Noise Modelling, Simulation and Benchmarking for Near-term Quantum Computers

Konstantinos Georgopoulos

School of Computing
Newcastle University

This thesis is submitted for the degree of
Doctor of Philosophy

March 2022

I would like to dedicate this thesis to my loving partner, Catherine, who put up with me working long hours every day and stood by me throughout this endeavour, and my loving parents, George and Elisabeth, for the support and love throughout all the years of my life.

Αφιέρωμα στους γονείς μου, Γιώργο και Έτα, για την αγάπη, το ήθος και τη στήριξη σε όλα τα χρόνια της ζωής μου, είθε το παρών κείμενο να φέρει για πάντα τη μνήμη τους.

Declaration

Acknowledgements

I would like to acknowledge the EPSRC for funding this research, the CDT team for the support and my supervisors, Dr. Paolo Zuliani, for his invaluable help and guidance throughout this endeavour, and Dr. Clive Emary for his excellent feedback on my publications.

Abstract

The field of quantum computing is rapidly evolving and concentrates increasing attention from both academia and industry. With predictions of commercial quantum computing around the corner, this thesis approaches some of the major challenges of the field: the efficiency of quantum circuits, the study of quantum noise and benchmarking the behaviour of Noisy Intermediate-Scale Quantum (NISQ) devices. The first part of the thesis examines two circuit approaches for quantum walks: one consisting of controlled inversions and the other replacing them with rotations. The rotational approach nullifies the large amount of ancilla qubits required by the inverters implementation. The theoretical results concentrate around the comparison of the two architectures in terms of structure, benefits and detriments, as well as computational resources. It is proven that the inverters approach requires exponentially fewer gates than the rotations but almost half the number of qubits in the system. Experiments on a quantum computer show that small quantum walks evolve closer to the expectations, whereas larger circuits are severely affected by noise.

The second major part of the thesis is concentrated around quantum noise, an effect that dominates every aspect of near-term quantum computers. The research is concerned with the modelling of noise in NISQ devices. The focus is on three error groups that represent the main sources of noise during a computation and each source is modelled via a quantum channel. A noise model that combines all three noise channels is engineered and used to simulate the evolution of the quantum computer using a set of calibrated error rates. Various experiments show that the new model provides a better approximation of the quantum computer's behaviour than when compared to other noise models. Following this, a genetic algorithm optimises the parameters used by the new noise model, bringing its behaviour even closer to the quantum computer. A comparison between the pre- and post-optimisation parameters reveals how certain operations can be more or less erroneous than the hardware-calibrated parameters show.

Finally, this thesis presents a framework that utilises quantum algorithms, the above noise model and an ideal simulator to benchmark quantum computers. The benchmark metrics highlight the difference between the quantum computer evolution and the simulated noisy and ideal evolutions. This framework is then used for benchmarking three IBMQ devices. The use of diverse algorithms as benchmarks stresses the computers in different ways, highlighting their behaviour for different circuits. The complexity of each quantum circuit affects the efficiency of a quantum computer, with increasing circuit size resulting to worse performance. The results show that the proposed benchmarks provide sufficient and well-rounded information regarding the performance of each quantum computer.

Table of contents

List of figures

List of tables

Chapter 1

Introduction

1.1 Background on Quantum Mechanics and Quantum Computing

Quantum computing is a primarily interdisciplinary scientific area of research that combines quantum physics and computing. It exploits fundamental properties of quantum reality and quantum states, such as superposition and entanglement, in order to perform highly complex computations in an efficient manner. The devices that perform such computations are called quantum computers. The prime argument in favour of quantum computation is the ability to massively speed-up classical applications and solve computational problems that are currently infeasible.

The field of quantum computing was, in its essence, established in 1980 when Paul Benioff proposed a quantum mechanical model of the well-known Turing machine [1]. The first indication of the potential of such a machine that follows microrealistic behaviour was suggested by Richard Feynman and Yuri Manin [2]. In the following years, quantum computing was firmly established as an independent scientific area of research with major landmark theoretical applications, like Shor's algorithm for quantum factoring [3], Grover's algorithm for quantum search [4] and many more.

Today, quantum computing attracts major attention from both academia and industry. It is currently predicted that the first commercial quantum computers will become available (IBMQ roadmap[1]) within the next five years.

1.1.1 Quantum Mechanical Principles for Quantum Computation

As a starting point, this section offers an introduction to the basic principles of quantum mechanics and, by extension, quantum computation that are widely studied and utilised throughout the entirety of the present thesis.

[1]https://www.ibm.com/blogs/research/2021/02/quantum-development-roadmap/

The Principle of Superposition

One of the most fundamental properties within quantum mechanics and quantum information and computation is the *principle of quantum superposition*, which states that, if $|x\rangle$ and $|y\rangle$ are two states of a quantum system, then any arbitrary (superposed) state of the form $\alpha |x\rangle + \beta |y\rangle$ is also a possible state of the quantum system, where $\alpha, \beta \in \mathbb{C}$ and $|\alpha|^2 + |\beta|^2 = 1$.

Intuitively, according to the superposition principle, any two (or more) quantum states (that also need to be renormalised) can be added together resulting to another valid quantum state, much like waves in classical physics. Conversely, every quantum state can be represented as a sum of two or more other distinct quantum states. Mathematically, this property is reflected as a property of the solutions to Schrödinger's equation: since the Schrödinger equation is linear, any linear combination of its solutions can also be a solution.

The Principle of Entanglement

The second fundamental principle of quantum mechanics is *quantum entanglement*. In physics this phenomenon occurs when a group of particles is generated, interact, or share spatial proximity in a way such that the quantum state of each particle of the group cannot be described independently of the state of the others. Additionally, according to the entanglement principle, no matter how far apart those particles are located, their quantum states will maintain this special relationship. Measurements of the properties of entangled particles can, under certain conditions, be found to be perfectly correlated.

Intuitively, quantum entanglement entails that a state which adheres to this property cannot be written as a product of its component states. For reasons that are still not clear, entangled states play an undeniably fundamental role in quantum computation, differentiating it massively from classical computation and providing enormous computational power. A simple example of an entangled state bearing the above characteristics is the two-qubit state

$$|\psi\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle).$$

In this case, the state $|\psi\rangle$ cannot be given as the product of two single-qubit states. It is, therefore, impossible to separate the states of each single qubit from the global (two-qubit composite) state. In other words, if the composite two-qubit system is in this entangled state, it is impossible to attribute to either single-qubit system a definite pure state.

1.1.2 Quantum Hamiltonians and Hamiltonian Simulation

In quantum mechanics, the Hamiltonian of a system is a Hermitian operator that corresponds to the total energy of that system. The spectrum of a Hamiltonian, or in other words, the (energy) eigenvalues of the operator, corresponds to the set of possible observable outcomes that can be obtained when measuring the energy of the system. The Hamiltonian contains the operators associated with the kinetic and potential energies.

Considering a particle in one dimension as a quantum system, its Hamiltonian can be written as

$$\hat{H} = \hat{T} + \hat{V},$$

where $\hat{T}$ is the kinetic energy and $\hat{V}$ the potential energy operators of the system.

The Hamiltonian of a quantum system is used to describe the evolution of said system in time via the time-dependent Schrödinger equation as

$$\mathrm{i}\hbar \frac{\mathrm{d}|\psi\rangle}{\mathrm{d}t} = \hat{H} |\psi\rangle , \tag{1.1}$$

where $\hbar = h/2\pi$ and h is Planck's constant, i is the imaginary number (i.e., $\mathrm{i}^2 = -1$), $|\psi\rangle$ is an arbitrary quantum state and t is the time parameter. Defining the quantum evolution via Schrödinger's equation is widely used in quantum computing, especially when the evolution occurs in continuous-time, as it is the case for the continuous-time quantum walk that this thesis is concerned with extensively. The quantum state of a system with an initial state $|\psi_0\rangle$ can be found at arbitrary time t by solving Schrödinger's equation as

$$|\psi(t)\rangle = e^{-\mathrm{i}\hat{H}t/\hbar} |\psi_0\rangle . \tag{1.2}$$

The above formulation of the Hamiltonian and its role within Schrödinger's equation holds when the quantum system in question is fully isolated from its environment. In reality though, a fully isolated system is extremely hard to achieve, a fact that holds true within quantum computing (giving rise to the noise, discussed further in Chapter 2) and whose effects in the field sit at the heart of the thesis. Considering a quantum system that interacts with its environment (i.e., not fully isolated), after some time it will reach an equilibrium state (i.e., either the ground state or an exited state), also referred to as a Gibbs state. This state can be determined by a Hamiltonian encompassing the sum of all possible energies, E_i, in the quantum system as

$$\hat{H}_{\mathrm{G}} = \sum_i E_i |\psi_i\rangle \langle\psi_i| ,$$

where $|\psi\rangle$ is a quantum state and $i \in \mathbb{N}$ represents the energy dimensionality of the quantum system, and it is finite or countable.

Assuming a system with n interacting, d-dimensional particles (d finite), the Hamiltonian representing the energy of the entire quantum system can be defined as a sum of the form

$$\hat{H} = \sum_{j=1}^{n} \hat{H}_j. \tag{1.3}$$

Intuitively, the above equation describes the Hamiltonian, $\hat{H}$, as the sum of all the Hamiltonians, $\hat{H}_j$, associated with n groups of locally interacting particles within the system. Thus, the Hamiltonians $\hat{H}_j$ are called *local Hamiltonians* and each one of them acts on a group of at most $k = \mathcal{O}(1)$ particles (k-local Hamiltonian). Here, the notation $\mathcal{O}(g(n))$ is a mathematical notation that describes the limiting behavior of a function when the argument tends towards a particular value or infinity. In computational complexity terms, it defines the rate of change of the execution efficiency of an algorithm (i.e., usually the time or steps needed) given a function g on the problem (input) size n.

The Hamiltonian Simulation Problem

One of the most prominent fields in quantum computation and quantum information is *Hamiltonian simulation*. It essentially attempts to discover efficient ways to decompose the Hamiltonian of a quantum evolution into a sequence of quantum gates that are easily implementable (or, at least, simpler than the Hamiltonian itself) on a quantum computer, associated with a specified error margin. This target comprises what is known as the *Hamiltonian simulation problem* (HSP). Definition 1.1 provides a more concrete description of the HSP.

