Quantum Gate Synthesis

480441984

2022

1 Introduction

Quantum computing is still an emerging field. There is a large gap between the theoretical power of quantum computers and the physical devices available today. Algorithms can select operators from an infinite number of unitary matrices but quantum computers can only implement a finite set. Hence, we need algorithms to convert theoretical algorithms into hardware compliant circuits. There have been significant developments in gate decomposition and pulse-level control. However, no algorithm achieves all three; high approximation accuracy, low error rate, and low classical computational power. Improving circuit synthesis is essential if quantum computers are to one day outperform classical computers.

This literature review starts with an introduction to quantum mechanics and quantum computing (section 2), the current state of quantum computing today (section 3), gate decomposition (section 4) and pulse control (section 5). This paper focuses on the NISQ era, but highlights the importance of synthesis for the future fault-tolerant era.

2 Basics of Quantum Computing

This section will provide an overview of the linear algebra and quantum mechanics necessary for quantum computing.

2.1 Qubits

Classical bits can either be in the 0 or 1 state. Quantum computing uses qubits instead. Qubits also have two states, $|0\rangle$ and $|1\rangle$. The $|.\rangle$ notation indicates that the object is a vector.

$$|0\rangle = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \quad |1\rangle = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \tag{2.1}$$

However, a single qubit can be in a superposition of these two states at any one time, represented by

$$|\psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle \tag{2.2}$$

where $|\alpha|^2 + |\beta|^2 = 1$ and $\{\alpha, \beta\} \in \mathbb{C}$. But, qubits can only be in a superposition until they are measured [56]. The measurement result can only be 0 or 1. $|\alpha|^2$ represents the probability of measuring the qubit as a 0, and $|\beta|^2$ the probability of measuring 1. After measurement the qubit will no longer be in a superposition but in a state consistent with its measurement result. For example, if the measurement of $|\psi\rangle$ is 0, then the post-measurement state will be $|0\rangle$.

Because $|\alpha|^2 + |\beta|^2 = 1$ we can rewrite

$$|\psi\rangle = e^{i\gamma}(\cos\frac{\theta}{2}|0\rangle + e^{i\phi}\sin\frac{\theta}{2}|1\rangle) \tag{2.3}$$

where θ, ϕ and γ are real numbers and e represents Euler's formula, that is $e^{i\gamma} = \cos\gamma + i\sin\gamma$. We can ignore the factor $e^{i\gamma}$ out the front because it has no observable effect, hence we can write

$$|\psi\rangle = \cos\frac{\theta}{2}|0\rangle + e^{i\phi}\sin\frac{\theta}{2}|1\rangle \tag{2.4}$$

Since there are only two free variables θ and ϕ, they can be used to represent a point on a three dimensional sphere [56], see Figure 1. This is referred to as a Bloch sphere. There are infinite points on the Bloch sphere, but as discussed above when a qubit is measured it will either collapse into the $|0\rangle$ or $|1\rangle$ state. This is a useful representation of a single qubit, but there is simple known visualisation for multiple qubits.

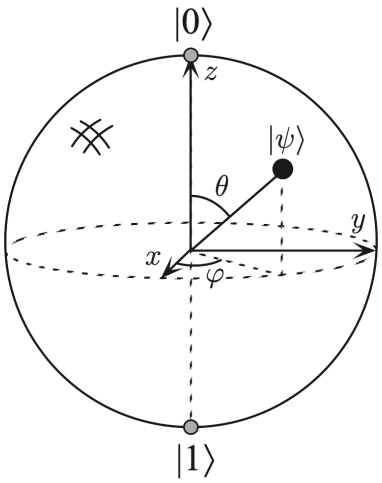

Figure 1: Bloch sphere representation of a single qubit [56]

The complex conjugate transpose of a vector is referred to as the dual

$$\langle\psi| = |\psi\rangle^\dagger \tag{2.5}$$

where $|\psi\rangle^\dagger = |\psi\rangle^{*T}$.

The inner product takes two input vectors $|v\rangle$ and $|w\rangle$ from a vector space and produces a complex number as output, denoted $(|v\rangle, |w\rangle) = \langle v|w\rangle$. The inner product operating on complex numbers has special properties:

1. It is linear in the second argument, $(|v\rangle, \sum_i \lambda_i |w_i\rangle) = \sum_i \lambda_i (|v\rangle, |w_i\rangle)$
2. There is conjugate symmetry, $\langle v|w\rangle = \langle w|v\rangle)^*$
3. Is positive definite, $\langle v|v\rangle \geq 0$

Vectors $|v\rangle$ and $|w\rangle$ are *orthogonal* if $\langle v|w\rangle = 0$. The *norm* of a vector $|v\rangle$ is defined by

$$|| \, |v\rangle \, || = \sqrt{\langle v|v\rangle} \tag{2.6}$$

The outer product is $|\psi\rangle \langle\psi|$, which results in an $n \times n$ matrix, where n is the number of rows of $|\psi\rangle$.

2.2 Gates

Classical computers use logical gates like AND, OR, and NOT, which are not reversible apart from NOT. In quantum, all gates are represented by a *unitary* matrix. These are also referred to as operators. A matrix U is said to be *unitary* if

$$U^\dagger U = I \tag{2.7}$$

where I is the identity matrix. Clearly all operators can be reversed, since the inverse always exists and is equal to its conjugate transpose: $U^{-1} = U^\dagger$.

Unitary matrices are geometrically important because they preserve the inner products between vectors, which means that it preserves lengths and angles between vectors. See the following where $|v\rangle$ and $|w\rangle$ are any two vectors, and U some arbitrary unitary matrix

$$(U \, |v\rangle \, , U \, |w\rangle) = \langle v| \, U^\dagger U \, |w\rangle = \langle v| \, I \, |w\rangle = \langle v|w\rangle \tag{2.8}$$

Operators can be chained, applying operators A and then B on $|\psi\rangle$ is denoted $B(A \, |\psi\rangle) = BA \, |\psi\rangle$.

The Pauli matrices are some of the most common and frequent operators

$$I \equiv \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \quad X \equiv \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \quad Y \equiv \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix} \quad Z \equiv \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \tag{2.9}$$

Other common quantum gates include

$$H \equiv \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \quad S \equiv \begin{bmatrix} 1 & 0 \\ 0 & i \end{bmatrix} \quad T \equiv \begin{bmatrix} 1 & 0 \\ 0 & \exp(i\pi/4) \end{bmatrix} \tag{2.10}$$

Note that $T^4 = S^2 = Z$, since T is a $\pi/8$ rotation, S is a $\pi/4$ rotation, and Z is a $\pi/2$ rotation around the Z axis. The Pauli group plus the H and S gates are referred to as the Clifford group.

If we think of the qubit in terms of the Bloch sphere, each operator can be visualised as some rotation around an axis. In particular, we can exponentiate the X, Y, Z matrices to define the *rotation operators* about the $\hat{x}, \hat{y}$, and $\hat{z}$ axis

$$R_x(\theta) \equiv e^{-\theta X/2} = \cos\frac{\theta}{2} I - i \sin\frac{\theta}{2} X = \begin{bmatrix} \cos\frac{\theta}{2} & -i\sin\frac{\theta}{2} \\ -i\sin\frac{\theta}{2} & \cos\frac{\theta}{2} \end{bmatrix} \tag{2.11}$$

$$R_y(\theta) \equiv e^{-\theta X/2} = \cos\frac{\theta}{2} I - i \sin\frac{\theta}{2} y = \begin{bmatrix} \cos\frac{\theta}{2} & -i\sin\frac{\theta}{2} \\ \sin\frac{\theta}{2} & \cos\frac{\theta}{2} \end{bmatrix} \tag{2.12}$$

$$R_z(\theta) \equiv e^{-\theta Z/2} = \cos\frac{\theta}{2} I - i \sin\frac{\theta}{2} Z = \begin{bmatrix} e^{-i\theta/2} & 0 \\ 0 & e^{i\theta/2} \end{bmatrix} \tag{2.13}$$

