processing procedure that consists of the sequential application of a two-pass multistart numerical optimization algorithm. This process minimizes the number of R_Z gates in the circuit to ensure that the Clifford+T count is minimized when converting from a Clifford+R_Z circuit to a Clifford+T circuit.

3.2 Multi-Objective Optimization

A standard quantum synthesis by numerical optimization problem can be defined as minimizing a matrix distance function (such as the one shown in equation (2.15)) between an implemented and a target unitary with respect to the parameters of the circuit producing the implemented unitary. This minimization problem is shown in equation (3.1). When $D(U(\vec{x}), U_{\text{target}}) = 0$, the two unitaries are identical, indicating

that a circuit that exactly implements the desired unitary has been found. In practice, usually an approximation is accepted when it is within some threshold, such that $D(U(\vec{x}), U_{\text{target}}) < \epsilon$.

$$\min_{\vec{x}} \quad D(U(\vec{x}), U_{\text{target}}) \tag{3.1}$$

The goal for a T-count reducing numerical optimization problem is to minimize the T-count that the circuit will have after conversion while keeping the matrix distance within a threshold. This constrained minimization problem is shown in equation (3.2).

$$\begin{aligned} \min_{\vec{x}} \quad & T\text{-Count}(\vec{x}) \\ \text{s.t.} \quad & D(U(\vec{x}), U_{\text{target}}) < \epsilon \end{aligned} \tag{3.2}$$

The function that maps a set of R_Z angles to a T-Count is discrete-valued, while numerical optimizers usually require continuous and preferably differentiable functions. $R_Z(\theta)$ gates where θ is a multiple of $\frac{\pi}{2}$ are Clifford gates and therefore require 0 T gates in Clifford+T form. When θ is a multiple of $\frac{\pi}{4}$ but not a Clifford gate, then it can be peformed with Clifford gates and only 1 T gate. With any other value of θ, the R_Z gate must be approximated with a Clifford+T circuit with a T-count which scales with the error threshold of the approximation [18, 27]. The T-count of these approximated circuits is largely independent of the value of θ. This situation is expressed in equation (3.3).

$$\begin{aligned} T\text{-Count}(\vec{x}) &= \sum_{i=0}^{l} T\text{-Count}(x_i) \\ T\text{-Count}(\theta) &= \begin{cases} 0 & \text{if } \exists_i \in \mathbb{Z} | \theta = i\frac{\pi}{2} \\ 1 & \text{if } \exists_i \in \mathbb{Z} | \theta = i\frac{\pi}{2} + \frac{\pi}{4} \\ O(\log(1/\epsilon)) & \text{otherwise} \end{cases} \end{aligned} \tag{3.3}$$

However, we can guide a numerical optimizer towards solutions to this problem by solving a relaxed version of the problem in which the true T-count function is replaced by a periodic function which is easily optimized and has local minima at

the values of θ where the T-count will be either 0 or 1. The triangle wave function $\left|\left(\theta - \frac{\pi}{8} \mod \frac{\pi}{4}\right) - \frac{\pi}{8}\right| \frac{8}{\pi}$ is the simplest choice. It ranges from 0 to 1 and has local minima at multiples of $\frac{\pi}{4}$. Our relaxed optimization problem is shown in equation (3.4).

$$\begin{aligned} \min_{\vec{x}} \quad & \text{Relaxed } T\text{-Count}(\vec{x}) \\ \text{s.t.} \quad & D(U(\vec{x}), U_{\text{target}}) < \epsilon \\ \text{Relaxed } T\text{-Count}(\vec{x}) = & \sum_{i=0}^{l} \left|\left(x_i - \frac{\pi}{8} \mod \frac{\pi}{4}\right) - \frac{\pi}{8}\right| \frac{8}{\pi} \end{aligned} \tag{3.4}$$

A problem of this form can be given to off-the-shelf numerical optimizers that are designed to minimize nonlinear objective functions with nonlinear constraints, where both the objective and constraints are continuous. Our constraint is differentiable, but our objective is not differentiable at minima and maxima. An alternative formulation of the Relaxed T-Count function that is differentiable everywhere is explored in section 4.4. Solutions to this relaxed problem can be rounded to solutions of the original problem, as long as we ensure that constraints are not violated after rounding.

3.3 Multistart Two-Pass Optimization

Numerical optimizers such as COBYLA and SLSQP [29, 43] can be used to solve the relaxed constrained optimization problem from equation (3.4). However, these optimizers require a feasible initial guess [29], and the constraint that $D(U(\vec{x}), U_{\text{target}}) < \epsilon$ is only satisfied in small regions of the search space. To find these points, we must first solve the standard synthesis by numerical optimization problem from equation (3.1). As that problem is constraint-free, all points are feasible, such that random initial points can be used. In most cases, there will be several solutions to this problem, and the regime around each solution may result in a different T-count. Additionally, the numerical optimizer may get stuck in a local minima rather than find a solution that satisfies $D(U(\vec{x}), U_{\text{target}}) < \epsilon$. The difficulties of the numerical optimization landscape are depicted in figure 3-1.