Definition 1.1 (Hamiltonian simulation problem (HSP))**.** *Given the Hamiltonian, $\hat{H}$, of a quantum system, a time, t, and an error tolerance, ϵ, find a quantum circuit that performs the unitary operation $e^{-\mathrm{i}\hat{H}t}$ on an unknown quantum state with error at most ϵ.*

Bounding the solutions to the HSP at a threshold ϵ essentially defines an acceptable approximation between an *analogue* Hamiltonian evolution and the *digital* circuit that simulates it. Consequently, the error ϵ also defines the complexity of the Hamiltonian simulation algorithms that aim to solve the HSP.

There is a large number of applications where an efficient solution to the Hamiltonian simulation problem would prove invaluable. Various examples include, but are not limited to: (i) quantum simulations within quantum physics, quantum chemistry, material sciences, medicine, and more; (ii) adiabatic optimisation (iii) quantum algorithms, especially in continuous-time, i.e., continuous-time quantum walks, solving linear equations, and more.

Lie-product Formula for Hamiltonian Decomposition

An efficient and well studied approach to the HSP is to simulate a decomposition of the Hamiltonian of a quantum system to a sum of terms. Consider a Hamiltonian $\hat{H}$ that can be decomposed into a sum of m terms of the form

$$\hat{H} = \sum_{j=1}^{m} \hat{H}_j.$$

This equation is the same as equation (1.3) for the local Hamiltonians, giving an easy and intuitive guide to how such a sum of terms could be comprised through a local Hamiltonian decomposition.

Following the decomposition of the Hamiltonian, the next step is to combine the individual simulations of the decomposed terms such that they comprise the total simulated system. One way to do this, and the one preferred throughout this thesis, is achieved by using the Lie-product formula (also known as the Trotter product formula [5, 6]):

$$e^{-\mathrm{i}\hat{H}t} = e^{-\mathrm{i}\left(\hat{H}_1+\cdots+\hat{H}_m\right)t} = \lim_{r\to\infty}\left(e^{-\mathrm{i}\hat{H}_1 t/r}e^{-\mathrm{i}\hat{H}_2 t/r}\dots e^{-\mathrm{i}\hat{H}_m t/r}\right)^r,$$

for arbitrary time, t. In this equation, r represents the number of times that the product needs to be repeated, or in computational terms, the number of iterations of the product that are needed to get the initial $e^{-\mathrm{i}\hat{H}t}$. Taking an infinite number of iterations of the decomposition to achieve exact equality between the Hamiltonian exponential and its decomposition is computationally infeasible. Hence, the efforts are focused in approximating the initial Hamiltonian and the above equation is rewritten as:

$$e^{-\mathrm{i}\hat{H}t} = e^{-\mathrm{i}\left(\hat{H}_1+\cdots+\hat{H}_m\right)t} \approx \left(e^{-\mathrm{i}\hat{H}_1 t/r} e^{-\mathrm{i}\hat{H}_2 t/r} \dots e^{-\mathrm{i}\hat{H}_m t/r}\right)^r . \tag{1.4}$$

Similarly with the limit, as r tends to infinity, the decomposition gets arbitrarily closer to the Hamiltonian exponential, or otherwise, the error of the simulation tends towards zero. The next step is to identify a sufficiently large r so that the approximation is adequately close to the original Hamiltonian exponential. To ensure the error is bounded on a satisfactory low value, ϵ, it has been proven by [7] that it suffices to take

$$r = \left(\left\|\hat{H}\right\|_2 t\right)^2 / \epsilon$$

iterations of the product, where $\left\|\hat{H}\right\|_2$ is the 2-norm of the Hamiltonian.

1.1.3 Quantum Machine Learning

Quantum machine learning is an interdisciplinary field of quantum information that lies at the intersection of machine learning, statistics and quantum physics. It utilises the power of quantum computing to provide immense computational speed-ups over classical machine learning algorithms. In more technical terms, quantum machine learning techniques utilise algorithms that exhibit proven quantum advantage in order to engineer machine learning routines that are more efficient than their classical analogues when running on a quantum computer. This includes hybrid methods that involve both classical and quantum processing, where computationally difficult subroutines are outsourced to a quantum device.

Quantum machine learning is attracting significant attention from both academia and industry as a result of the explosion of the field of quantum computing within the recent years. Researchers and quantum computing providers are beginning to view quantum machine learning techniques as a major source of quantum computational advantage. The reason for that is the significant speedups offered by quantised algorithms used in machine learning.

- *Quantum walks*: offering quadratic speedup over classical random walks [8] and exponential speedups via certain black-box applications [9]. This algorithm is essential to the field of quantum machine learning, acting as the base upon which a huge number of machine learning applications is built (i.e. Markov chain Monte Carlo methods [10], structured or unstructured search [11, 12] and many more) [13, 14].

- *Quantum metropolis sampling*: the quantised version of one of the most used machine learning techniques out there with significant quantum advantage (quadratic or exponential, coming from quantum walks) [15, 16].
- *Quantum linear algebra*: offering exponential speedups over classical linear algebra techniques used within machine learning (i.e. solving linear equations, working with higher-dimensional matrices, and much more) [17].
- *Quantum search*: either based on Grover's algorithm [4] or on quantum walks [12, 11], searching databases mapped to quantum mechanical states is proven to be, at least, quadratically faster than in the classical case (exponential techniques based on black-box isolation need an ideal quantum environment; thus, they are often taken as purely theoretical).
- *Quantum deep learning*: the birth of a number of machine learning and deep learning platforms, like TensorFlow Quantum[2], has given a huge leap to deep learning applications and large amounts of funding poured into them by both academia and industry.

1.2 Quantum Algorithms

This section introduces the four quantum algorithms that will be used within the thesis: (i) discrete-time quantum walks (DTQW or simply QW), (ii) continuous-time quantum walks (CTQW) (iii) quantum phase estimation (QPE), and (iv) quantum search (QS) or otherwise, Grover's algorithm. It is noteworthy here that Markov chain Monte Carlo methods are usually applied over a continuous space. Within this thesis, both DTQW and CTQW are considered to occur only in discrete space.

1.2.1 One-dimensional Discrete-time Quantum Walks

This thesis often uses a discrete-time quantum walk (DTQW) on a finite cycle with N states and an arbitrary number of steps [18]. An example of such a cycle for $N = 8$ states is shown in Figure 1.1. For simplicity during the remainder of the manuscript, a quantum walk (either discrete or continuous) is assumed to occur on a one-dimensional graph. In this construct, the evolution of the walker is guided by a balanced quantum coin in superposition. The fact that the quantum coin is balanced means that the walker takes every path on the graph with the same probability. Imagine a quantum particle that moves freely between adjacent discrete points on a line, then at each time step, a balanced quantum coin is flipped and the quantum state undergoes a unitary transformation, otherwise called "shift". Then the particle progresses according to the state of the quantum coin, thus evolving the walk. Figure 1.2 shows the dynamics of the quantum system during a step of the quantum walk.

[2]For more on TensorFlow Quantum: https://www.tensorflow.org/quantum.

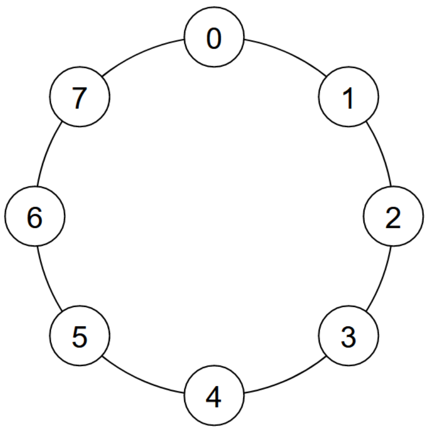


Fig. 1.1 A cycle graph with 8 states (or nodes), or otherwise, an 8-cycle.

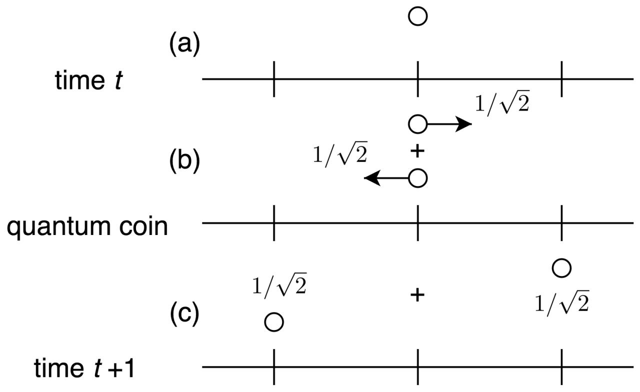


Fig. 1.2 Dynamics of the balanced quantum walk. (a) The walk begins at time t. (b) The flip of the quantum coin, where the particle is in equal superposition to go left or right. (c) The particle moves to generate the superposed state at time $t+1$.

This process can be described by the repeated application of a unitary evolution operator, U_{qw}. This operator acts on a Hilbert space $\mathcal{H}^C \otimes \mathcal{H}^S$, where $\mathcal{H}^C$ is associated with a quantum coin and $\mathcal{H}^S$ with the state space (positions, nodes on the graph) of the walk. In order to describe the quantum walk define the unitary operator, U_{qw}, as

$$U_{\text{qw}} = S \cdot (C \otimes I) \tag{1.5}$$

where S is the shift operator describing the walker's propagation, C is the quantum coin operator that guides the quantum evolution and I is the identity. The quantum coin is implemented via the Hadamard operator with well-known matrix representation

$$C = H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}. \tag{1.6}$$

This type of quantum coin is often called a *Hadamard coin*, and the quantum walk that uses a Hadamard coin is often referred to as *Hadamard quantum walk*.