Controlled gates enable the logic 'If A is true, then do G'. One of the most common controlled operators is the Controlled-NOT (CNOT). It is a two qubit gate, acting on a control qubit and a target qubit. If the target qubit is $|1\rangle$ then the target qubit is flipped, if $|0\rangle$ the target qubit remains

the same. This is represented by $|c\rangle |t\rangle \rightarrow |c\rangle |t \oplus c\rangle$, where $\oplus$ is the classical Exclusive OR (XOR). The matrix representation

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \tag{2.14}$$

2.3 Tensor Products

In order to represent systems of multiple qubits we need the tensor product, a way to put vector spaces together to form a larger space. For example the basis states of a single qubit system is $|0\rangle, |1\rangle$, and for a two-qubit system it will be $|00\rangle, |01\rangle, |10\rangle, |11\rangle$. $|vw\rangle$ is shorthand notation for $|v\rangle \otimes |w\rangle$. This is known as the *Kronecker product*.

$$|\psi\rangle = \alpha_{00} |00\rangle + \alpha_{01} |01\rangle + \alpha_{10} |10\rangle + \alpha_{11} |11\rangle \tag{2.15}$$

Again the probability of measuring x is equal to $|\alpha_x|^2$, where $x \in \{00, 01, 10, 11\}$. A quantum system with n qubits has 2^n quantum states, and therefore needs 2^n complex numbers to describe it. Suppose that A is a m by n matrix, and B is a p by q matrix. This results in an mp by nq matrix

$$A \otimes B \equiv \begin{bmatrix} A_{11}B & A_{12}B & \dots & A_{1n}B \\ A_{22}B & A_{22}B & \dots & A_{2n}B \\ \vdots & \vdots & \vdots & \vdots \\ A_{m1}B & A_{m2}B & \dots & A_{mn}B \end{bmatrix} \tag{2.16}$$

For example

$$|0\rangle \otimes |0\rangle = |00\rangle = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \otimes \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 \times 1 \\ 1 \times 0 \\ 0 \times 1 \\ 0 \times 0 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{2.17}$$

Another example for clarity

$$\begin{bmatrix} 1 \\ 2 \end{bmatrix} \otimes \begin{bmatrix} 3 \\ 4 \end{bmatrix} = \begin{bmatrix} 1 \times 3 \\ 1 \times 4 \\ 2 \times 3 \\ 2 \times 4 \end{bmatrix} = \begin{bmatrix} 3 \\ 4 \\ 6 \\ 8 \end{bmatrix} \tag{2.18}$$

2.4 Universality

A universal set of gates universal means that an arbitrary unitary qubit operation can be approximated to an arbitrary accuracy $\epsilon > 0$ with a circuit only consisting of these gates [56]. The Clifford+T set equal to $\{X, Y, Z, H, S, T\}$ is an example of a common universal family for single qubit operators [40].

Furthermore, any unitary operation on n qubits may be implemented exactly by composing single qubit and controlled-gates [56]. But, not all unitary operations can be efficiently implemented that is with a polynomial number of gates. Some unitary operations on n qubits require $\Omega(2^n \log(1/\epsilon)/\log(n))$ gates to approximate to within a distance ϵ using any finite set of gates [6].

2.5 Process Tomography

The quantum equivalent of classical system identification is *quantum process tomography*. This enables us to experimentally determine the dynamics of an arbitrary quantum system.

First, we will start with *quantum state tomography*: experimentally determining an unknown quantum state. Only measuring a qubit once, will not tell us much. But if we are able to measure it many times, in different basis, we can effectively reconstruct the state.

Now, with state tomography, we can build on this to find useful representations of quantum processes. We are looking to find a set of operators to represent a channel $\mathcal{E}$ applied to a density operator ρ such that

$$\mathcal{E}(\rho) = \sum_i E_i \rho E_i^\dagger \tag{2.19}$$

To do this, we choose d^2 pure quantum states $|\psi_1\rangle, ..., |\psi_{d^2}\rangle$ so that the corresponding density matrices $|\psi_1\rangle\langle\psi_1|, ..., |\psi_{d^2}\rangle\langle\psi_{d^2}|$ form a basis set for the space of matrices. Let the state space of the system have d dimensions; for a single qubit $d = 2$. Then, for each state $|\psi_j\rangle$ we apply the chosen process and we have $\mathcal{E}(|\psi_j\rangle\langle\psi_j|)$.

To convert the experimental results to operators it is useful to consider E_i as a combination of a fixed set of operators $\tilde{E}_i$ that form a basis for the set of operators on the state space, so that

$$E_i = \sum_m e_{im} \tilde{E}_m \tag{2.20}$$

where $e_{im} \in \mathbb{C}$. So we can rewrite equation 2.19 as

$$\mathcal{E}(\rho) = \sum_{mn} \tilde{E}_m \rho \tilde{E}_n^\dagger \chi_{mn} \tag{2.21}$$

where $\chi_{mn} = \sum_i e_{im} e_{in}^*$ are the entries of the χ matrix. So, once the set of operators E_i has been fixed, $\mathcal{E}$ can be completely described by complex numbers. This is known as the *chi matrix representation*. There are $d^4 - d^2$ independent real parameters for a χ matrix in general. A general linear map of d by d complex matrices to d by d matrices is described by d^4 independent parameters, but the additional constraints that ρ remains Hermitian with trace one; that is, the completeness relation

$$\sum_i E_i^\dagger E_i = I \tag{2.22}$$

creates an extra d^2 constraints. For example, in the case of a single qubit there are 12 free parameters but for two qubits there are 240. For complete instructions on how to derive the χ matrix see [56] Chapter 8.4.2.

Once χ is defined, the operator-sum representation follows immediately. Let the unitary matrix $U^\dagger$ diagonalize χ,

$$\chi_{mn} = \sum_{xy} U_{mx} d_x \delta_{xy} U_{ny}^*. \tag{2.23}$$

Then the operation elements of $\mathcal{E}$ are

$$E_i = \sqrt{d_i} \sum_j U_{ji} E_j \tag{2.24}$$

3 Quantum Computing Today

Theoretically quantum algorithms have the ability to search an unordered list in $O(\sqrt{N})$ using Grover's algorithm [34]. This is not physically possible with classical computation, a classical computer would have to scan the whole list, resulting in a worst case time complexity $O(N)$. Quantum computers with 50-100 qubits could outperform classical computers, but presently the noise severely limits the length of circuits that can be executed reliably [59]. This is known as the current Noisy Intermediate-Scale Quantum (NISQ) era.

3.1 Quantum Errors

First, let's define the three main sources of noise in qauntum computers:

1. **State Preparation and Measurement Errors (SPAM)**. If the state prepared at the beginning of execution is not what we expect it to be, these are called state preparation errors [20]. For example, if we want to start with qubits in the $|0\rangle$ state, but the state we prepared is not $|0\rangle$. Measurement errors occur when trying to read out results [20].

2. **Gate Errors**. These are the errors introduced by applying gates. This can be systematic; from poorly implemented measurement operators, or from the noise introduced by the gate to the system [20]. In particular the quality of multi-qubit gates is an important because it is this entanglement that enables quantum algorithms to demonstrate advantage over classical computation.

3. **Decoherence Errors**. The decay of the quantum information due to interaction with the environment over time. This decay of qubits' superposition makes long computations impossible [58].

3.2 NISQ Era

Superconducting qubits are so far one of the best approaches for implementing quantum logic with sufficiently high controllability and low noise [38]. Microwave irradiation is used to implement quantum gates [38]. Every gate is mapped to a pulse. Electromagnetic coupling to the qubit with microwaves at the qubit transition frequency drives Rabi oscillations. Control of the phase and amplitude of the drive.