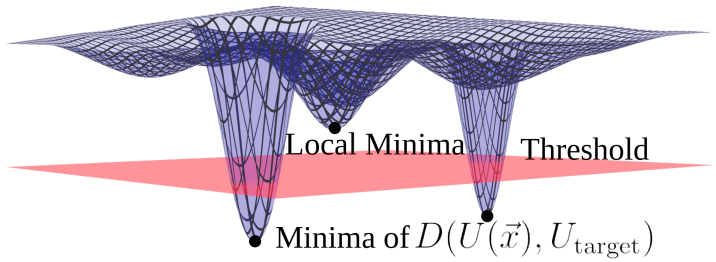

Figure 3-1: *A depiction of an optimization landscape demonstrating the difficulties of the numerical optimization problem for quantum gate synthesis. There are multiple minima below the threshold, as well as local minima above the threshold. Note that the real landscape of a quantum gate synthesis numerical optimization problem has many dimensions. Multiple local or global minima cannot be viewed simply by sweeping over one or two parameters because the complexity of the landscape is in the relationship between many different parameters.*

To ensure that we arrive at the lowest Relaxed T-Count solution, we must run the initial optimization several times, filter the solutions to remove duplicates and infeasible starting points, and then run the constrained optimization using the remaining solutions to the initial problem as starting points. Finally, we must filter through the results from the constrained optimization problem to find the solution with the smallest value of the objective function. This solution will be the overall solution to the relaxed constrained optimization problem from equation (3.4) and an approximation of an overall solution to equation (3.2). This procedure is described as pseudocode in algorithm 1.

3.4 Sequential Optimization

The solutions to equation (3.4) are not quite solutions to equation (3.2). To convert them, we must round them while ensuring that the $D(U(\vec{x}), U_{\text{target}}) < \epsilon$ constraint is held. This can be accomplished by finding the parameter x_i with the smallest value of Relaxed T-Count(x_i), and rounding it to the nearest multiple of $\frac{\pi}{4}$, and then re-optimizing the remaining parameters to ensure that rounding did not make it impossible to achieve the error threshold. This process is repeated until all parameters

have been rounded successfully, or until rounding the lowest-cost parameter made it impossible to find a solution to the optimization problem which satisfies the constraint, in which case the last successful solution is returned. Additionally, so far we have focused on guiding parameters to the nearest multiple of $\frac{\pi}{4}$, which could be a solution of T-count 0 or 1. It is already a significant improvement to reduce the instances where R_Z gates must be performed by approximate circuits which have T-counts on the order of 100 gates (for error thresholds of $\epsilon = 10^{-10}$). Ideally, we would like to minimize the number of gates that are performed with 1 T-gate as well. To do this, we can run the whole optimization procedure with a Relaxed T-Count($\vec{x}$) and rounding procedure that are tuned to multiples of $\frac{\pi}{2}$ first, to round as many parameters as possible to 0 T-count Clifford gates, and then repeat the process while targeting multiples of $\frac{\pi}{4}$ to round as many of the remaining parameters to 1 T-count gates. Any further remaining R_Z gates must be approximated, and this can be done through established techniques [18, 27]. This sequential optimization procedure is described as pseudocode in algorithm 2. Note that because numerical optimization runtime scales with the number of parameters being optimized, each subsequent optimization application will complete more quickly than the previous iteration.

Algorithm 1 Multistart Two-Pass Optimization

function MINIMIZE(f, $\vec{x}_0$)
    return $\arg\min_{\vec{x}} f(\vec{x})$
function MINIMIZE S.T. CONSTRAINTS(f, g, $\vec{x}_0$)
    return $\arg\min_{\vec{x}} f(\vec{x})$ s.t. $g(\vec{x}) < \epsilon$
function MULTISTART TWO-PASS OPTIMIZATION($U(\vec{x})$, U_{target})
    1st pass results $\leftarrow \{\}$
    for all $i \in$ number of starting points **do**
        $\vec{x}_i \leftarrow$ MINIMIZE($D(U(\vec{x}), U_{\text{target}})$, randomly generated starting point)
        if $D(U(\vec{x}_i), U_{\text{target}}) < \epsilon$ and $\vec{x}_i \notin$ 1st pass results **then**
            1st pass results $\leftarrow$ 1st pass results $\cup \{\vec{x}_i\}$
    2nd pass results $\leftarrow \{\}$
    for all $\vec{x}_i \in$ 1st pass results **do**
        $\vec{x}_j \leftarrow$ MINIMIZE S.T. CONSTRAINTS(Relaxed T-Count($\vec{x}$), $D(U(\vec{x}), U_{\text{target}})$, $\vec{x}_i$)
        if $D(U(\vec{x}_j), U_{\text{target}}) < \epsilon$ **then**
            2nd pass results $\leftarrow$ 2nd pass results $\cup \{\vec{x}_j\}$
    return $\arg\min$ Relaxed T-Count($\vec{x}_j$)$\forall \vec{x}_j \in$ 2nd pass results