Considering a walker that moves on a one-dimensional line or graph, after each flip of the coin, the said walker can either *increase* or *decrease* its position by a step of 1. This is defined by the shift operator S and can be described via increment and decrement functions, as demonstrated by [19]. The mathematical description of the shift operator is

$$S = S^{-} \otimes |0\rangle \langle 0| + S^{+} \otimes |1\rangle \langle 1| , \tag{1.7}$$

where $S^{+} : |x\rangle \to |x+1\rangle$ moves the walker one step to the right, increasing its position, and $S^{-} : |x\rangle \to |x-1\rangle$ to the left, decreasing its position. On a cycle graph with N possible states (i.e., N nodes on the graph) the walker moves from position $|0\rangle$ to $|N-1\rangle$ upon decrement on the former state, with the opposite being true upon increment on the latter.

After describing the dynamics and defining the quantum walk as a unitary quantum evolution, it is essential to discuss the three intrinsic characteristics that lead to the massive differentiation between discrete-time and space quantum walks and their classical counterpart: (i) asymmetry, (ii) modularity and (iii) quadratic efficiency.

Asymmetry in Discrete-time and Space Quantum Walks

One important characteristic of a discrete-time quantum walk on a one-dimensional line or graph that uses a balanced Hadamard quantum coin is its *asymmetry* [20]. In general, the asymmetry is the result of the Hadamard coin introducing bias in the path selection. After the evolution of the walker's position, the probability of each state to be measured may not be the same. The reason for this phenomenon is quantum interference, which can be either constructive or destructive. This can affect the quantum walk for more than one iteration of the shift operator, S. Precisely, the leftwards path (S^{-}) interferes more destructively, whereas the rightwards (S^{+}) path undergoes more constructive interference.

Interference in quantum mechanics occurs mainly due to the mathematical properties of the amplitudes. To be more precise, the amplitudes are complex numbers and can, thus, be positive or negative. When the wave function (partially) collapses the probabilities can be calculated as the modulus squared of the amplitudes in the superposition. For example, encountering a superposition in the general form $\alpha |0\rangle + \beta |1\rangle$, where α, $\beta \in \mathbb{C}$ are the amplitudes, then the probabilities in the classical world are $|\alpha|^2$ and $|\beta|^2$, where of course $|\alpha|^2 + |\beta|^2 = 1$. Thus, it is easy to deduce that any negative or complex amplitude vanishes in the classical world (i.e., post-measurement) and the probabilities remain real numbers in $[0, 1]$. On the other hand, pre-measurement, while the laws of quantum mechanics are in control, negative amplitudes can potentially interfere with the qubit phase during the evolution of the quantum walk.

In order to enforce symmetry in quantum walks there exist two potential and relatively easy solutions. The first is an initialisation trick, where the particle starts in a balanced superposition of the form $\frac{1}{\sqrt{2}}(|0\rangle + \mathrm{i}\,|1\rangle)$. The second solution makes use of a different coin operator instead of the Hadamard coin, C. In this case the coin operator, say C_{sym}, will

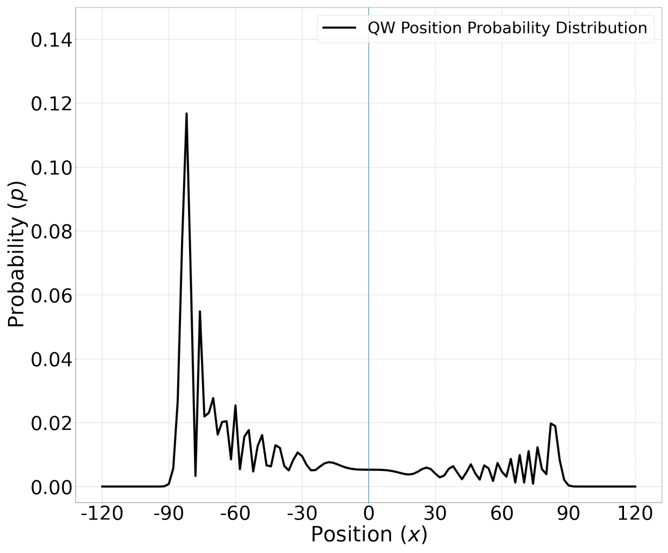


Fig. 1.3 Asymmetry in quantum walks on an infinite one-dimensional line. The walker evolves for 120 coin-flips and is initialised on state $|0\rangle$.

not bias the coin towards a certain base vector and can be constructed as

$$C_{\text{sym}} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & \mathrm{i} \\ \mathrm{i} & 1 \end{pmatrix} .$$

Figure 1.3 provides a visual representation of the asymmetry of a quantum walk evolution on an infinite, one-dimensional line. The walker is initialised on state $|0\rangle$ and is allowed to evolve for 120 steps.

Modular Behaviour of Discrete-time and Space Quantum Walks

A very important property that arises from the dynamics and behaviour of a discrete-time quantum walk is the *modularity property* [21].

Definition 1.2 (Modularity property)**.** *The modularity property (or modular behaviour) of a quantum walk expresses the relationship between the parity of the number of coin-flips of the quantum walk, the initial position of the walker and the measured states.*

According to the modularity property, as defined in Definition 1.2, if the walker is initialised on an even numbered position (including the $|0\rangle$ state) then, after an odd (even) number of steps the position of the particle will be a superposition of odd (even) states. Alternatively, if the walker is initialised on an odd position, after an odd (even) number of steps the particle will exist in a superposition of even (odd) states.

The modular behaviour of the quantum walk is best depicted through an example. Consider a quantum walk on a one-dimensional line where the walker is initialised on the even state $|2\rangle$. Letting the walker evolve for an odd number of steps (for example, one

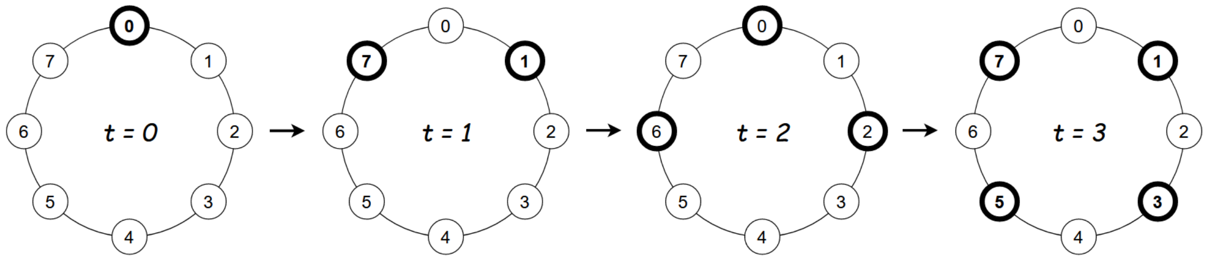


Fig. 1.4 The evolution of a quantum walk on a cycle graph with eight states. Due to the modular behaviour of the process, only specific nodes (highlighted) in every time step will appear with some probability.

coin-flip) will result in a superposition of odd states (for example, a superposition of the states $|3\rangle$ and $|1\rangle$). On the other hand, letting the walker evolve for an even number of steps (i.e., two coin-flips) means the resulting state will be a superposition of even states (i.e., states $|0\rangle$, $|2\rangle$ and $|4\rangle$). Another example of the modular behaviour for a quantum walk on a cycle with eight possible states and a different starting state ($|0\rangle$) is shown in Figure 1.4.

The modular behaviour is maintained strictly under the ideal evolution of the quantum walk, but is violated under the influence of noise. One more detail that makes this property very interesting is the fact that the level of violation of the modularity of the measured states reflects the level of noise that affects the quantum evolution. To be more specific, within the output distribution, the higher the probability of the states that should not appear according to the modularity property, the noisier the quantum evolution.

Quantum Advantage

The final characteristic of the discrete-time quantum walk that needs mentioning here is the quantum advantage compared to the classical random walk. It has been proven that quantum Markov chains, and by extension quantum walks, show near-quadratic increase in the variance with respect to time as opposed to their classical analogues [8]. Theoretically, the variance, σ^2_{qw}, as a function of the coin flips (or number of steps), t, can be calculated as [22]

$$\sigma^2_{\text{qw}} = \frac{\sqrt{2}-1}{2} t^2 \approx 0.2 \times t^2. \tag{1.8}$$

Surprisingly, at the date of this thesis and to the best of the author's knowledge, there is no documentation of a complete and easy to read analytical proof of equation (1.8). Thus, such a step-by-step proof is put together from the literature and provided in Section 3.1. Furthermore, this quadratic tendency of the quantum walk's propagation can be verified by computing the simulated quantum walk variance, with result as depicted in Figure 1.5.

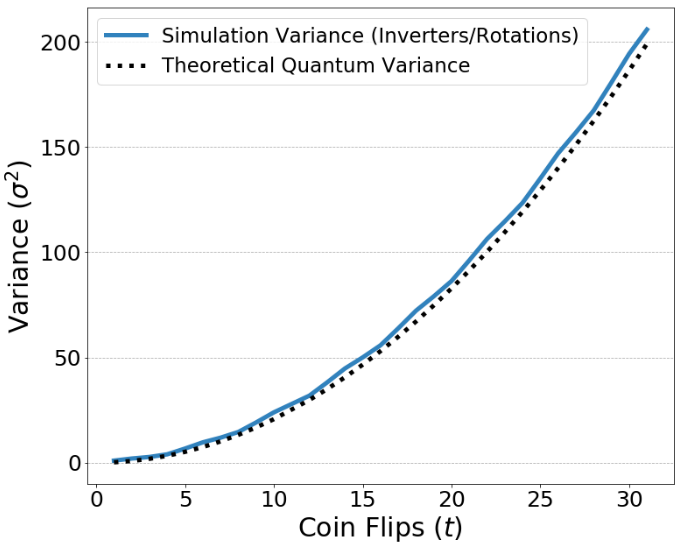


Fig. 1.5 Variance of the quantum walk as a function of the coin flips. Here the simulation variance (solid line) is the one computed on a $N = 256$-states cycle via simulations (for more information, refer to Section 3.2.3). The theoretical variance (dotted line) is calculated from equation (1.8).