Quantum computers only implement a limited set of gates, or more specifically pulses, because the computers must be calibrated at least every 24 hours [67]. Calibration involves a series of experiments to determine optimal control parameters [5]. Google's 54-qubit device, with only a single type of two-qubit gate, requires approximately 4 hours of calibration each day [5]. Consequently, to balance downtime and accuracy, they only tune a small set of computationally powerful gates: a universal gate set. Since, any n qubit circuit can be reduced to single qubit operations and a controlled gate, the problem of creating a hardware compliant circuit can be reduced problem of single qubit unitary approximation.

IBM has built quantum machines and exposed them for public use through the cloud. Alongside this, they launched Qiskit an open source quantum computing language that can be used to write corresponding circuits. IBM state their basis gate set is

$$\{U_1(\lambda), U_2(\lambda, \phi), U_3(\lambda, \phi, \theta), X, R_z(\theta), X, CNOT\} \tag{3.1}$$

but in fact, the physical gate set is essentially only [2]

$$\{R_z(\theta), R_x(\frac{\pi}{2}), CNOT\} \tag{3.2}$$

$R_x(\frac{\pi}{2})$ is a $\frac{\pi}{2}$ rotation about the X axis which is similar to the Hadamard matrix.

$$R_x(\frac{\pi}{2}) = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & -i \\ -i & 1 \end{bmatrix} \tag{3.3}$$

CNOT is the only 2 qubit gate that is implemented. This is built from echoed cross-resonance pulses [13, 61] with rotary tones [63]. CNOTs can only be applied to qubits that are connected and in general the connectivity graph is not complete. This is known as chip topology. A complete graph for 5 qubits would have 20 connections, but for example *imbq_quito* only has 4 connections, see Figure 2. This means CNOTs can only be applied between some pairs of qubits. Furthermore, the CNOT error rate varies depending on the pair, from approximately 1% up to 2% [15]. Two qubit gates are approximately 34 times more expensive than single qubit ones. See in Figure 2, the average CNOT error rate is 1.294% compared to 0.038% for the H gate [15]. The inaccuracies of the gates applied accumulate, rendering long circuits unreliable [58]. Existing computers currently only offer accuracy up to 10^{-4} [5], while useful quantum algorithms require accuracy of 10^{-10} because of their circuit length [66].

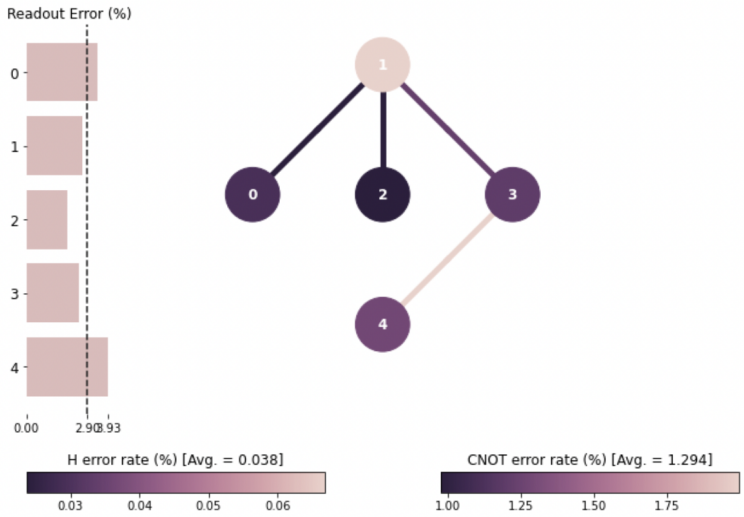

Figure 2: ibmq_quito connectivty graph [15]

The decoherence time of qubits today is extremely short, around 0.1ms [60]. This restricts near-term experiments to approximately 100 gates [59]. Useful quantum algorithms will require 10^{10} gates or more [66]. Thus, latency reduction is critical for the NISQ era to reduce the coherence error [56], and enable longer circuits to be run.

Since the qubits are not deterministic, multiple shots of a each experiment are needed to get a distribution of results. Increasing the number of shots, minimises the sampling error for any experiment. IBM limits the number of shots per experiment to 20,000. Related experiments must also be performed in the same calibration period. For more assurance, each experiment should be run multiple times. Then statistical measures such as the mean, standard deviation, and confidence intervals can be generated.

3.3 Fault Tolerant

Fault-tolerant quantum computing is eventually the goal: devices with imperfect elementary components that still work effectively [58]. Once the average probability of error per quantum gate is less than a certain critical value, the *accuracy threshold*, arbitrarily long quantum computation could be executed reliably.

Fault-tolerant gates must prevent propagation of single qubit errors to other qubits. Some gates can be implemented directly with transversal operations, but these alone are not a universal set. To create a universal set of gates complicated constructions with ancilla states and fault-tolerant measurement are needed [68]. For example, T gates are significantly more expensive than Clifford gates [68].

There are only a few gates that can be directly implemented in a fault-tolerant manner [68], due to fault tolerance protocols and error correction. The fault tolerance protocols also greatly reduce the efficiency of given instructions [4]. The Clifford+T set is one of the most common choices for fault-tolerant computers [40].

4 Gate Decomposition

Since any unitary gate can be applied to a qubit, there are infinitely many possibilities, but both NISQ and fault-tolerant quantum computers can only implement a discrete set. So, we need a way to decompose any arbitrary gate into a sequence of basis gates. For example, let the basis set be $B = \{B_1, B_2, ..., B_n\}$, and the target unitary be U, we are looking for some sequence

$$B_y B_x ... B_a \approx U \tag{4.1}$$

where B_a is the first applied gate. U_s is equal to all the basis gates multiplied

$$U_s = B_y B_x ... B_a \tag{4.2}$$

Some operators can be exactly decomposed, but many common unitaries are known to be impossible to implement with the Clifford+T basis set. The phase rotations of $e^{2\pi i/2^m}$, commonly seen in the quantum Fourier transform, are not implementable using Clifford and T gates for $m \geq 4$ [68].

There are three overall objectives for synthesis.

1. **Accuracy**. That is minimising the distance $||U - U_S||$, where U is the target unitary and U_S is the computed solution. An error threshold ϵ is used for termination when $||U - U_S|| \leq \epsilon$. For matrices there are many distance methods available. Early synthesis algorithms including [18] use the diamond norm [17]

   $$\sup_{||\psi||=1} = ||(\mathrm{U} - \mathrm{S})\psi|| < \epsilon \tag{4.3}$$

   To reduce computational overhead, more recent efforts [39, 36] use the Hilbert-Schmidt inner product between the conjugate transpose of U and U_s

   $$\langle U|U_s\rangle_{HS} = Tr(U^\dagger U_s) \tag{4.4}$$

2. **Error rate**. How long are the sequences and what is the overall cost of the sequence. Each gate has its own error and these errors are additive, see Box 4.1 of [56] for full proof. Shorter sequences will have lower error rates.

3. **Classical computational power**. How much memory and time is required to synthesize the circuit. Ideally logarithmic or polynomial time algorithms instead of exponential.

Additionally, single qubit and CNOT gates can be used to implement an arbitrary unitary operation on n qubits. See section 4.5.2 of [56] for a full proof. This means the problem of decomposing any circuit can be reduced the problem of approximating single and two-qubit unitaries. This is also what quantum computers implement today: only single and two-qubit gates [5].

4.1 Exhaustive Search

The simplest solution is exhaustive search. However, this quickly becomes infeasible. For a set G of size $g = |G|$ and a maximum sequence length of l, the search space is approximately $O(g^l)$. Take a small set of basis gates $g = 5$, and $l = 10$, the search tree is $5^{10} = 9,765,625$.

Improvements of [22] that ensure branches are unique only make sequence approximation feasible up to error $\epsilon = 10^{-4}$. At this point, classical computers run out of resources. While [43] combines algebraic and search methods, it is still only feasible up to $\epsilon = 10^{-15}$. These algorithms achieve optimal circuit sizes, but are at the extreme cost of exponential runtimes.