Algorithm 2 Sequential Optimization

function RELAXED T-COUNT(x)
    return $\sin^2(\omega x)$
function SEQUENTIAL OPTIMIZATION($U(\vec{x})$, U_{target})
    for all $\omega \in 2, 4$ **do**
        $\vec{x}_{\text{opt}} \leftarrow$ MULTISTART TWO-PASS OPTIMIZATION($U(\vec{x})$, U_{target})
        while $D(U(\vec{x}_{\text{opt}}), U_{\text{target}} < \epsilon$ **do**
            $\vec{x}_{\text{best}} \leftarrow \vec{x}_{\text{opt}}$
            $i \leftarrow \arg\min$ RELAXED T-COUNT(x_i) $\forall x_i \in \vec{x}_{\text{opt}}$
            $U(\vec{x}) \leftarrow U(\vec{x}|x_i = \lfloor x_i / \frac{\pi}{\omega} \rceil \frac{\pi}{\omega})$
            $\vec{x}_{\text{opt}} \leftarrow$ MULTISTART TWO-PASS OPTIMIZATION($U(\vec{x})$, U_{target})
        $\vec{x}_{\text{opt}} \leftarrow \vec{x}_{\text{best}}$
    return $\vec{x}_{\text{best}}$

Chapter 4

Numerical Results

4.1 Software Implementation

We developed a software implementation of the algorithm presented in chapter 3 written in `Python 3` using `numpy` and the implementations of the numerical optimizers `BFGS`, `COBYLA`, `SLSQP`, and `Levenberg-Marquardt` in `scipy`. We also leveraged the library for numerical synthesis of quantum gates `qsearch` [12,13,50]. After performing our circuit-optimizing routine, we synthesize any remaining R_Z gates to Clifford+T circuits with the program `gridsynth` [44, 46]. We implement several variations of the Relaxed T-Count($\vec{x}$) function in addition to the triangle wave described in equation (3.4): we implement a sinusoidal wave, and logarithmic and exponential modifications to both the sinusoidal and triangle waves. The performance of various Relaxed T-Count functions is explored in section 4.4.

4.2 Exact Synthesis

For several well-known unitaries which can be performed exactly with Clifford+T circuits, our algorithm was able to derive circuits that match the best known T-count. When achieving exact synthesis, every R_Z parameter in the circuit becomes rounded to a 0 or 1 T-count gate during the sequential optimization process. The results from exact synthesis of several well-known unitaries is shown in table 4.1. For

Gate	Qubits	R_Z Pre-Opt	R_Z Post-Opt	T-Count	Time (s)
QFT	2	21	0	3	6
Toffoli	3	49	0	7	79
Fredkin	3	49	0	7	53
Or	3	49	0	7	61

Table 4.1: *Summary of the results from exact synthesis. These gates were synthesized with the program* `qsearch` *[12, 13, 50], which relies on numerical optimization. The circuits were then processed with sequential two-pass multistart optimization, which removed all R_Z gates, resulting in an exactly synthesized Clifford+T circuit. The number of R_Z gates in the circuit before and after optimization is reported, as well as the total run time (including both synthesis and optimization time).*

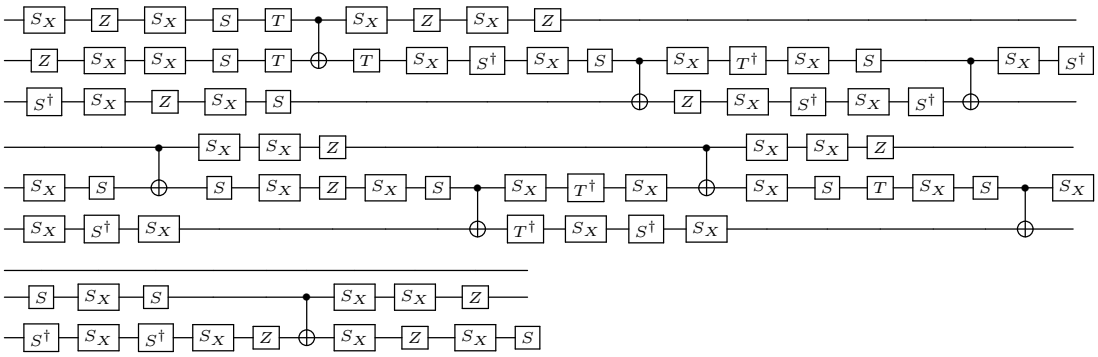

Figure 4-1: *Our synthesized Toffoli circuit. Out algorithm used more Clifford gates than necessary, but used only 7 T gates, which matches the best results from previous work. Future work may include an effort to reduce the number of Clifford gates as well.*