1.2.2 One-dimensional Continuous-time Quantum Walks

Continuous-time quantum walks (CTQW) are a very interesting construct that share dynamics with the discrete-time quantum walk, but exhibit very different behaviour. They were first introduced by Farhi and Gutmann in [23]. Consider an undirected graph $G(V, E)$, where V is the set of its vertices and E the set of its edges. In classical mechanics, a diffusion equation can be used which essentially, considering a vertex j, describes a process which leaks probability to, or receives probability from, neighbouring vertices. The number of neighbouring nodes equals the degree of the node j, or $\deg(j)$. This diffusion operation can be expressed as

$$\frac{\mathrm{d}}{\mathrm{d}t} p_j(t) = \sum_{k \in V} L_{j,k} p_k(t) \tag{1.9}$$

where t is an arbitrary continuous time parameter, p_j is a function that describes the probabilistic exchange between node j and its neighbouring nodes and L is symmetric and Hermitian and is called the Laplacian of G, given by

$$L_{j,k} = \begin{cases} -\deg(j) & j = k \\ 1 & (j,k) \in E \\ 0 & \text{otherwise} \end{cases}$$

Thus, since L is a Hemitian matrix, it can play the role of a Hamiltonian in Schrödinger's equation as

$$\mathrm{i} \frac{\mathrm{d}}{\mathrm{d}t} |\psi(t)\rangle = L\, |\psi(t)\rangle \tag{1.10}$$

where, for simplicity it can be assumed that $\hbar = 1$ and $|\psi(t)\rangle$ is the state of the system at arbitrary time t. For more comprehensive information on the Laplacian, see Appendix A.

Equation (1.10) represents the quantised version of a continuous-time random walk. As evident from the above analysis, the CTQW does not need a quantum coin to drive the evolution, unlike a discrete-time quantum walk. A further analysis of the Hamiltonian of a continuous-time quantum walk can be found in Appendix B.

Continuous-time Quantum Walks on General Graphs

This part describes how a CTQW evolves on a general undirected graph. Instead of a complex Hamiltonian it is simpler to work with an adjacency matrix, A, which can be quite easily defined as

$$A_{j,k} = \begin{cases} 1 & \text{if } (j,k) \in E \\ 0 & \text{if } (j,k) \notin E \end{cases} \tag{1.11}$$

where $j, k \in V$.The adjacency matrix A is also Hermitian and thus it can form the Hamiltonian for the evolution of the quantum state. Denoting j and k as positions on the graph, Schrödinger's equation defines the evolution of the CTQW on the graph as

$$\mathrm{i}\frac{\mathrm{d}}{\mathrm{d}t} \langle j|\psi(t)\rangle = \sum_{j,k} \langle j|H|k\rangle \langle k|\psi(t)\rangle , \tag{1.12}$$

where $\langle j|\psi(t)\rangle$ is the amplitude of the quantum state being on vertex j at time t, $\hbar = 1$ for simplicity and $|\psi(t)\rangle$ is the quantum state vector within a position Hilbert space, $\mathcal{H}^S$. The Hamiltonian of the CTQW can be constructed via the adjacency matrix, A, from equation (1.11) as $H = \gamma A$ where γ is the hopping rate per edge per unit time (i.e., the probability to traverse from a node to its neighbour).

It is noteworthy here that for the position Hilbert space, $\mathcal{H}^S$, the same notation is used as for the DTQW case. In general, the size of the Hilbert space for the CTQW is not the same as for the DTQW due to the lack of quantum coin. More specifically, for the DTQW the Hilbert space is of the form $\mathcal{H}^S \otimes \mathcal{H}^C$, where $\mathcal{H}^C$ is the coin space, whereas for the CTQW the Hilbert space is simply $\mathcal{H}^S$, as seen above.

For the experiments within this thesis, the graphs are considered to be of a fixed degree. Continuous-time quantum walks, similarly to the discrete-time quantum walks, "seemingly" traverse all possible paths of the graph in superposition. Considering a Hamiltonian of the form $H = \gamma(A - dI)$ for graphs of a fixed degree d, equation (1.12) can be solved in the same manner as Schrödinger's equation as [24]

$$|\psi(t)\rangle = e^{-\mathrm{i}\gamma(A-dI)t} |\psi(0)\rangle ,$$

where $|\psi(0)\rangle$ is the initial quantum state (i.e., at $t = 0$). Since A commutes with the identity, I, the previous equation can be written as $|\psi(t)\rangle = e^{-\mathrm{i}\gamma At} e^{\mathrm{i}\gamma dIt} |\psi(0)\rangle$. The factor $e^{\mathrm{i}\gamma dIt}$ is a global phase irrelevant to the result that makes no difference to observable

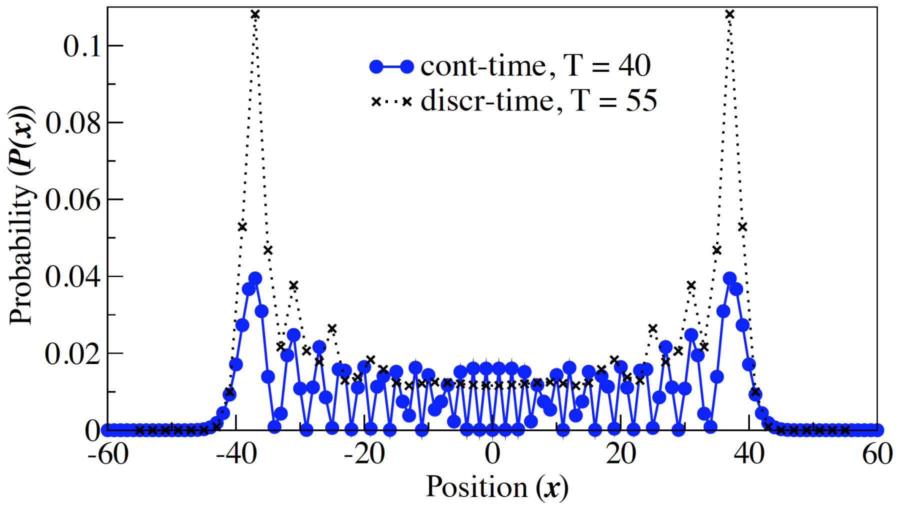


Fig. 1.6 Continuous-time (dots) versus discrete-time (crosses) quantum walk on a line after 40 and 55 steps respectively. The initial state of the quantum walk is $(-|1\rangle + i|1\rangle)/\sqrt{2}$ in order for the DTQW to be symmetric.

quantities and thus can be omitted, resulting in equation

$$|\psi(t)\rangle = e^{-\mathrm{i}\gamma At}|\psi(0)\rangle \tag{1.13}$$

for graphs of a fixed degree.

It is easy to spot the similarity between the evolution operator for the CTQW deriving from equation (1.13) and the one for the DTQW, defined in equation (1.5). Specifically,

$$|\psi(t)\rangle = U^t|\psi(0)\rangle\,,$$

where $U = S \cdot C$ in the discrete-time case and $U = e^{-\mathrm{i}\gamma A}$ in the continuous-time. This construction of an evolution unitary via exponentiation of the form $e^{-\mathrm{i}\gamma At}$, where $H = \gamma A$, is very common and is also used within this thesis for the implementation of continuous-time quantum walks, as shown in Section 6.2.4.

Continuous-time Quantum Walks on the Line

It is very easy to describe a continuous-time quantum walk on a one-dimensional line by using Bessel functions (see Appendix A). The solution of equation (1.12) in this case will take the form

$$|\psi(t)\rangle = \sum_{x=-\infty}^{+\infty} (-\mathrm{i})^x J_x(t)\,|x\rangle\,, \tag{1.14}$$

where J_x is the Bessel function of order $x \in \mathbb{Z}$ and $|\psi(t)\rangle$ is the state of the system at time t. An example of a continuous-time quantum walk on a line, as well as a discrete-time quantum walk for comparison, can be seen in Figure 1.6.

As can be seen from the above figure, an important characteristic of CTQW is that they do not exhibit modular behaviour. In other words, during the continuous-time quantum evolution the walker will always appear on all positions, regardless of their parity. This case does not stand for discrete-time, where the parity of the possible positions of a walker is dependent on the parity of the initial state and the number of coin-flips of the walk.

Additionally, much like discrete-time, continuous-time quantum walks show quadratic spread compared to classical random walks. For the CTQW starting at the origin, for asymptotically large times T, the variance, σ, can be found as [25, 23]

$$\sigma^2(T) = \frac{T^2}{\gamma}. \tag{1.15}$$

The continuous-time quantum walk on the line is always symmetric if the hopping rate γ is the same for both directions of the line. This is also an important difference with discrete-time quantum walks, where the coin induces bias on the direction, unless special choices are made for the initial state of the quantum walk, or a special operator is used as the quantum coin.

Continuous-time Quantum Walks on an N-cycle

In this case, the adjacency matrix A is a circulant matrix, i.e., a square matrix in which each row vector is rotated one element to the right relative to the preceding row vector. The eigenvalues of A can be expressed as $\lambda_x = 2\cos(2\pi x/N)$ and the corresponding eigenvectors $|b_x\rangle$, with $\langle y|b_x\rangle = \frac{1}{\sqrt{N}}e^{-2\pi \mathrm{i} x_y/N}$ for $x = 0, \ldots, N-1$. Taking as initial state $|\psi(0)\rangle = |0\rangle$, $|\psi(t)\rangle$ can be solved by decomposing $|0\rangle$ in terms of the eigenvectors $|b_x\rangle$, giving

$$|\psi_t\rangle = \frac{1}{\sqrt{N}}\sum_{x=0}^{N-1} e^{-\mathrm{i}t\lambda_x}\,|b_x\rangle\,. \tag{1.16}$$

The Hamiltonian for a continuous-time quantum walk on an N-cycle (used for the experiments in Chapter 6) can be formulated as $H = \gamma A = \frac{1}{d}A$, where $d = 2$ is the degree of every node in the graph (every node has exactly two neighbours). Thus, it is easy to derive that $H = \frac{1}{2}A$, making the construction of the evolution unitary $e^{-\mathrm{i}Ht}$ very straightforward.