4.2 Solovay-Kitaev

The Solovay-Kitaev theorem is one of the first key developments in gate decomposition. They proved that if a set of single-qubit quantum gates generates a dense subset of $SU(2)$ then that set is guaranteed to fill $SU(2)$ quickly, meaning any desired gate can be approximated by a relatively short sequence of gates of length $O(\log^c(1/\epsilon))$. This means that a universal gate-set G can approximate any unitary $U \in SU(2)$ to accuracy ϵ by a finite sequence of gates from G of length $O(\log^c(1/\epsilon))$, where c is a constant.

This algorithm is general and can be applied to any set of basis gates. It initially searches for a base sequence that roughly approximates a target gate. However, the base sequences are generated by brute force and is exponential in the size of the basis gate set. Dawson and Nielsen state a sequence depth $l = 16$ is sufficient [18]. This is extremely resource expensive, for the Clifford+T set with 6 gates this results in the generation of $6^{16} \approx 2.81 \times 10^{12}$ sequences.

It then recursively appends other base sequences, which approximate the remaining distance from the current sequence to the target. This increases the sequence length by approximately a factor of 5 each iteration. Moreover, if the generated database of sequences are not sufficiently long, the appended sequences can be equal to the identity matrix I which does not improve the accuracy.

The best known upper bound on a circuit derived from the Solovay-Kitaev algorithm is $O(log^{3+\delta}(1/\epsilon))$, where δ can be chosen arbitrary small [37] and $\Omega(log(1/\epsilon))$ is the lower bound [32]. Dawson and Nielsen present pseudo-code and a direct mathematical proof. Applying quantum gates have significant error, in practice longer sequences likely reduce rather than improve the approximation accuracy. The Solovay-Kitaev algorithm alone is not suitable for the NISQ era.

We can use additional resources to improve the running time. The phase kickback version of the Solovay-Kitaev uses a special resource state $|\gamma\rangle$ on $O(\log(1/\epsilon))$ qubits allows to achieve the desired accuracy of approximation by a depth $O(\log(\log(1/\epsilon)))$ circuit containing $O(\log(1/\epsilon)$ gates. This is at the cost of $O(\log^2 1/\epsilon)$ ancillae bits and a circuit of depth $O(\log^2(\log(1/\epsilon)))$ containing $O(\log^2(1/\epsilon)\log\log(1/\epsilon))$ gates. Though exact preparation of $|\gamma\rangle$ is not possible using only the

Clifford+T set and qubits initialized to $|0\rangle$ [41, 24]. Even with this space-time trade-off, the gate sequences are too long for NISQ devices.

4.3 Normal Forms

In 2008, a normal form for sequences over {H, S, T} that produces unique elements in $SU(2)$ [49]. Every circuit over this basis can easily be transformed into a normal form, which is unique for the sequence. It follows that the number of unique unitary operators that contain at most n T-gates is $192(3 \times 2^n - 2)$. This is only applicable to single qubit unitaries.

Additionally, the normal form of [49] is expressed in $SU(2)$, which makes the algebra sensitive to the sign of the global phase. In 2012, [10] addressed this with their canonical form extending the normal form to produce elements in $PSU(2)$, which ignore the global phase. Again, this is only for the basis set {H, S, T}. This considerably reduces sequence search space, although it is still exponential in the sequence length. The normal form ensures that there are no duplicates in the database and that each unique sequence is depth-optimal. So, applying this to the base case of the Solovay-Kitaev algorithm results in $\epsilon = 5 \times 10^{-8}$, a factor of 10^{-6}. They experiment with a 1GB, 2GB, and 4GB base database. There is improvement expanding the database size from 1GB to 2GB, but no improvement from 2GB to 4GB.

To date, there are no known normal forms for multi-qubit circuits.

4.4 Direct Synthesis

A unitary matrix only has an exact representation over the Clifford+T gate set with local ancillas if and only if its entries are in the ring $\mathbb{Z}[\frac{1/\sqrt{2}}{i}]$ [24]. This ring consists of complex numbers in the form

$$\frac{1}{2^n}(a + bi + c\sqrt{2} + di\sqrt{2}) \tag{4.5}$$

where $n \in \mathbb{N}$ and $a, b, c, d \in \mathbb{Z}$. Moreover, [24] show that one ancilla always suffices. From this, a simple yet inefficient algorithm can be derived that generates $O(3^{2^n} nk)$ elementary Clifford+T gates. This is an upper bound for operators with an exact Clifford+T gate set. [41] present an asymptotically optimal algorithm for finding a circuit with the minimal number of Hadamard gates and asymptotically shortest circuits. Though this is only if the entries belong to the ring [42]. There is currently no efficient ring round off procedure. If the square norm of an element of the single qubit unitary matrix $|u_i j|^2$ can be represented as $(a + \sqrt{2}b)/2^n$, where a and b are integers such that $GCD(a, b)$ is odd, the total number of gates required to synthesise the unitary is linear, $\Theta(n)$.

4.5 Gate Costs

The Solovay-Kitaev algorithm also does not consider that gates have different costs in implementation. [53] presents an algorithm that generates a database of cost-optimal sequences based on Dijkstra's algorithm. Each basis gate is assigned an individually chosen cost. Two ways to assign cost where tested. The average sequence cost decreases by up to $54 \pm 3\%$ when using the Z-rotation catalyst circuit approach and by up to $33 \pm 2\%$ when using the magic state distillation approach. The number of magic states is only a rough cost approximation function [53]. An exact cost calculation would need to consider qubits count, circuit depth, magic state distillation cost and any error-correcting implementation. Hence, the algorithm is dependent on the accuracy of the heuristic chosen. Furthermore, this research is only applicable to the Clifford hierarchy of gates. However, it could be applied base step of the Solovay-Kitaev when using the Clifford+T basis set.

In the NISQ era, CNOTs are error prone as discussed in section 3.2. To mitigate this, the algorithm of [17] uses an A* inspired algorithm to minimise CNOT gates. When comparing against state-of-the-art available synthesis packages they show 2.4× average (up to 5.3×) reduction in CNOT count. They also account for chip topology. They combine numerical optimisation with a tree search for the shortest path. The circuit is built in layers: at each step they add a fixed structure 2-qubit building block where chip connectivity allows it. The building block contains generic parameterised single qubit unitaries and CNOT gates. Note, $U_3(\lambda, \phi, \theta)$ is an example of a parameterised gate. This parameterised circuit is optimised with respect to the distance function with the algorithm of [57]. At each step, a heuristic is used to determine the next circuit to expand. The algorithm stops when the decomposed circuit is within distance ϵ of the target unitary. While they can find near optimal depth circuits, it is again very slow. Moreover, it does not account for the varying CNOT costs between different pairs of qubits. Though it is topology aware and easily re-targeted to new gate sets or topologies.

The benchmark suite is composed of third party algorithms used by other evaluation studies [7, 54]. They showcase the optimality of their solution by including circuits with known optimal implementations, such as Quantum Fourier Transform [55], HHL [31] and important quantum kernels such as Toffoli gate. They also consider circuits with no known optimal depth circuits from domain generators such as the Variational Quantum Eigensolver [50] (VQE) or Transverse Field Ising Models [7] (TFIM).

4.6 T count reduction

There has been significant work reducing the T-count and T-depth as they have been identified as important metrics for fault-tolerant computing. Let $\mathcal{T}(U)$ represent the T-count of the unitary U. Amy *et al.* [4] also proposed counting the number of T stages in a circuit, rather than the number of T gates. A T stage is a group of one or more T and or $T^\dagger$ gates on distinct qubits that can be performed simultaneously.