these unitaries, we match the best-known results from other techniques [1,16,38]. We saw faster performance on 3-qubit unitaries than the algorithm presented in [1], which takes 497 seconds to synthesize 3-qubit unitaries, as well as the algorithm from [16], which took 5.7 hours to synthesize 3-qubit unitaries with 4 T gates, compared to our algorithm which consistently synthesizes 3-qubit unitaries in under 2 minutes. Our synthesized Toffoli circuit is presented in figure 4-1. Our algorithm used more Clifford gates than necessary, but used only 7 T-gates, which matches the best results from previous work. Future work may include an effort to reduce the number of Clifford gates as well.

This approach relies on the same threshold-based techniques that we use for approximate synthesis. However, only a comparatively small number of unitaries exist that are implemented exactly with a small number (e.g. 7) of T gates. As it would

Gate	Qubits	R_Z Pre-Opt	R_Z Post-Opt	T-Count	$D(U(\vec{\theta_f}), U)$	Time (s)
QFT	3	49	3	180	5.60×10^{-11}	103
QFT	4	77	7	356	8.92×10^{-11}	1579

Table 4.2: *Summary of the results from approximate synthesis. Two sizes of QFT were synthesized with the synthesis program* `qsearch` *[12,13,50], which relies on numerical optimization. The circuits were then processed with sequential two-pass multistart optimization to reduce the number of R_Z gates requiring optimization. The number of R_Z gates in the circuit before and after optimization is reported, as well as the final T-count, matrix distance, and total runtime (including both synthesis and optimization time).*

generally take $O(log\frac{1}{\epsilon})$ T gates to implement another unitary within ϵ of a small T-count unitary [20], we can conclude that any small T-count unitary within ϵ of the target unitary must be exactly the target unitary. This analysis allows us to conclude that the synthesis is exact whenever the final number of T gates is significantly less than the expected lower bound on the number of T gates to approximately implement a unitary within ϵ of the target. In practice, circuits that are fully reduced to Clifford+T gates by our optimization algorithm without leaving any residual R_Z gates for approximation can often be considered exact.

4.3 Approximate Synthesis

Most unitaries cannot be exactly implemented with Clifford+T circuits [20, 37]. In these cases, the sequential optimization procedure will remove many, but not all of the R_Z parameters in the circuit. The remaining R_Z gates must be performed by approximation using a R_Z to Clifford+T synthesis algorithm. The results from approximate synthesis of two sizes of QFT are shown in table 4.2. We were able to approximate a 3-qubit QFT, which the technique from [16] was unable to finish within 24 hours, as well as a 4-qubit QFT.

4.4 Selection of Relaxed T-Count Function

The only requirement for the Relaxed T-Count Function presented in equation (3.4) to give correct results is that it has minima in the correct locations. The design of

the function should be selected to facilitate a fast optimization time. We have tried several different functions and measured their effect on optimization performance.

Our functions are generated by using two periodic functions, the triangle wave mentioned previously, and a sinusoidal wave. These are shown in equation (4.1). The advantage of the linear function is that it is simpler to calculate and does not have a vanishing gradient as it approaches the minima. The advantage of the sinusoidal function is that it is differentiable everywhere, such that it does not have a cusp at each minima.

$$\begin{aligned} \text{Sinusoidal } T\text{-Count}(\theta) &= \sin^2(\omega\theta) \\ \text{Linear } T\text{-Count}(\theta) &= \left|\left(\theta - \frac{\pi}{2\omega} \mod \frac{\pi}{\omega}\right) - \frac{\pi}{2\omega}\right| \frac{2\omega}{\pi} \end{aligned} \tag{4.1}$$

These periodic functions may be used directly, or modified with a profile function. We tried two profile functions. These profile functions are designed to follow strict requirements such that a solution with one parameter at a minimum and another at a large value will be weighted more favorably than a function with two parameters with low but not minimized values. The exponential function is designed to guarantee that $f(x+\delta) + f(x-\delta) < 2f(x) \forall x, \delta$. The logarithmic function is designed to follow the stronger condition $f(x\delta) + f(x/\delta) \leq 2f(x) \forall x, \delta$. These profile functions are shown in equation (4.2).

$$\begin{aligned} f_{\text{exponential}}(x) &= 2\left(\frac{1}{2} - \left(\frac{1}{2}\right)^{x+1}\right) \\ f_{\text{logarithmic}}(x) &= \log(x+1) \end{aligned} \tag{4.2}$$

We implemented all of these functions, such that we have 6 variants to try for the Relaxed T-Count. We compared the effect on performance by running sequential two-pass multistart optimization on a 3-qubit QFT circuit 10 times for each function, and averaged the running time. The results are presented in figure 4-2.