As in the discrete-time case, the probability distribution of a CTQW on a cycle, $P(x,t)$, does not mix asymptotically and is known to have exact instantaneous mixing only for $N = 3$ and $N = 4$, as shown by [26]. The time-averaged probability distribution for the continuous-time quantum walk for large time interval T can be defined as

$$\overline{P(x,T)} = \frac{1}{T}\int_0^T P(x,t)\mathrm{d}t,$$

where $P(x,t) = |\langle x|\psi(t)\rangle|^2$ for arbitrary state x. This does indeed mix asymptotically, just as in the DTQW case, with a non-uniform limiting distribution. Finally, due to the lack of modular behaviour on the continuous-time case, the properties of CTQW on an N-cycle are not dependent on whether the cycle has an odd or even number of nodes, something that is not the case on the discrete-time quantum walk.

Continuous-time vs Discrete-time Quantum Walks

Up to this point, the dynamics of continuous-time quantum walk on some constructs of interest have been discussed. It is safe to conclude that the two quantum walks exhibit different behaviour, with the main points of comparison shown below.

- A CTQW evolves under a Hamiltonian defined with respect to the line/graph that the walk traverses. This difference is reflected on the evolution operator of a CTQW, $e^{-\mathrm{i}Ht}$, as compared to a DTQW, $(S \cdot C)^t$.
- CTQWs do not utilise a quantum coin. This leads to a Hilbert space of a different size than DTQWs. More specifically, the CTQW uses a position Hilbert space $\mathcal{H}^S$, whereas a DTQW uses a position and coin Hilbert space, $\mathcal{H}^S \otimes \mathcal{H}^C$.
- CTQWs are symmetric, provided that the hopping rate γ is the same for every direction on the line or graph. In contradiction, DTQWs exhibit directional bias introduced by the quantum coin. This bias can be eliminated by using a specific coin operator, or by initialising the walk on a specific state.
- CTQWs do not exhibit modular behaviour, as this was defined in Definition 1.2. All states appear after measurement when a continuous-time quantum walk evolves for time t. This is a substantial difference between discrete- and continuous-time quantum walks, where in the former, there is a strict relation between the parity of the observable states, the parity of the initial state and the number of coin flips.
- Both CTQWs and DTQWs exhibit quadratic speedup of the variance. A walker undergoing quantum walk will spread quadratically faster than in the case of a classical random walk both in continuous and discrete time [27, 28].

Finally, differences aside, it is indisputable that continuous- and discrete-time share a deep connection. This can be further realised through the work of Childs [29], which shows a way of simulating continuous-time quantum walks by using discrete-time quantum walks.

1.2.3 Quantum Search

Within this research, quantum search, also known as Grover's algorithm, refers to a quantum algorithm for unstructured search that finds with high probability a unique, marked item within a simple dataset. Suppose a search on an unstructured array of N items. A classical algorithm needs, at the worst case, $\mathcal{O}(N)$ operations to perform the search, whereas Grover's quantum search requires $\mathcal{O}(\sqrt{N})$ operations [4], speeding up the task quadratically.

Let x denote the items within the array and w the marked item (i.e., the item the routine is searching for). Also let f be a function such that $f(x) = 1$ if and only if $x = w$ and $f(x) = 0$ otherwise. The quantum states can be encoded as $x \in \{0,1\}^n$, where $N = 2^n$, thus being able to be represented as qubits in the quantum computer. Define the oracle U_f acting on a quantum state, $|x\rangle$, as

$$U_f |x\rangle \rightarrow (-1)^{f(x)} |x\rangle .$$

Thus, the oracle does nothing to the unmarked quantum states and negates the phase of the marked state $|w\rangle$ as $U_f |w\rangle = (-1)^{f(w)} |w\rangle = (-1)^1 |w\rangle = - |w\rangle$. Geometrically, this unitary matrix corresponds to a reflection about the origin for the marked item in an $N = 2^n$ dimensional vector space.

The algorithm initialises the computation in the state $|0^{\otimes n}\rangle$. Without looking into the array there is no indication as to where the marked item is. This can be expressed by putting all the qubits in a uniform superposition (i.e., a Hadamard operation, $H^{\otimes n}$) as $|s\rangle = H^{\otimes n} |0^{\otimes n}\rangle = \frac{1}{\sqrt{N}} \sum_{x=0}^{N-1} |x\rangle$, where $N = 2^n$. Following this, the balanced superposition of states means that, upon measurement, all the items in the array have an equal probability of appearing.

The algorithm then uses a trick known as *amplitude amplification*, a process that essentially increases the amplitude (and thus, the probability) of a quantum state (in this case, the marked item). The amplitude amplification consists of an iteration (also known as the Grover operator) with the following steps.

1. Apply the oracle U_f which marks the required element in the array, leading to

   $$U_f |s\rangle .$$

   Geometrically, this leads to a reflection of the $|s\rangle$ state about the x-axis.

2. Superpose the quantum states in the state register using Hadamard transform, $H^{\otimes n}$.

3. Perform a conditional phase shift on the quantum register, with every computational basis state except $|0\rangle$ receiving a phase shift of -1. This step is expressed by a reflection on the state space described by the unitary $U_s = 2 |s\rangle \langle s| - I$, also called the Grover diffusion operator, which can easily be proven. The state is now

   $$U_s U_f |s\rangle .$$

   This transformation rotates the initial state $|s\rangle$ closer to the marked element.

4. Superpose the quantum states in the state register using Hadamard transform, $H^{\otimes n}$.

At the end of this routine the amplitude of the marked state will have been amplified. After $\sqrt{N}$ repetitions of the amplitude amplification, the probability of the marked state to be measured is arbitrarily close to 1. Grover's algorithm utilises the above routine, as can be seen in Algorithm 1.

Algorithm 1: Quantum search algorithm

```
1 Initialisation. Apply Hadamard transform to the state register, $H^{\otimes n}$, putting all
    states in a balanced superposition.
2 Marking the state. Apply the oracle $U_f$ on the state register.
3 Superposing. Apply Hadamard transform, $H^{\otimes n}$, to the state register.
4 Reflecting. Apply the reflection operator $U_s$
5 Superposing. Apply Hadamard transform, $H^{\otimes n}$, to the state register.
6 Iterating. Repeat $\sqrt{N}$ times from Step 2.
7 Measure. Measure the output to retrieve the marked item with probability close
    to 1.
```

1.2.4 Quantum Phase Estimation

Quantum phase estimation (QPE) is one of the most important subroutines in quantum computation. It serves as a central building block for many quantum algorithms, most notably Shor's algorithm for quantum factoring. The objective of the algorithm is the following: given a unitary operator U, estimate the phase θ in the representation $U|\psi\rangle = e^{2\pi i\theta}|\psi\rangle$. Here $|\psi\rangle$ is an eigenvector and $e^{2\pi i\theta}$ is the corresponding eigenvalue. Since U is unitary, all of its eigenvalues have modulus 1. In order to implement the QPE, the quantum system needs to be amassed by two registers: one that contains the state of $|\psi\rangle$, which is named the *state register*, and one in which the phase θ will be encoded with n qubits as $2^n\theta$, here called the *phase register*. The process of quantum phase estimation can be described is as follows.

1. **Setup:** initialise the state register with the state $|\psi\rangle$. The additional set of n qubits that form the phase register are set in state $|0^{\otimes n}\rangle$. After initialisation, the global state of the system will be:
$$|\psi_0\rangle = |0\rangle^{\otimes n}|\psi\rangle$$
2. **Superposition:** an n-bit Hadamard gate is applied on the phase register, leading the global state to:
$$|\psi_1\rangle = \frac{1}{2^{n/2}}(|0\rangle + |1\rangle)^{\otimes n}|\psi\rangle$$
3. **Controlled Unitary Operations:** consider a controlled unitary U_C that applies the unitary operator U on the target register (i.e., the phase register) only if its corresponding control qubit is $|1\rangle$. Since U is a unitary operator with eigenvector $|\psi\rangle$ such that $U|\psi\rangle = e^{2\pi i\theta}|\psi\rangle$, it follows that:

$$U^{2^j}|\psi\rangle = U^{2^j-1}U|\psi\rangle = U^{2^j-1}e^{2\pi i\theta}|\psi\rangle = \cdots = e^{2\pi i 2^j\theta}|\psi\rangle\,.$$

Applying all the n-controlled operations $U_{\mathrm{C}}^{2^j}$ with $0 \leq j \leq n-1$, and using the relation $|0\rangle \otimes |\psi\rangle + |1\rangle \otimes e^{2\pi\mathrm{i}\theta}|\psi\rangle = \big(|0\rangle + e^{2\pi\mathrm{i}\theta}|1\rangle\big) \otimes |\psi\rangle$, leads to the global state:

$$\begin{aligned}|\psi_2\rangle &= \frac{1}{2^{n/2}}\Big(|0\rangle + e^{2\pi\mathrm{i}\theta 2^{n-1}}|1\rangle\Big) \otimes \cdots \otimes \Big(|0\rangle + e^{2\pi\mathrm{i}\theta 2^{1}}|1\rangle\Big) \otimes \Big(|0\rangle + e^{2\pi\mathrm{i}\theta 2^{0}}|1\rangle\Big) \otimes |\psi\rangle \\ &= \frac{1}{2^{n/2}} \sum_{k=0}^{2^n-1} e^{2\pi\mathrm{i}\theta k}|k\rangle \otimes |\psi\rangle\end{aligned}$$

where k denotes the integer representation of n-bit binary numbers.