The meet-in-the-middle approach enables the search of circuits of depth l by only generating circuits of at most depth $\lceil l/2 \rceil$ [4]. It achieves exactly equivalent depth optimal circuit in time $O(d|B|^{d/2})$, where d is the depth of the circuit and B is the basis set. Amy *et al.* propose

Lemma 4.1. *Let $S_i \subset U(2^n)$ be the set of all unitaries implementable in depth i over the gate set B. Given a unitary U, there exists a circuit over B of depth l implementing U if and only if $S^\dagger_{\lfloor l/2 \rfloor} U \cap S_{\lceil l/2 \rceil} \neq \emptyset$*

Thus, generated sequences of depth i can be used to search for circuits of twice the depth. There exists a circuit of depth $2i-1$ or $2i$ implement U if and only if $S^\dagger_{i-1} U \cap S_i \neq \emptyset$ or $S^\dagger_i U \cap S_i \neq \emptyset$ respectively. At each iteration, $S^\dagger_{i-1} U$ and $S^\dagger_i U$ are checked to see if there are any collisions with S_i. If there is a collision, a circuit is returned, otherwise the circuit is extended. Circuits of increasing depth are generated iteratively, so it always terminates at the shortest matching depth sequence.

This can be used for multi-qubit circuits, but only deals with the case where unitaries can be exactly computed. This algorithm could be extended to look for collisions with S_i within ϵ instead of exact ones. It is another alternative to speed up the brute force step for the Solovay-Kitaev algorithm.

After proving their algorithm, they show its improvement over an existing Solovay-Kitaev open

source implementation. They also specify the computer and they tested it on to assist anyone trying to replicate their results. They used Debian Linux on their group's research server, containing a quad-core, 64-bit Intel Core i5 2.80GHZ processor and 16 GB RAM, plus an additional 16 GB of swap space.

While [4] can optimise any given cost function, [30] extend this work to directly optimise T-depth and show proofs of T-depth minimality for various 2-4 bit circuits. Since every unitary can be represented by alternating Clifford and T gates. This is used to derive another canonical channel representation. Instead of naively brute forcing over all circuits of alternating Clifford and T gates, a similar meet in the middle approach is used. They define sorted coset databases $\mathcal{D}_k^n$, all unitaries are unique and $\mathcal{T}(U) = k$, and n is the number of qubits. The algorithm is given a unitary U and integer m. It will determine if $\mathcal{T}(U) \leq m$ and if so return a circuit. It generates all coset databases $\mathcal{D}_k^n$ for $k \in \{1, 2, ..., \lceil \frac{m}{2} \rceil\}$. It uses binary search to see if there is an existing matching sequence where $\mathcal{T}(U) \leq \lceil \frac{m}{2} \rceil$. Otherwise, it begins searching the databases in order, starting with $r = \lceil \frac{m}{2} \rceil + 1$. For each $W \in \mathcal{D}^n_{r-\lceil \frac{m}{2} \rceil}$ it uses binary search to find $V \in \mathcal{D}^n_{\lceil \frac{m}{2} \rceil}$ such that $(W^\dagger \hat{U})^{(c)} = V^{(c)}$. This follows since the unitary U can be split such that $U = WV$ so $W^\dagger U = W^\dagger WV = V$. If it finds a match it will return a the sequence WV sequence where $\lceil \frac{m}{2} \rceil < \mathcal{T}(U) \leq m$. This runs in $O(2^{n^m} \text{poly}(m, N))$, an exponential running time because it generates all of the coset databases. Again, this is only for unitaries that can be approximated exactly.

5 Pulse Control

Pulse control is the lowest-level of quantum manipulation available. A *pulse* is a *time-series* of complex-valued amplitudes with a maximum unit norm, $[d_0, ..., d_{n-1}]$. Each d_j , $j \in \{0, ..., n-1\}$, is called a *sample* [3]. Every system has a cycle-time of `dt`. This is the finest time control of the pulse which is typically defined by the sample rate of the coprocessor's waveform generators. The ideal output has amplitude

$$D_j = Re[e^{i2\pi f j \texttt{dt} + \Phi} d_j] \tag{5.1}$$

at time $j\texttt{dt}$, where f and Φ are a modulation frequency and a phase.

There are same three overall objectives for pulse control:

1. **Accuracy**. How close is the computed pulse sequence to the target unitary.
2. **Error rate**. What is the overall error of applying the pulse. Note that sequence length is no longer necessarily a good approximation for the error rate.
3. **Classical computational power**. How much time and memory is required to find the pulse or circuit.

As discussed in section 4, any circuit can be decomposed into single qubit unitaries and CNOTs. For example, decomposing the Toffoli gate results in nine single-qubit gates and 6 CNOTs in Figure 3. A total of 15 gates. A 1% reduction in error per gate would result in a 15% reduction in error for this single gate. Therefore, incremental error improvement in one qubit pulses can achieve *exponential* gains for multi-qubit gates.

Methods are needed to optimise these laser pulses. It began with brute-force optimization of a few pulse parameters [33]. Prior to the early 2000s there was also pulse-timing control [65, 64, 52], Brumer-Shapiro coherent control [11], stimulated-Raman-Adiabatic-Passage (STIRAP) [23, 16, 8], to genetic algorithms [14]. Section 5.1 covers Quantum Optimal Control (QOC), one of the best

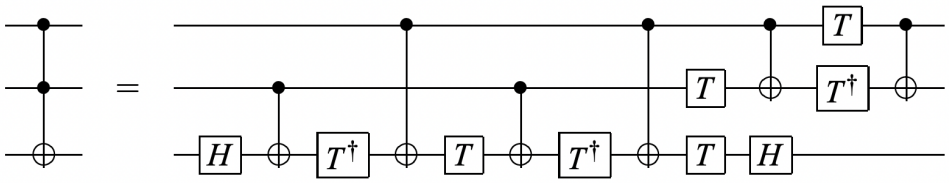

Figure 3: Decomposition of Toffoli Gate

ways to directly compile quantum operators into control pulses [25]. Examples of QOC include GRadient Descent Pulse Engineering (GRAPE) [19] and Krotov algorithms [46]. Section 5.2, covers the latest improvements on NISQ pulses still have room for refinement.

5.1 Quantum Optimal Control (QOC)

QOC groups gates together, where each group can be represented by a single unitary matrix U. If the group is $\{G_1, G_2, ..., G_n\}$, then $U = \prod_i^n G_i$. Thus, U is independent of the group size. Hence, the advantage of QOC is that it does not scale with the number of gates as much as gate based compilation. An optimal control is a set of differential equations describing the paths of the control variables that minimize the cost function. The cost function must be a function of the state and control variables that can be optimised with analytical or numerical methods [12]. Thus, QOC is clearly very computationally expensive.

GRadient Descent Pulse Engineering (GRAPE) was initially introduced for NMR spectroscopy in 2005 [35]. It assumes that the control sequence discretization step is small so that each step the algorithm moves in the direction of increasing performance. This ensures convergence of control amplitudes to extremal points of the desired performance function. This is an improvement over conventional numerical difference methods. For example, if there are $m \times N$ control amplitufes and $N = 500$ and $m = 4$, the conventional approach requires 2001 full time evaluations in comparison to just 2 for GRAPE. Despite this improvement, GRAPE is still extremely slow. For more than 5 qubits, GRAPE can take several hours to compile the pulses [12]. Though it does achieve speedup of $2 - 5\times$ gate-based compilation for a range of quantum algorithms [62].

Leung *et al.* attempted to reduce the compilation time with automatic GPU differentiation [47]. However, the acceleration is only noticeable for more than 10 qubits (an order of magnitude), and for less than 10 qubits it is minimal [47]. Even so, this compilation would still take weeks or months for useful quantum circuits in the order of 10^{10} gates, making GRAPE untenable for the future.

The work of [27] utilises partial compilation to achieve the benefits of GRAPE but with reasonable classical computational time. They propose two strategies for partial compilation:

1. **Strict partial compilation**. Pre-compute optimal pulses for parametrization-independent blocks of gates

2. **Flexible partial compilation**. Use precomputed hyperparameter optimization to dramatically speedup full GRAPE

The results show pulse speedups of 1.5x-3x over gate decomposition for typical benchmarks, with 10x-100x reduction in compilation latency in comparison to full GRAPE. Notwithstanding, this optimised GRAPE is still not practical for circuits with thousands of qubits and gates in the order of 10^{10}.