The linear function performed 36% faster than the sinusoidal function when run without any profile function. The profile functions decreased performance by up to 69%. Based on these observations, we chose the linear Relaxed T-Count function

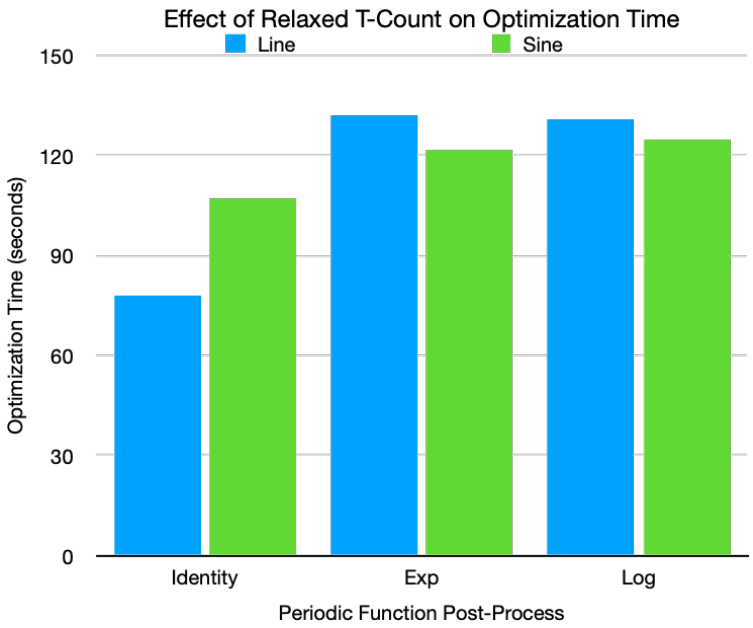

Figure 4-2: *Comparison of Relaxed T-Count variants in the effect they have on optimization time. The horizontal axis shows which profile function was applied, and the vertical axis shows the mean time to perform sequential two-pass multistart optimization on the 3-qubit QFT circuit. The blue bars present the data from the linear triangle wave as the periodic function, and the green bars present the data from the sinusoidal wave.*

without a profile function. Although the profile functions did not increase the speed of optimization, it is possible that they guide the optimization towards solutions where more than one gate can be successfully rounded at a time. We leave the application of profile functions for future study.

Chapter 5

Conclusion and Future Work

We provide a technique for numerical synthesis of arbitrary multi-qubit unitaries with low T-count. This procedure involves first using numerical synthesis techniques to produce a Clifford+R_Z circuit, and then applying sequential optimization using a multistart two-pass optimization approach to replace as many R_Z gates as possible with 0 or 1 T gates, before approximating the remaining R_Z gates with established approaches. This procedure was able to significantly reduce the number of R_Z gates that must be approximated. For gates that can be performed exactly, it was able to remove all R_Z gates, resulting in exact synthesis. The Clifford+T circuits produced through exact synthesis match the best known results for synthesizing these circuits by using other techniques [1,16,38] (note that some of these other techniques optimize T-depth as well as T-count, which we currently do not account for).

Numerical quantum gate synthesis enables powerful optimization techniques that can take advantage of circuit identities that would be difficult to discover by hand [1,35,38,41,56,58]. Our optimization approach enables numerical synthesis to produce circuits with low T-count. T gates require an order of magnitude more resources in the form of qubits and runtime when compared to Clifford gates, and therefore form a bottleneck to implementing practical algorithms. Real-world applications, such as quantum chemistry simulations or 2000-bit number factorization, require on the order of 10^8 to 10^{14} T-gates with current techniques [7,8]. Techniques to minimize the T-count bring applications such as quantum chemistry simulations closer to reality.

Thus far, we have implemented our sequential two-pass multistart optimization algorithm as a post-processing step for circuits generated with the `qsearch` synthesis algorithm [12, 13, 50], although it should work just as well with any other numerical synthesis algorithm that can produce Clifford+R_Z circuits such as the algorithm presented in [34]. The next step will be to combine our optimization procedure with a circuit partitioning scheme that will partition a large circuit into smaller subcircuits, optimize them, and then put them back together. Such a scheme would allow our algorithm to be run on much larger circuits than those that numerical synthesis algorithms can handle.