4. **Inverse Fourier Transform:** The quantum Fourier transform (QFT) maps an n-qubit input state $|x\rangle$ into an output as

$$\begin{aligned}\mathrm{QFT}\,|x\rangle =& \frac{1}{2^{n/2}}\Big(|0\rangle + e^{\frac{2\pi\mathrm{i}}{2}x}|1\rangle\Big) \otimes \Big(|0\rangle + e^{\frac{2\pi\mathrm{i}}{2^2}x}|1\rangle\Big) \otimes \ldots \otimes \\ & \Big(|0\rangle + e^{\frac{2\pi\mathrm{i}}{2^{n-1}}x}|1\rangle\Big) \otimes \Big(|0\rangle + e^{\frac{2\pi\mathrm{i}}{2^n}x}|1\rangle\Big).\end{aligned}$$

It is evident that the expression of the global state $|\psi_2\rangle$ is the result of applying QFT on the global expression $|\psi_1\rangle$ of Step 2. Therefore, to recover the state $|2^n\theta\rangle$, an inverse Fourier transform (QFT^{-1}) is applied on the phase register. Doing so, it is found that

$$|\psi_3\rangle = \frac{1}{2^{n/2}} \sum_{k=0}^{2^n-1} e^{2\pi\mathrm{i}\theta k}|k\rangle \otimes |\psi\rangle \xrightarrow{\mathrm{QFT}_n^{-1}} \frac{1}{2^n} \sum_{x=0}^{2^n-1}\sum_{k=0}^{2^n-1} e^{-\frac{2\pi\mathrm{i}k}{2^n}(x-2^n\theta)}|x\rangle \otimes |\psi\rangle$$

5. **Measurement:** the above expression peaks near $x = 2^n\theta$. For the case when $2^n\theta$ is an integer, measuring in the computational basis gives the phase in the phase register with high probability, as the global state now is:

$$|\psi_4\rangle = |2^n\theta\rangle \otimes |\psi\rangle.$$

For the case when $2^n\theta$ is not an integer, it can be shown that the above expression still peaks near $x = 2^n\theta$ with probability at least $4/\pi^2 \approx 40\%$ [30].

Thus, similarly to Grover's case, the above mathematical structure can be summarised as the quantum phase estimation algorithm, shown in Algorithm 2.

1.3 Aim and Objectives

This thesis is concentrated around the study and modelling of quantum noise, as well as building a framework that allows for the benchmarking of quantum computers while using an implemented model. The objectives of the research can be identified as follows.

- Quantum walks have the potential to form the basis for many advanced quantum algorithms that exhibit strong quantum advantage, like quantum Metropolis-Hastings, quantum Markov chain Monte Carlo methods or quantum search. The work is

Algorithm 2: Quantum phase estimation algorithm

```
1 Initialising. Initialise the quantum system. There exist two qubit registers, the
    phase register initialised in state $|0\rangle^{\otimes n}$, and the state register initialised in $|\psi\rangle$.
2 Superposing. Apply Hadamard gates on the phase register.
3 Applying the unitary. Apply the controlled unitary operation as shown in the
    third step of the mathematical analysis.
4 Inverse QFT. Apply inverse quantum Fourier transform in order to decode the
    state of the phase register.
5 Measure. Measure the phase register in order to get an approximation of the
    desired phase, $\theta$.
```

concentrated around scalable implementations of quantum walks on near-term quantum computers with the objective to produce an indication of the efficiency of each approach considering the characteristics of the quantum computer.

- Currently, quantum noise is one of the main obstacles preventing the achievement of universal quantum computation in practice. The study of quantum noise is of great benefit to the field and has the potential to offer a better understanding of the behaviour of qubits and machines. This research aims to produce such an understanding through detailed study of the noise and, ultimately, produce a model able to approximate the noisy evolution of a quantum computer as accurately as possible.

- The final objective revolves around benchmarking quantum machines using scalable and impactful applications. As quantum computers become bigger and better, they will eventually be capable of running high-level applications and larger quantum algorithms will become increasingly relevant. Thus, the goal is to create a concrete methodology that allows the benchmarking of quantum computers within such a scaling algorithmic environment.

Consequently, the overall goals of the work presented in this thesis can be summarised as:

(i) modelling and simulating the noisy behaviour of near-term quantum computers and (ii) benchmarking quantum computers in a scaling algorithmic environment.

1.4 Thesis Outline

The outline of the remainder of the present thesis is presented below. Each separate chapter introduces a part of the work done during the PhD and is accompanied by a discussion and conclusion on the findings, as well as potential future work.

- **Chapter 2** builds a quantum mechanical and mathematical background for the various types of quantum noise, as well as the quantum channels used throughout the modelling of the noisy behaviour of quantum computers. Additionally, it provides an introduction to the NISQ era and the characteristics of the quantum computers used for the experiments.

- **Chapter 3** presents the work undertaken on the subject of discrete-time and -space quantum walks and their circuit approaches, offering a detailed comparison regarding the characteristics of their implementations.
- **Chapter 4** details the development of a model that approximates the noisy evolution of a quantum computer. The techniques and mathematical and quantum mechanical foundations of the model are explained, as well as a large number of experiments executed that showcase the performance of the noise model.
- **Chapter 5** analyses a technique based on classical parameter optimisation that can be used to increase the efficiency of the quantum noise model (presented on Chapter 4) and approximate the noise with much greater precision.
- **Chapter 6** showcases a concrete framework for benchmarking near-term quantum computers with high-level, scaling quantum algorithms with useful applications. Furthermore, the results of intense benchmarking of various quantum computers are presented.
- **Chapter 7** offers a summary of the work done, conclusions and an evaluation of the research aims and objectives set in Section 1.3.

1.5 List of Publications

Portions of the work laid out within this thesis have been documented in the following publications.

- Konstantinos Georgopoulos, Clive Emary and Paolo Zuliani. “Comparison of Quantum-walk Implementations on Noisy Intermediate-scale Quantum Computers”, Physical Review A 103, 022408, February 2021, DOI: https://doi.org/10.1103/PhysRevA.103.022408. This published material occupies Chapter 3 (excluding Sections 3.1 and 3.3).
- Konstantinos Georgopoulos, Clive Emary and Paolo Zuliani. “Modelling and Simulating the Noisy Behaviour of Near-term Quantum Computers”, Physical Review A 104, 062432, December 2021, DOI: https://doi.org/10.1103/PhysRevA.104.062432. The work published in this paper occupies Chapters 4 and 5.
- Konstantinos Georgopoulos, Clive Emary and Paolo Zuliani. “Quantum Computer Benchmarking via Quantum Algorithms.”, arXiv:2112.09457 [quant-ph], e-print: https://arxiv.org/abs/2112.09457. This work has been submitted for journal publication, and is presented in Chapter 6.

Chapter 2

Quantum Noise and the NISQ Era

The theory and applications of the concepts introduced in Chapter 1 operate under the assumption that the corresponding quantum system(s) are *closed*, i.e., they do not exchange information in any form with their immediate environment. In reality, fully isolated systems are exceedingly difficult to come across or create in nature. For that very reason, the quantum computers that exist today and, more relevantly, the devices used to carry out the experiments detailed in this thesis, are not perfectly isolated, i.e., they are *open*. This means that information or energy is exchanged with the environment, which in addition to hardware infidelities, introduces errors during the computation in the form of *quantum noise*.

Currently and during the near-term future of the field of quantum computing, this noise is expected to dominate most aspects of a computation. Furthermore, the longer the evolution of a quantum computer the more noise it is subjected to, making interesting computations very hard to carry out reliably and preventing the current machines from achieving the goal of universal quantum computation. There are various techniques that can be used in order to minimise errors or noise during a computation, or otherwise, perform *error correction*. Nevertheless, the number of qubits necessary to error correct quantum computations increases with the size of the computation. Hence, the limited number of qubits within current quantum processors makes meaningful error correction infeasible.

2.1 Theory of Quantum Noise

This chapter aims to introduce the main theoretical aspects and concepts of quantum noise that are used throughout this research.

2.1.1 Pure and Mixed Quantum States

A quantum state is, in essence, a mathematical entity that provides a formal description of the outcome of a possible measurement on a quantum system. Knowledge of the quantum state, as well as how this state can evolve in time (via, for example, Schrödinger's equation) exhausts all that can be predicted for that quantum system. Assuming a system that has an arbitrary number of states, $|\psi_i\rangle$, with respective probabilities, p_i, where $i \in \mathbb{N}$ is an index, an *ensemble of quantum states* can be defined as $\{p_i, |\psi_i\rangle\}$. Such ensembles express pairs of quantum states along with their associated probabilities.

In quantum mechanics, a mixture of quantum states also represents another possible quantum state. In general, a quantum state that cannot be written as a mixture of other states is called a *pure state*. All other states are called *mixed states*. Mixed states arise in quantum mechanics in two different situations, first when the preparation of the system is not fully known, and thus one must deal with a statistical ensemble of possible preparations, and second when one wants to describe a physical system which is entangled with another, as its state can not be described by a pure state.

2.1.2 Trace and Partial Trace

The trace of an $n \times n$ square matrix, $B = [b_{ij}]_{i,j}$, is defined to be the sum of the diagonal elements of the matrix, b_{ii}, i.e.,

$$\mathrm{tr}(B) = \sum_{i=1}^{n} b_{ii}.$$

In quantum mechanics, the trace operation can be shown to be a quantum operation, as follows. Let $\mathcal{H}^{Q}$ be an arbitrary input Hilbert space with an orthonormal basis $|1\rangle \ldots |d\rangle$ and $\mathcal{H}^{Q'}$ be a d-dimensional output Hilbert space, spanned by the state $|0^{\otimes d}\rangle$. Define an arbitrary quantum operation, $\mathcal{E}$, acting on an arbitrary matrix ρ as

$$\mathcal{E}(\rho) = \sum_{i=1}^{d} |0\rangle \langle i|\rho|i\rangle \langle 0| .$$

It stands that $\mathcal{E}(\rho) = \mathrm{tr}(\rho)\,|0\rangle\langle 0|$ so that, up to the unimportant $|0\rangle\langle 0|$ multiplier, $\mathcal{E}$ is identical to the trace function.

In the above equation, the matrix ρ is a special operator that represents a *density matrix* or density operator. The density matrix is an alternative way to define the quantum state of a physical system. A more formal definition is provided below.

Definition 2.1 (Density matrix). *A density matrix, or density operator, is a positive semi-definite Hermitian operator acting on the Hilbert space of a quantum system. Assume an ensemble of quantum states $\{p_i, |\psi_i\rangle\}$, where $i \in \mathbb{N}$ is an index and $|\psi_i\rangle$ is a quantum state with associated probability p_i, the density matrix of the quantum system can be defined as*

$$\rho = \sum_{i} p_i \,|\psi_i\rangle \langle \psi_i| ,$$

where $\sum_i p_i = 1$.