Another approach also attempts to mitigate GRAPE's long compilation time using minimum spanning trees (MST). This achieves a 9.88× compilation speedup compared to the standard compilation while reducing latency by 2.43× compared with gate-based compilation [12]. This balances latency reduction and compilation time. They precompute for common gate groups by randomly profiling a set of programs. Thus, they only use GRAPE if the group is not found in their precomputed set. The key insight is that pulses of groups can be generated faster if there is an existing pulse of a similar group to start optimisation. However, there is still room to improve the latency reduction, as the brute force improvement is 3.01×. Again, still too classically computationally expensive.

While QOC has showed promise theoretically, testing has shown noisy experimental systems are not ready for QOC methods [28]. On top of long compilation time, QOC depends on accurate models of the analog pulses and the device Hamiltonian. Pulses built of an inaccurate Hamiltoian accumulate significant error. The analog pulses have sampling errors from the waveform generator [51] and phase offset errors from the temperature difference between and the cold qubits and classical electronics [3], [45]. Furthermore, Hamiltonians are difficult to determine experimentally and change between daily recalibrations. Finally, QOC also needs evaluation of partial derivatives of a fidelity metric, easy analytically but extremely difficult with noisy architecture.

5.2 NISQ Pulse Compilation

More recently, there has been focus on improving NISQ systems by taking advantage of the daily calibrations that are already performed [21, 28]. Thereby bypassing the above issues. Optimisation with open pulse results in a mean 1.6× error reduction and 2× speedup for near-term algorithms [28]. These pulses are simpler and reduce control error. Their most notable optimisation involves approximating direct rotations. Qiskit decomposes an arbitrary single qubit gate, U_3, as follows

$$U_3 = \begin{bmatrix} \cos\left(\frac{\theta}{2}\right) & -e^{i\lambda}\sin\left(\frac{\theta}{2}\right) \\ -e^{i\phi}\sin\left(\frac{\theta}{2}\right) & e^{i(\phi+\lambda)}\cos\left(\frac{\theta}{2}\right) \end{bmatrix} \tag{5.2}$$

$$U_3(\theta,\phi,\lambda) = R_z(\phi+\frac{\pi}{2})R_x(\frac{\pi}{2})R_z(\theta+\pi)R_x(\frac{\pi}{2})R_z(\lambda) \tag{5.3}$$

In Qiskit, Z axis rotations can be implemented in software at "no cost" (by a compiler transformation on all future gates involving the target qubit). So, instead of relying on this, [28] use the pre-calibrated X gate to reduce the time for an X gate from 71.1 ns to 35.6 ns in Qiskit. These two pulses are equivalent as they have the same area under the curve, but the direct implementation is 2× as fast. Partial rotations can be achieved by scaling the amplitude by $\frac{\theta}{\pi}$ to rotate the qubit by θ radians [28]. Now qubits can be implemented with one X pulse instead of two

$$U_3(\theta,\phi,\lambda) = R_z(\phi+\pi)R_x(\theta)R_z(\lambda-\pi) \tag{5.4}$$

The fidelity improvements in [28] can be attributed to three sources:

1. Shorter Pulses. Reduce error as there is less time for the qubits to decohere.
2. Less calibration error susceptibility. Standard Qiskit decomposition applies two pre-calibrated pulses, squaring the impact of imperfect imperfections.
3. Smaller pulse amplitudes. Vertically downscaling amplitudes and horizontally stretching results in smaller pulse amplitudes that have smaller spectral components, reducing leakage to undesired frequency sidebands.

They also optimise the CNOT in a similar way. Qiskit uses two cross resonance pulses $CR(\pm\frac{\pi}{4})$, and can instead be implemented with one $CR(\frac{\pi}{2})$ pulse to achieve 24% speed up for common operations [28].

The above fidelity improvements in [28] were verified by a random benchmarking approach outlined by [44]. They select $K-1$ random single-qubit unitary operations. These are executed with one final single-qubit operation which is the inverse of all preceding operations. In a system without noise, the qubit should return to the initial state of $|0\rangle$ with 100% probability, but as K increases from 2 to 25 the noise error increases. For each K, they randomized 5 sequences of unitary operation for a total of $5 \times 24 \times 3 \times 8k = 2.88M$. They resulting gate fidelities for optimized, optimized-slow, and standard are 99.87%, 99.83%, and 99.82% respectively. This implied that shorter pulses account for 70% of the fidelity improvement, while less susceptibility to calibration imperfection and smaller pulse amplitudes account for the remaining 30% improvement [28]. This a great method to verify results while reducing state preparation and measurement error [44].

Gokhale *et al.* only focus on near-term algorithms [28]. They discuss the effect of changing the amplitude but not duration of the pulse, beta (the correction amplitude), or sigma (how wide or narrow the Gaussian peak is). One could also consider what are the trade-offs between amplitude and leakage to undesirable sidebands versus shorter pulses and less decoherence time.

Similarly, scaling cross-resonance pulses and exposing each pulse as a gate to remove redundant single-qubit operations with the transpiler results in 50% error reduction in the fidelity of $R_{ZZ}(\theta)$ and arbitrary $SU(4)$ gates [21].

Pre-calibrated gates were again used to optimise SWAP gates, another very common operation [29]. Due to the sparse connectivity of qubits, like in Figure 2, SWAP gates are needed so that an operation can be performed on any two chosen qubits. Consequently, SWAPs make up an overwhelming majority of operations, with this percentage approaching 100% as the number of qubits increase, since the average pairwise distance between qubits increases [29]. They achieve this through four ways:

1. **SWAP Orientation**. In implementation there are two gates CNOT and NOTC, the difference being the orientation. A SWAP can be decomposed into CNOT-NOTC-CNOT or NOTC-CNOT-NOTC. While assembly languages don't distinguish between the two, the CNOT requires 3 pulses, while the NOTC requires 5 pulses. So it is always faster to implement a SWAP via CNOT-NOTC-CNOT. Speedup depends on the architecture, for IBMQ's Johannesburg (20 qubits, 23 connected pairs) and Paris (27 qubits, 28 connected pairs) devices, the mean speedups are 2.5% and 3.2% respectively. This option also requires fewer single qubit rotations.

2. **Cross-Gate Pulse Cancellation**. The circuit $R_y(-90)$ $R_z(-90)$ $R_y(180)$ can be simplified to a $R_z(90)$ $R_x(-90)$, rotating the qubit by 90 degrees instead of 270.

3. **Commutation through Cross-Resonance**. The bottom qubit of the Cross-Resonance interaction is only affected by R_x gates. The R_x gates commute, so we can move the R_x gates on one side of a CR's target to the opposite side. Now we can compress the active rotation from 360 degrees to 90 degrees.

4. **Cross-Resonance Polarity**. Echoed CR native gate does not necessarily need to be ordered. IBM implements positive half-CR then negative halfCR, but switching the direction results in some positive cancellation opportunities to cancel the first $R_y(180)$ gate, reducing the active rotation by 180 degrees.

The lower the active rotation, the less error there should be. The important advantage that it requires no extra calibration so it can be executed without any extra calibration and will not hamper availability time.

They used Interleaved Randomized Benchmarking (IRB) [48] to measure the fidelity of Optimized and Standard SWAPs on each pair of connected qubits. The error of the Optimized SWAP is 3.3%, versus 3.7% for the Standard SWAP. An average reduction of 13% SWAP error. To emphasize the improvements at the application level they show the reduction improvement in success probability of 10-40% for four algorithms

1. Bernstein-Vazirani which aims to detect a 'secret' all-ones bitstring [9]
2. Generalized Toffoli [26] which computes the AND of input qubits
3. Quantum Adder [1] which computes $|11\rangle + |11\rangle = |110\rangle$
4. Long SWAP which chains SWAPs to move a source qubit to a distant target, i.e. $|100...000\rangle \rightarrow |000...001\rangle$.