Additionally, the algorithm itself has room for improvement. Rounding all parameters that fall within a threshold of an exact multiple of $\frac{\pi}{4}$ instead of just the closest parameter could offer a $O(l)$ speed-up (where l is the number of R_Z gates in the original circuit) if the threshold is chosen well. Another optimization would be to work with U_3 gates instead of R_Z gates. Most Clifford+T synthesis approaches focus on R_Z gates, but using an algorithm to convert U_3 gates directly to R_Z gates without using the $R_Z S_X R_Z S_X R_Z$ format as an intermediate could allow more opportunities for different parameters in the circuit to interact to allow reduced T-counts.

The algorithm presented here offers a way to bridge the gap between numerical synthesis and Clifford+T synthesis. This development paves the way for new approaches to quantum circuit optimization for fault-tolerant quantum computing.

Bibliography

[1] Matthew Amy, Dmitri Maslov, Michele Mosca, and Martin Roetteler. A meet-in-the-middle algorithm for fast synthesis of depth-optimal quantum circuits. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 32(6):818–830, 2013.

[2] Sean D. Barrett and Pieter Kok. Efficient high-fidelity quantum computation using matter qubits and linear optics. *Physical Review A*, 71(6), jun 2005.

[3] Alex Bocharov, Martin Roetteler, and Krysta M. Svore. Efficient synthesis of universal repeat-until-success quantum circuits. *Phys. Rev. Lett.*, 114:080502, Feb 2015.

[4] Héctor Bombín. Gauge color codes: optimal transversal gates and gauge fixing in topological stabilizer codes. *New Journal of Physics*, 17(8):083002, aug 2015.

[5] Sergey Bravyi and Alexei Kitaev. Universal quantum computation with ideal clifford gates and noisy ancillas. *Phys. Rev. A*, 71:022316, Feb 2005.

[6] Sergey Bravyi and Robert König. Classification of topologically protected gates for local stabilizer codes. *Phys. Rev. Lett.*, 110:170503, Apr 2013.

[7] Andrew M. Childs, Dmitri Maslov, Yunseong Nam, Neil J. Ross, and Yuan Su. Toward the first quantum simulation with quantum speedup. *Proceedings of the National Academy of Sciences*, 115(38):9456–9461, sep 2018.

[8] Hyeongrak Choi, Mihir Pant, Saikat Guha, and Dirk Englund. Percolation-based architecture for cluster state creation using photon-mediated entanglement between atomic memories. *npj Quantum Information*, 5(1):1–7, 2019.

[9] Jerry M. Chow, A. D. Córcoles, Jay M. Gambetta, Chad Rigetti, B. R. Johnson, John A. Smolin, J. R. Rozen, George A. Keefe, Mary B. Rothwell, Mark B. Ketchen, and M. Steffen. Simple all-microwave entangling gate for fixed-frequency superconducting qubits. *Phys. Rev. Lett.*, 107:080502, Aug 2011.

[10] J. I. Cirac and P. Zoller. Quantum computations with cold trapped ions. *Phys. Rev. Lett.*, 74:4091–4094, May 1995.

[11] Sam R. Cohen and Jeff D. Thompson. Quantum computing with circular rydberg atoms. *PRX Quantum*, 2:030322, Aug 2021.

[12] Marc G. Davis, Ethan Smith, Ana Tudor, Koushik Sen, Irfan Siddiqi, and Costin Iancu. Towards optimal topology aware quantum circuit synthesis. In *2020 IEEE International Conference on Quantum Computing and Engineering (QCE)*, pages 223–234, 2020.

[13] Marc G. Davis, Ethan Smith, and Ed Younis. qsearch. https://github.com/BQSKit/qsearch, 2020.

[14] Christopher M. Dawson and Michael A. Nielsen. The solovay-kitaev algorithm, 2005.

[15] Yao-Min Di and Hai-Rui Wei. Optimal synthesis of multivalued quantum circuits. *Phys. Rev. A*, 92:062317, Dec 2015.

[16] Vlad Gheorghiu, Michele Mosca, and Priyanka Mukhopadhyay. T-count and t-depth of any multi-qubit unitary. *npj Quantum Information*, 8, 11 2022.

[17] Alexei Gilchrist, Nathan K. Langford, and Michael A. Nielsen. Distance measures to compare real and ideal quantum processes. *Phys. Rev. A*, 71:062310, Jun 2005.

[18] Brett Giles and Peter Selinger. Exact synthesis of multiqubit clifford+t circuits. *Phys. Rev. A*, 87:032332, Mar 2013.

[19] Timothée Goubault de Brugière, Marc Baboulin, Benoît Valiron, and Cyril Allouche. Synthesizing Quantum Circuits via Numerical Optimization. In *19th International Conference in Computational Science - ICCS 2019*, volume 11537 of *Lecture Notes in Computer Science*, pages 3–16, Faro, Portugal, June 2019.

[20] Aram W. Harrow, Benjamin Recht, and Isaac L. Chuang. Efficient discrete approximations of quantum gates. *Journal of Mathematical Physics*, 43(9):4445–4451, sep 2002.