The second useful term encountered within the manuscript is that of the *partial trace*. Assuming two arbitrary state spaces $\mathcal{H}^{C}$ and $\mathcal{H}^{D}$, and the composite space $\mathcal{H}^{C} \otimes \mathcal{H}^{D}$, the following definition is given.

Definition 2.2 (Partial trace). *The partial trace, tr_D, is a mapping from the density matrix ρ_{CD} in the composite space $\mathcal{H}^C \otimes \mathcal{H}^D$, onto the density matrix ρ_C in $\mathcal{H}^C$ as*

$$\rho_C \equiv \mathrm{tr}_D(\rho_{CD}).$$

In other words, the partial trace can be used to "trace out" a system from a composite system that it is a part of.

Most interestingly, it can be proven that the partial trace is also a quantum operation. Suppose a joint system QR and the wish to trace out the system R. Let $|j\rangle$ be a the basis of R. A linear operator $E_i : \mathcal{H}^{QR} \mapsto \mathcal{H}^Q$ is defined as

$$E_i \left(\sum_j \lambda_j |q_j\rangle |j\rangle \right) \equiv \lambda_i |q_i\rangle ,$$

where λ_j are complex numbers and $|q_j\rangle$ are arbitrary states of the system Q. Define $\mathcal{E}$ to be a quantum operation with operation elements $\{E_i\}$ such that

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

which is known to represent a quantum operation from system QR to system Q. Notice that

$$\mathcal{E}(\rho \otimes |j\rangle \langle j'|) = \rho \delta_{j,j'} = \mathrm{tr}_R(\rho \otimes |j\rangle \langle j'|),$$

where ρ is an arbitrary Hermitian operator on the state space of Q, and $|j\rangle$ and $|j'\rangle$ are members of the orthonormal basis for system R. By linearity of $\mathcal{E}$ and tr_R, it follows that $\mathcal{E} = \mathrm{tr}_R$, i.e., the partial trace is a quantum operation.

2.1.3 Operator-sum Representation

An elegant way to represent various quantum operations that is very useful to the study of quantum noise is the *operator-sum representation.* Assume a quantum system, A, in an initial state expressed as a density matrix ρ_A, and environment E in an initial pure state $|0_E\rangle \langle 0_E|$. A quantum operation, say $\mathcal{E}$, that takes as input the system AE can be defined as

$$\mathcal{E}(\rho_A) = \mathrm{tr}_E \left(U(\rho_A \otimes |0_E\rangle \langle 0_E|)U^\dagger \right), \tag{2.1}$$

where U is an arbitrary unitary operator.

Assume now that the environment is also a quantum system with an orthonormal basis, $|k\rangle$. The partial trace tr_E over the environment, as expressed in equation (2.1), can be evaluated as follows:

$$\begin{aligned} \mathcal{E}(\rho_A) &= \sum_k \langle k| \left(U(\rho_A \otimes |0_E\rangle \langle 0_E|)U^\dagger \right) |k\rangle \\ &= \sum_k E_k \rho_A E_k^\dagger, \end{aligned} \tag{2.2}$$

where $E_k = \langle k|U|0_E\rangle$ is an operator on the state space of the principal system. Equation (2.1) is known as the operator-sum representation of the quantum operator, $\mathcal{E}$. The operators $\{E_k\}$ are known as the operation elements, or otherwise, Kraus operators.

The operation elements need to satisfy a constraint known as the completeness relation. The completeness relation in the quantum case arises from the requirement that the trace of the operation $\mathcal{E}(\rho)$ needs to be equal to one, i.e., the operations are trace-preserving

$$1 = \mathrm{tr}(\mathcal{E}(\rho)) = \mathrm{tr}\left(\sum_k E_k \rho E_k^\dagger\right) = \mathrm{tr}\left(\sum_k E_k^\dagger E_k \rho\right).$$

This relationship is true for all ρ, and thus, as all quantum operations are trace-preserving, it follows that

$$\sum_k E_k^\dagger E_k = I. \tag{2.3}$$

There are also non-trace-preserving operations, for which $\sum_k E_k^\dagger E_k \leq I$, but these describe processes in which extra information about what occurred in the process is obtained by measurement, something that this thesis is not concerned with.

The operator-sum representation is an important result as it gives an intrinsic way of characterising the dynamics of the principal quantum system. It describes the dynamics without having to explicitly consider properties of the environment; all the necessary knowledge is contained into the operators E_k, which act on the principal system alone. This simplifies calculations and often provides considerable theoretical insight. Furthermore, many different environmental interactions may give rise to the same dynamics on the principal system. If it is only the dynamics of the principal system which are of interest, then it makes sense to choose a representation of the dynamics which does not include unimportant information about other systems.

2.1.4 Decoherence in Quantum Systems

In quantum mechanics, particles (for example electrons) are described by a wavefunction, a mathematical representation of the quantum state of a system. This mathematical entity "lives" within a Hilbert space and its evolution is described by Schrödinger's equation. In physics, two wave sources are perfectly coherent if their frequency and waveform are identical and their phase difference is constant, a property that is reflected within their wavefunctions. As long as there exists a definite phase relation between different states, the system is said to be coherent. A definite phase relationship is necessary to perform quantum computing. Coherence is preserved under the laws of quantum physics, given a *fully isolated* quantum system.

If a quantum system were perfectly isolated, it would maintain coherence indefinitely, but it would be impossible to manipulate or investigate it. If it is not perfectly isolated, for example during a measurement, coherence is shared with the environment and appears to be lost with time: a process called *quantum decoherence*. Thus, in simple terms, decoherence is the loss of quantum coherence due to interactions between a quantum system and its environment. As a result of this process, quantum behaviour is apparently lost, just as energy appears to be lost by friction in classical mechanics.

In quantum information science, decoherence can be viewed as the loss of information from a system into the environment (often modelled as a heat bath), since every system is loosely coupled with the energetic state of its surroundings. During decoherence, if the system is viewed in isolation, its dynamics are non-unitary and thus irreversible. On the other hand, the combined system and environment evolves in unitary fashion if it is a perfectly isolated system in turn. As with any coupling, entanglements are generated between the system and environment. These have the effect of sharing quantum information with, or transferring it to the surroundings.

Quantum decoherence has been used to understand the collapse of the wave function in quantum mechanics. Decoherence does not actually generate this phenomenon but only provides an explanation for apparent collapse, as the quantum nature of the system "leaks" into the environment. That is, components of the wave function are decoupled from a coherent system and acquire phases from their immediate surroundings. A total superposition of the global or universal wavefunction still exists (and remains coherent at the global level), but its ultimate fate remains an interpretational and philosophical issue. Specifically, within the context of this thesis, the concept of decoherence does not attempt to explain the measurement problem. Rather, it provides a mechanism for the transition of the system to a mixture of states that seem to correspond to the states observers perceive. Moreover, observations reveal that this mixture looks like a proper quantum ensemble in a measurement situation, as measurements lead to the "realisation" of precisely one state in the ensemble or quantum superposition. In other words, the stochastic nature of a quantum superposition breaks down to an observable classical outcome under the effects of decoherence.

There have been many theoretical and philosophical studies and arguments around the nature of decoherence, what it means and how it affects the nature of reality. Some researchers support decoherence as an absolute phenomenon inducing macroscopic (classical) behaviour into an otherwise microscopic (quantum) reality by collapsing the superposed states of quantum particles. Others believe that quantum reality never truly breaks down and what observers experience as classical behaviour is the partial collapse of the wavefunction of a universe whose state is endlessly branching in an eternal cosmic bath, a view dubbed as "the many-worlds interpretation of quantum mechanics". In other words, while within the observable universe decoherence drives the classical nature interior observers experience, an outside view of the world bubble as the one truly isolated quantum system would present the universe in a superposition of all its possible states, all the possible histories happening at the same time and all the branches of reality merging into a single, uncollapsed wavefunction, one that, of course, would be coherent while remaining fully isolated. If the external observer "took a peek" into the bubble (or made a measurement of the entire universe) then within their world history, the wavefunction of the universe would (partially) collapse, and one of the superposed histories (not necessarily one within which this thesis is written) would be laid bare, decohered for them to observe.

Whichever point of view one supports, there is no denying that quantum decoherence is the mechanism responsible for a visible, fully or partially collapsible universe, one in which humans can live and (potentially) thrive.

2.1.5 Types of Quantum Noise

This section offers a general overview of the various types of quantum noise that occur during a quantum computation. In a closed system, the probabilistic evolution of a qubit state is deterministic. That is, if the starting state of the qubit and its Hamiltonian are known, then it is possible to predict the probabilistic state of the qubit at any time in the future. However, in open systems, the situation changes. The qubit now interacts with uncontrolled degrees of freedom in its environment, which is referred to as fluctuations or noise. In the presence of noise, as time progresses, the qubit state looks less and less like the state that would have been predicted and, eventually, the state is lost. There are many different sources of noise that affect quantum systems, and they can be broadly categorised into two primary types: (i) systematic noise and (ii) stochastic noise.

Systematic Quantum Noise

Systematic noise arises from a process that is traceable to a fixed gate or measurement error. For example, a quantum gate is applied to a qubit that is believed will impart a 180° rotation. However, the gate is not tuned properly and, rather than rotating the qubit 180°, it slightly over-rotates or under-rotates the qubit by a fixed amount. The underlying error is systematic, and it therefore leads to the same rotation error each time it is applied. However, when such erroneous gates are used in practice in a variety of control sequences, the observed results may appear to be influenced by random noise. This is because the gate is generally not applied in the same way for each experiment: it could be applied a different number of times, interspersed with different gates in different orders, and therefore generally differs from experiment to experiment. However, once systematic errors are identified, they can generally be corrected through proper calibration or the use of improved hardware.