References

[1] Weightedadder, 2022. https://qiskit.org/documentation/stubs/qiskit.circuit.library.WeightedAdder.html.

[2] Adetokunbo Adedoyin, John Ambrosiano, Petr Anisimov, Andreas Bärtschi, William Casper, Gopinath Chennupati, Carleton Coffrin, Hristo Djidjev, David Gunter, Satish Karra, et al. Quantum algorithm implementations for beginners. *arXiv preprint arXiv:1804.03719*, 2018.

[3] Thomas Alexander, Naoki Kanazawa, Daniel J Egger, Lauren Capelluto, Christopher J Wood, Ali Javadi-Abhari, and David C McKay. Qiskit pulse: programming quantum computers through the cloud with pulses. *Quantum Science and Technology*, 5(4):044006, 2020.

[4] 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.

[5] Frank Arute, Kunal Arya, Ryan Babbush, Dave Bacon, Joseph C Bardin, Rami Barends, Rupak Biswas, Sergio Boixo, Fernando GSL Brandao, David A Buell, et al. Quantum supremacy using a programmable superconducting processor. *Nature*, 574(7779):505–510, 2019.

[6] Adriano Barenco, Charles H Bennett, Richard Cleve, David P DiVincenzo, Norman Margolus, Peter Shor, Tycho Sleator, John A Smolin, and Harald Weinfurter. Elementary gates for quantum computation. *Physical review A*, 52(5):3457, 1995.

[7] Lindsay Bassman, Sahil Gulania, Connor Powers, Rongpeng Li, Thomas Linker, Kuang Liu, TK Satish Kumar, Rajiv K Kalia, Aiichiro Nakano, and Priya Vashishta. Domain-specific compilers for dynamic simulations of quantum materials on quantum computers. *Quantum Science and Technology*, 6(1):014007, 2020.

[8] K Bergmann, H Theuer, and BW Shore. Coherent population transfer among quantum states of atoms and molecules. *Reviews of Modern Physics*, 70(3):1003, 1998.

[9] Ethan Bernstein and Umesh Vazirani. Quantum complexity theory. *SIAM Journal on computing*, 26(5):1411–1473, 1997.

[10] Alex Bocharov and Krysta M Svore. Resource-optimal single-qubit quantum circuits. *Physical Review Letters*, 109(19):190501, 2012.

[11] Paul Brumer and Moshe Shapiro. Control of unimolecular reactions using coherent light. *Chemical physics letters*, 126(6):541–546, 1986.

[12] Jinglei Cheng, Haoqing Deng, and Xuehai Qia. Accqoc: Accelerating quantum optimal control based pulse generation. In *2020 ACM/IEEE 47th Annual International Symposium on Computer Architecture (ISCA)*, pages 543–555. IEEE, 2020.

[13] Jerry M Chow, Antonio D Córcoles, Jay M Gambetta, Chad Rigetti, Blake R Johnson, John A Smolin, Jim R Rozen, George A Keefe, Mary B Rothwell, Mark B Ketchen, et al. Simple all-microwave entangling gate for fixed-frequency superconducting qubits. *Physical review letters*, 107(8):080502, 2011.

[14] Xi Chu and Shih-I Chu. Optimization of high-order harmonic generation by genetic algorithm and wavelet time-frequency analysis of quantum dipole emission. *Physical Review A*, 64(2):021403, 2001.

[15] IBM Quantum Computing. Ibm quantum services - ibmq_quito, 2022. https://quantum-computing.ibm.com/services?services=systems&search=quito&system=ibmq_quito.

[16] George W Coulston and Klaas Bergmann. Population transfer by stimulated raman scattering with delayed pulses: analytical results for multilevel systems. *The Journal of chemical physics*, 96(5):3467–3475, 1992.

[17] 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. IEEE, 2020.

[18] Christopher M Dawson and Michael A Nielsen. The solovay-kitaev algorithm. *arXiv preprint quant-ph/0505030*, 2005.

[19] P De Fouquieres, SG Schirmer, SJ Glaser, and Ilya Kuprov. Second order gradient ascent pulse engineering. *Journal of Magnetic Resonance*, 212(2):412–417, 2011.

[20] Simon J Devitt, William J Munro, and Kae Nemoto. Quantum error correction for beginners. *Reports on Progress in Physics*, 76(7):076001, 2013.

[21] Nathan Earnest, Caroline Tornow, and Daniel J Egger. Pulse-efficient circuit transpilation for quantum applications on cross-resonance-based hardware. *Physical Review Research*, 3(4):043088, 2021.

[22] Austin G Fowler. Constructing arbitrary steane code single logical qubit fault-tolerant gates. *arXiv preprint quant-ph/0411206*, 2004.

[23] U Gaubatz, P Rudecki, S Schiemann, and K Bergmann. Population transfer between molecular vibrational levels by stimulated raman scattering with partially overlapping laser fields. a new concept and experimental results. *The Journal of Chemical Physics*, 92(9):5363–5376, 1990.

[24] Brett Giles and Peter Selinger. Exact synthesis of multiqubit clifford+ t circuits. *Physical Review A*, 87(3):032332, 2013.

[25] Steffen J Glaser, Ugo Boscain, Tommaso Calarco, Christiane P Koch, Walter Köckenberger, Ronnie Kosloff, Ilya Kuprov, Burkhard Luy, Sophie Schirmer, Thomas Schulte-Herbrüggen, et al. Training schrödinger's cat: quantum optimal control. *The European Physical Journal D*, 69(12):1–24, 2015.

[26] Pranav Gokhale, Jonathan M Baker, Casey Duckering, Natalie C Brown, Kenneth R Brown, and Frederic T Chong. Asymptotic improvements to quantum circuits via qutrits. In *Proceedings of the 46th International Symposium on Computer Architecture*, pages 554–566, 2019.

[27] Pranav Gokhale, Yongshan Ding, Thomas Propson, Christopher Winkler, Nelson Leung, Yunong Shi, David I Schuster, Henry Hoffmann, and Frederic T Chong. Partial compilation of variational algorithms for noisy intermediate-scale quantum machines. In *Proceedings of the 52nd Annual IEEE/ACM International Symposium on Microarchitecture*, pages 266–278, 2019.

[28] Pranav Gokhale, Ali Javadi-Abhari, Nathan Earnest, Yunong Shi, and Frederic T Chong. Optimized quantum compilation for near-term algorithms with openpulse. In *2020 53rd Annual IEEE/ACM International Symposium on Microarchitecture (MICRO)*, pages 186–200. IEEE, 2020.

[29] Pranav Gokhale, Teague Tomesh, Martin Suchara, and Frederic T Chong. Faster and more reliable quantum swaps via native gates. *arXiv preprint arXiv:2109.13199*, 2021.

[30] David Gosset, Vadym Kliuchnikov, Michele Mosca, and Vincent Russo. An algorithm for the t-count. *arXiv preprint arXiv:1308.4134*, 2013.

[31] Aram W Harrow, Avinatan Hassidim, and Seth Lloyd. Quantum algorithm for linear systems of equations. *Physical review letters*, 103(15):150502, 2009.

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

[33] W Jakubetz, B Just, J Manz, and HJ Schreier. Mechanism of state-selective vibrational excitation by an infrared picosecond laser pulse studied by two techniques: fast fourier transform propagation of a molecular wave packet and analysis of the corresponding vibrational transitions. *Journal of Physical Chemistry*, 94(6):2294–2300, 1990.

[34] Richard Jozsa. Searching in grover's algorithm. *arXiv preprint quant-ph/9901021*, 1999.

[35] Navin Khaneja, Timo Reiss, Cindie Kehlet, Thomas Schulte-Herbrüggen, and Steffen J Glaser. Optimal control of coupled spin dynamics: design of nmr pulse sequences by gradient ascent algorithms. *Journal of magnetic resonance*, 172(2):296–305, 2005.