[21] Kentaro Heya and Naoki Kanazawa. Cross cross resonance gate, 2021.

[22] He-Liang Huang, Dachao Wu, Daojin Fan, and Xiaobo Zhu. Superconducting quantum computing: A review, 2020.

[23] Raban Iten, Roger Colbeck, Ivan Kukuljan, Jonathan Home, and Matthias Christandl. Quantum circuits for isometries. *Phys. Rev. A*, 93:032318, Mar 2016.

[24] Navin Khaneja and Steffen Glaser. Cartan decomposition of su(2n̂), constructive controllability of spin systems and universal quantum computing, 2000.

[25] Sumeet Khatri, Ryan LaRose, Alexander Poremba, Lukasz Cincio, Andrew T. Sornborger, and Patrick J. Coles. Quantum-assisted quantum compiling. *Quantum*, 3:140, may 2019.

[26] A. Yu. Kitaev, A. H. Shen, and M. N. Vyalyi. *Classical and Quantum Computation*. American Mathematical Society, USA, 2002.

[27] Vadym Kliuchnikov. Synthesis of unitaries with clifford+t circuits, 2013.

[28] E. Knill, R. Laflamme, and W. Zurek. Threshold accuracy for quantum computation, 1996.

[29] D. Kraft. *A Software Package for Sequential Quadratic Programming*. Deutsche Forschungs- und Versuchsanstalt für Luft- und Raumfahrt Köln: Forschungsbericht. Wiss. Berichtswesen d. DFVLR, 1988.

[30] P. Krantz, M. Kjaergaard, F. Yan, T. P. Orlando, S. Gustavsson, and W. D. Oliver. A quantum engineer's guide to superconducting qubits. *Applied Physics Reviews*, 6(2):021318, jun 2019.

[31] Stefan Krastanov, Mikkel Heuck, Jeffrey H. Shapiro, Prineha Narang, Dirk R. Englund, and Kurt Jacobs. Room-temperature photonic logical qubits via second-order nonlinearities. *Nature Communications*, 12, 01 2021.

[32] J M Kreikebaum, K P O'Brien, A Morvan, and I Siddiqi. Improving wafer-scale josephson junction resistance variation in superconducting quantum coherent circuits. *Superconductor Science and Technology*, 33(6):06LT02, apr 2020.

[33] Shaowei Li, Anthony D. Castellano, Shiyu Wang, Yulin Wi, Zhiguang Gong, Ming anmd Yan, Hao Rong, Hui Deng, Chen Zha, Cheng Guo, Lihua Sun, Chengzhi Peng, Xiaobo Zhu, and Jian-Wei Pan. Realisation of high-fidelity nonadiabatic cz gates with superconducting qubits. *npj Quantum Information*, 10 2019.

[34] Liam Madden and Andrea Simonetto. Best Approximate Quantum Compiling Problems. *arXiv e-prints*, page arXiv:2106.05649, June 2021.

[35] Esteban A Martinez, Thomas Monz, Daniel Nigg, Philipp Schindler, and Rainer Blatt. Compiling quantum algorithms for architectures with multi-qubit gates. *New Journal of Physics*, 18(6):063029, jun 2016.

[36] Faraz Najafi, Jacob Mower, Nicholas C. Harris, Francesco Bellei, Andrew Dane, Catherine Lee, Xiaolong Hu, Prashanta Kharel, Francesco Marsili, Solomon Assefa, Karl K. Berggren, and Dirk Englund. On-chip detection of non-classical light by scalable integration of single-photon detectors. *Nature Communications*, 6(1), jan 2015.

[37] Michael A. Nielsen and Isaac L. Chuang. *Quantum Computation and Quantum Information*. Cambridge University Press, 2022.

[38] Philipp Niemann, Anshu Gupta, and Rolf Drechsler. T-depth optimization for fault-tolerant quantum circuits. In *2019 IEEE 49th International Symposium on Multiple-Valued Logic (ISMVL)*, pages 108–113, 2019.

[39] Philipp Niemann, Robert Wille, and Rolf Drechsler. Advanced exact synthesis of clifford+t circuits. *Quantum Information Processing*, 19, 08 2020.

[40] Tirthak Patel, Ed Younis, Costin Iancu, Wibe de Jong, and Devesh Tiwari. Quest: Systematically approximating quantum circuits for higher output fidelity. In *Proceedings of the 27th ACM International Conference on Architectural Support for Programming Languages and Operating Systems*, ASPLOS '22, page 514–528, New York, NY, USA, 2022. Association for Computing Machinery.