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

[37] Alexei Yu Kitaev, Alexander Shen, Mikhail N Vyalyi, and Mikhail N Vyalyi. *Classical and quantum computation*. Number 47. American Mathematical Soc., 2002.

[38] Morten Kjaergaard, Mollie E Schwartz, Jochen Braumüller, Philip Krantz, Joel I-J Wang, Simon Gustavsson, and William D Oliver. Superconducting qubits: Current state of play. *Annual Review of Condensed Matter Physics*, 11:369–395, 2020.

[39] Vadym Kliuchnikov, Alex Bocharov, and Krysta M Svore. Asymptotically optimal topological quantum compiling. *Physical review letters*, 112(14):140504, 2014.

[40] Vadym Kliuchnikov, Kristin Lauter, Romy Minko, Adam Paetznick, and Christophe Petit. Shorter quantum circuits. *arXiv preprint arXiv:2203.10064*, 2022.

[41] Vadym Kliuchnikov, Dmitri Maslov, and Michele Mosca. Fast and efficient exact synthesis of single qubit unitaries generated by clifford and t gates. *arXiv preprint arXiv:1206.5236*, 2012.

[42] Vadym Kliuchnikov, Dmitri Maslov, and Michele Mosca. Asymptotically optimal approximation of single qubit unitaries by clifford and t circuits using a constant number of ancillary qubits. *Physical review letters*, 110(19):190502, 2013.

[43] Vadym Kliuchnikov, Dmitri Maslov, and Michele Mosca. Practical approximation of single-qubit unitaries by single-qubit quantum clifford and t circuits. *IEEE Transactions on Computers*, 65(1):161–172, 2015.

[44] Emanuel Knill, Dietrich Leibfried, Rolf Reichle, Joe Britton, R Brad Blakestad, John D Jost, Chris Langer, Roee Ozeri, Signe Seidelin, and David J Wineland. Randomized benchmarking of quantum gates. *Physical Review A*, 77(1):012307, 2008.

[45] Sebastian Krinner, Simon Storz, Philipp Kurpiers, Paul Magnard, Johannes Heinsoo, Raphael Keller, Janis Luetolf, Christopher Eichler, and Andreas Wallraff. Engineering cryogenic setups for 100-qubit scale superconducting circuit systems. *EPJ Quantum Technology*, 6(1):2, 2019.

[46] Vadim Krotov. *Global methods in optimal control theory*, volume 195. CRC Press, 1995.

[47] Nelson Leung, Mohamed Abdelhafez, Jens Koch, and David Schuster. Speedup for quantum optimal control from automatic differentiation based on graphics processing units. *Physical Review A*, 95(4):042318, 2017.

[48] Easwar Magesan, Jay M Gambetta, Blake R Johnson, Colm A Ryan, Jerry M Chow, Seth T Merkel, Marcus P Da Silva, George A Keefe, Mary B Rothwell, Thomas A Ohki, et al. Efficient measurement of quantum gate error by interleaved randomized benchmarking. *Physical review letters*, 109(8):080505, 2012.

[49] Ken Matsumoto and Kazuyuki Amano. Representation of quantum circuits with clifford and $\pi/8$ gates. *arXiv preprint arXiv:0806.3834*, 2008.

[50] Jarrod R McClean, Jonathan Romero, Ryan Babbush, and Alán Aspuru-Guzik. The theory of variational hybrid quantum-classical algorithms. *New Journal of Physics*, 18(2):023023, 2016.

[51] David C McKay, Thomas Alexander, Luciano Bello, Michael J Biercuk, Lev Bishop, Jiayin Chen, Jerry M Chow, Antonio D Córcoles, Daniel Egger, Stefan Filipp, et al. Qiskit backend specifications for openqasm and openpulse experiments. *arXiv preprint arXiv:1809.03452*, 2018.

[52] R Mitrić, M Hartmann, J Pittner, and V Bonačić-Koutecký. New strategy for optimal control of femtosecond pump-dump processes applicable to systems of moderate complexity. *The European Physical Journal D-Atomic, Molecular, Optical and Plasma Physics*, 24(1):177–180, 2003.

[53] Gary J Mooney, Charles D Hill, and Lloyd CL Hollenberg. Cost-optimal single-qubit gate synthesis in the clifford hierarchy. *Quantum*, 5:396, 2021.

[54] Prakash Murali, Jonathan M Baker, Ali Javadi-Abhari, Frederic T Chong, and Margaret Martonosi. Noise-adaptive compiler mappings for noisy intermediate-scale quantum computers. In *Proceedings of the Twenty-Fourth International Conference on Architectural Support for Programming Languages and Operating Systems*, pages 1015–1029, 2019.

[55] Victor Namias. The fractional order fourier transform and its application to quantum mechanics. *IMA Journal of Applied Mathematics*, 25(3):241–265, 1980.

[56] Michael A. Nielsen and Isaac L. Chuang. *Quantum Computation and Quantum Information: 10th Anniversary Edition*. Cambridge University Press, USA, 10th edition, 2011.

[57] Michael JD Powell. A direct search optimization method that models the objective and constraint functions by linear interpolation. In *Advances in optimization and numerical analysis*, pages 51–67. Springer, 1994.

[58] John Preskill. Fault-tolerant quantum computation. In *Introduction to quantum computation and information*, pages 213–269. World Scientific, 1998.

[59] John Preskill. Quantum computing in the nisq era and beyond. *Quantum*, 2:79, 2018.

[60] Chad Rigetti, Jay M Gambetta, Stefano Poletto, Britton LT Plourde, Jerry M Chow, Antonio D Córcoles, John A Smolin, Seth T Merkel, Jim R Rozen, George A Keefe, et al. Superconducting qubit in a waveguide cavity with a coherence time approaching 0.1 ms. *Physical Review B*, 86(10):100506, 2012.

[61] Sarah Sheldon, Easwar Magesan, Jerry M Chow, and Jay M Gambetta. Procedure for systematically tuning up cross-talk in the cross-resonance gate. *Physical Review A*, 93(6):060302, 2016.

[62] Kaitlin N Smith, Gokul Subramanian Ravi, Thomas Alexander, Nicholas T Bronn, Andre Carvalho, Alba Cervera-Lierta, Frederic T Chong, Jerry M Chow, Michael Cubeddu, Akel Hashim, et al. Summary: Chicago quantum exchange (cqe) pulse-level quantum control workshop. *arXiv preprint arXiv:2202.13600*, 2022.

[63] Neereja Sundaresan, Isaac Lauer, Emily Pritchett, Easwar Magesan, Petar Jurcevic, and Jay M Gambetta. Reducing unitary and spectator errors in cross resonance with optimized rotary echoes. *PRX Quantum*, 1(2):020318, 2020.

[64] David J Tannor, Ronnie Kosloff, and Stuart A Rice. Coherent pulse sequence induced control of selectivity of reactions: Exact quantum mechanical calculations. *The Journal of chemical physics*, 85(10):5805–5820, 1986.

[65] David J Tannor and Stuart A Rice. Control of selectivity of chemical reaction via control of wave packet evolution. *The Journal of chemical physics*, 83(10):5013–5018, 1985.

[66] Vera von Burg, Guang Hao Low, Thomas Häner, Damian S Steiger, Markus Reiher, Martin Roetteler, and Matthias Troyer. Quantum computing enhanced computational catalysis. *Physical Review Research*, 3(3):033055, 2021.

[67] Nicolas Wittler, Federico Roy, Kevin Pack, Max Werninghaus, Anurag Saha Roy, Daniel J Egger, Stefan Filipp, Frank K Wilhelm, and Shai Machnes. Integrated tool set for control, calibration, and characterization of quantum devices applied to superconducting qubits. *Physical Review Applied*, 15(3):034080, 2021.

[68] Xinlan Zhou, Debbie W Leung, and Isaac L Chuang. Methodology for quantum logic gate construction. *Physical Review A*, 62(5):052316, 2000.