[41] Tirthak Patel, Ed Younis, Costin Iancu, Wibe de Jong, and Devesh Tiwari. Quest: Systematically approximating quantum circuits for higher output fidelity. In *Proceedings of the 27th ACM International Conference on Architectural Support for Programming Languages and Operating Systems*, ASPLOS '22, page 514–528, New York, NY, USA, 2022. Association for Computing Machinery.

[42] S. Poletto, Jay M. Gambetta, Seth T. Merkel, John A. Smolin, Jerry M. Chow, A. D. Córcoles, George A. Keefe, Mary B. Rothwell, J. R. Rozen, D. W. Abraham, Chad Rigetti, and M. Steffen. Entanglement of two superconducting qubits in a waveguide cavity via monochromatic two-photon excitation. *Phys. Rev. Lett.*, 109:240505, Dec 2012.

[43] M. Powell. A view of algorithms for optimization without derivatives. *Mathematics TODAY*, 43, 01 2007.

[44] Neil J. Ross and Peter Selinger. Optimal ancilla-free clifford+t approximation of z-rotations. *Quantum Info. Comput.*, 16(11–12):901–953, sep 2016.

[45] Alessandro Rossi, Paul G. Baity, Vera M. Schäfer, and Martin Weides. Quantum computing hardware in the cloud: Should a computational chemist care? *International Journal of Quantum Chemistry*, 121(14), may 2021.

[46] Peter Selinger and Neil J. Ross. gridsynth. https://www.mathstat.dal.ca/ selinger/newsynth/, 2012.

[47] Vivek V. Shende, Stephen S. Bullock, and Igor L. Markov. Synthesis of quantum logic circuits. In *Proceedings of the 2005 Asia and South Pacific Design Automation Conference*, ASP-DAC '05, page 272–275, New York, NY, USA, 2005. Association for Computing Machinery.

[48] Shuai Shi, Biao Xu, Kuan Zhang, Gen-Sheng Ye, De-Sheng Xiang, Yubao Liu, Jingzhi Wang, Daiqin Su, and Lin Li. High-fidelity photonic quantum logic gate based on near-optimal rydberg single-photon source. *Nature Communications*, 13, 08 2022.

[49] Peter W. Shor. Scheme for reducing decoherence in quantum computer memory. *Phys. Rev. A*, 52:R2493–R2496, Oct 1995.

[50] Ethan Smith, Marc G. Davis, Jeffrey M. Larson, Ed Younis, Lindsay Bassman Oftelie, Wim Lavrijsen, and Costin Iancu. Leap: Scaling numerical optimization based synthesis using an incremental approach. *ACM Transactions on Quantum Computing*, aug 2022. Just Accepted.

[51] Anders Sørensen and Klaus Mølmer. Entanglement and quantum computation with ions in thermal motion. *Phys. Rev. A*, 62:022311, Jul 2000.

[52] Anders S. Sørensen and Klaus Mølmer. Measurement induced entanglement and quantum computation with atoms in optical cavities. *Phys. Rev. Lett.*, 91:097905, Aug 2003.

[53] Robert R. Tucci. An introduction to cartan's kak decomposition for qc programmers, 2005.

[54] K. X. Wei, E. Magesan, I. Lauer, S. Srinivasan, D. F. Bogorin, S. Carnevale, G. A. Keefe, Y. Kim, D. Klaus, W. Landers, N. Sundaresan, C. Wang, E. J. Zhang, M. Steffen, O. E. Dial, D. C. McKay, and A. Kandala. Quantum crosstalk cancellation for fast entangling gates and improved multi-qubit performance, 2021.

[55] K. Wright, K. M. Beck, S. Debnath, J.M. Amini, Y. Nam, N. Grzesiak, J.-S. Chen, N. C. Pisenti, M. Chmielewski, C. Collins, K. M. Hudek, J. Mizrahi, J. D. Wong-Campos, S. Allen, J. Apisdorf, P. Solomon, M. Williams, A. M. Ducore, A. Blinov, S. M. Kreikemeier, V. Chaplin, M. Keesan, C. Monroe, and J. Kim. Benchmarking an 11-qubit quantum computer. *Nature Communications*, 10, 11 2019.

[56] Xin-Chuan Wu, Marc Grau Davis, Frederic T. Chong, and Costin Iancu. QGo: Scalable Quantum Circuit Optimization Using Automated Synthesis. *arXiv e-prints*, page arXiv:2012.09835, December 2020.

[57] Xing-Can Yao, Tian-Xiong Wang, Hao-Ze Chen, Wei-Bo Gao, Austin G. Fowler, Robert Raussendorf, Zeng-Bing Chen, Nai-Le Liu, Chao-Yang Lu, You-Jin Deng, Yu-Ao Chen, and Jian-Wei Pan. Experimental demonstration of topological error correction. *Nature*, 482, 02 2012.

[58] Ed Younis and Costin Iancu. Quantum circuit optimization and transpilation via parameterized circuit instantiation, 2022.