Stochastic Quantum Noise

The second type of noise is stochastic noise, arising from random fluctuations of parameters that are coupled to the qubit [31]. For example, thermal noise of a resistor in the control lines leading to the qubit will have voltage and current fluctuations with a noise power that is proportional to both temperature and bandwidth. Or, the oscillator that provides the carrier for a qubit control pulse may have amplitude or phase fluctuations. Additionally, randomly fluctuating electric and magnetic fields in the local qubit environment can couple to the qubit. This creates unknown and uncontrolled fluctuations of one or more qubit parameters, which leads to qubit decoherence.

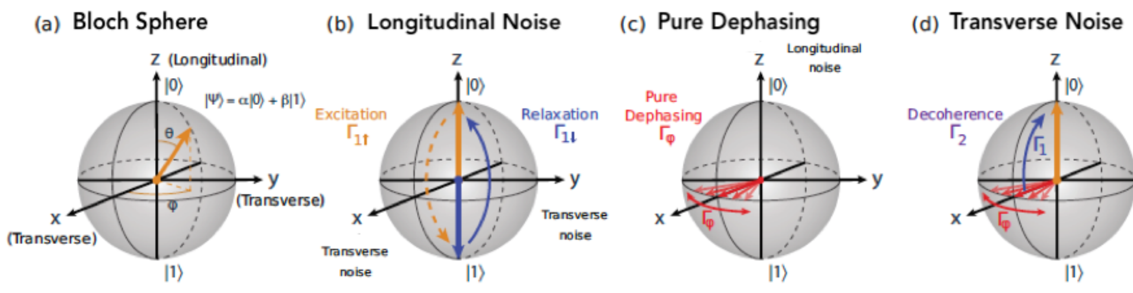


Fig. 2.1 Longitudinal and transverse noise, otherwise called thermal decoherence and dephasing respectively, represented on the Bloch sphere [32].

2.1.6 Thermal Decoherence

Decoherence is a mechanism encountered often in quantum mechanics which was introduced in Section 2.1.4. Within the context of this chapter decoherence is considered as thermal exchange between the quantum system and its immediate environment. It describes the drive of the quantum system towards an equilibrium state (or otherwise called a Gibbs state). This can happen in one of two ways: (i) either by spontaneous emission of a quantum of energy (a photon), called *thermal relaxation*, which drives the qubit towards the ground state ($|0\rangle$), or (ii) by spontaneous absorption of a photon, called *thermal excitation*, which takes the qubit to an excited state (for example, $|1\rangle$).

The thermal decoherence rate, Γ_1, describes depolarisation along the qubit quantisation axis. Considering a Bloch sphere, a qubit with polarisation $p_q = 1$ is entirely in the ground state $|0\rangle$ at the north pole, $p_q = -1$ is entirely in the excited state $|1\rangle$ at the south pole, and $p_q = 0$ is a completely depolarised mixed state at the center of the Bloch sphere. As illustrated in Figure 2.1(b), longitudinal noise is caused by transverse noise, via the x- or y-axis, with the intuition that off-diagonal elements of an interaction Hamiltonian are needed to connect and drive transitions between states $|0\rangle$ and $|1\rangle$.

Decoherence occurs due to energy exchange with the environment, generally leading to both an up transition rate $\Gamma_{1\uparrow}$ (thermal excitation), and a down transition rate $\Gamma_{1\downarrow}$ (thermal relaxation). Together, these form the thermal decoherence (or longitudinal noise) rate Γ_1 as

$$\Gamma_1 = \Gamma_{1\uparrow} + \Gamma_{1\downarrow} = \frac{1}{T_1}.$$

The parameter T_1 is the decoherence time and it is the characteristic time scale over which a qubit will decohere to an equilibrium state. For superconducting qubits, this steady-state value is generally the ground state $|0\rangle$, due to Boltzmann statistics and typical operating conditions. This will be proven in later sections specifically for the quantum computers that the engineered noise models simulate. The Boltzmann equilibrium statistics leads to the detailed balance relationship $\Gamma_{1\uparrow} = e^{-\hbar\omega_q/k_\mathrm{B}\Theta}$, where Θ is the temperature and k_B is Boltzmann's constant, with an equilibrium qubit polarisation approaching $p_q = \tanh(\hbar\omega_q/2k_\mathrm{B}\Theta)$. Typical qubits are designed at frequency $\omega_q/2\pi \approx 5\,\mathrm{GHz}$ and are operated at dilution refrigerator temperatures $\Theta = 15$ to $20\,\mathrm{mK}$. In this limit, the up-rate $\Gamma_{1\uparrow}$ is exponentially suppressed by the Boltzmann factor $e^{-\hbar\omega_q/k_\mathrm{B}\Theta}$, and so only the down-rate $\Gamma_{1\downarrow}$ contributes significantly, relaxing the population to the ground state. Thus, qubits generally spontaneously lose energy to their cold environment, but the environment rarely introduces a qubit excitation. As a result, the equilibrium polarisation approaches unity [33]. In other words, the extremely low temperatures within the dilution refrigerator induce thermal deexcitation with exponential probability in T_1, whereas thermal excitation occurs extremely rarely.

2.1.7 Dephasing

Finally, dephasing can be described as the absolute form of decoherence of a quantum system occurring as perturbations (or loss of phase) of the qubits towards classical behaviour during the evolution of the quantum system (or the quantum computation). In other words, dephasing describes ways in which coherence decays over time, leading the system to no longer exhibit quantum mechanical behaviour.

Dephasing in general can be broken down to two distinct mechanisms, pure dephasing and (simple) dephasing. The pure dephasing rate Γ_Φ describes decoherence in the x-y plane of the Bloch sphere, as showcased in Figure 2.1(c). It is referred to as pure in order to distinguish it from other phase breaking processes such as energy excitation or relaxation, as well as the (simple) dephasing (also referred to as transverse noise).

The dephasing (or transverse noise) rate, Γ_2, describes the loss of coherence of a superposed state pointed along the x-axis on the equator of the Bloch sphere, as illustrated in Figure 2.1(d). Decoherence is caused in part by longitudinal noise, which fluctuates the qubit frequency and leads to pure dephasing Γ_Φ (red). It is also caused by transverse noise, which leads to energy relaxation of the excited-state component of the superposition state at a rate Γ_1 (blue). Such a relaxation event is also a phase-breaking process, because once it occurs, the Bloch vector points to the north pole, $|0\rangle$, and there is no longer any knowledge of which direction the Bloch vector had been previously pointing along the equator; the relative phase of the superposition state is lost.

Similarly to the thermal decoherence time, T_1, the dephasing time $T_2 = 1/\Gamma_2$ defines the time it takes for a qubit to lose its phase. Decoherence and dephasing times T_1 and T_2, as well as pure dephasing time, $T_\Phi = 1/\Gamma_\Phi$, follow the well-known relationship

$$T_2 = 2T_1 + T_\Phi. \tag{2.4}$$

In the remainder of the thesis, for the purposes of a simpler analysis, the pure dephasing, T_Φ, is considered as a time embedded in the dephasing time T_2 of the quantum system. Thus, equation (2.4) can be rewritten as

$$T_2 = 2T_1, \tag{2.5}$$

a relationship that proves one of the most important in noise modelling.

2.1.8 Quantum Channels

Perhaps the most important concept of modeling and simulating quantum noise is the constructs called *quantum channels*. These represent the building blocks of most noise models and are invaluable to the field. This section aims to introduce the most common quantum channels and ones that will be used throughout this thesis.

In the general sense, the dynamics of a closed quantum system are *unitary*. In other words, assuming the state of a quantum system is initially $|\psi\rangle$ then, in a perfectly isolated system, a unitary operator U can evolve that state perfectly and without error, denoted through the operation $U|\psi\rangle$. In terms of density matrices, the above operation is equivalent to the map $|\psi\rangle\langle\psi| \mapsto U|\psi\rangle\langle\psi| U^\dagger$. This translates to the fact that a quantum evolution occurring within a closed system is reversible, i.e., it will not create or destroy quantum information.

Another operation that is very useful and often occurs in the study of quantum noise is the addition or, conversely, the removal of a part of a quantum system. In general, adding a system can be done by using a tensor product which introduces the new part to the initial system. Discarding a system can be done using the partial trace, as shown in Definition 2.2. Mathematically, these two operations can be expressed as follows

- **Adding a system:** assuming a quantum system with density matrix, ρ_A, and a second quantum system with density matrix ρ_B, the composite quantum system, namely ρ_{AB}, can be expressed as $\rho_{AB} \mapsto \rho_A \otimes \rho_B$

- **Discard a subsystem:** assuming a composite system with density matrix ρ_{AB}, then the subsystem with density matrix ρ_B can be removed (otherwise called, *traced out*) from the composite system using a partial trace over the system B (tr_B) as $\rho_{AB} \mapsto \rho_A = \text{tr}_B(\rho_{AB})$ (see Section 2.1.2). Similarly, one can remove the subsystem with density matrix ρ_A.

First of all, one important definition to assist with understanding quantum channels is that of an *isometry*, as follows.

Definition 2.3 (Isometry)**.** *An isometry K is a map in $L(\mathbb{C}^{d_A}, \mathbb{C}^{d_B})$ for $d_B \geq d_A$ such that $K^\dagger K = I_{d_A}$. Assuming arbitrary operators O_1, O_2, $L(O_1, O_2)$ is the space of all linear operations mapping $O_1 \mapsto O_2$. Note for any state $|\psi\rangle$, $\||\psi\rangle\| = \|K|\psi\rangle\|$, $\forall |\psi\rangle \in A$. Furthermore, $KK^\dagger = I_{d_B}$ iff $d_A = d_B$. In terms of density matrices, an isometry maps $\rho \mapsto K\rho K^\dagger$.*

The benefit of using isometries, as defined above, is the fact they can be used to combine the two operations of unitarity and adding a quantum system in order to create a composite system as $\mathcal{N}(\rho) = K\rho K^\dagger$, where K is an isometry, ρ is the density matrix of the initial quantum system and $\mathcal{N}(\rho)$ is a quantum operation. On the other hand, a part of the quantum system can be discarded using a partial trace $\mathcal{N}(\rho_{AB}) = \text{tr}_B(\rho_{AB})$; i.e., $\mathcal{N}$ here represents the partial trace operation. The above operations of isometry and partial trace can also be combined to add and discard parts of a closed quantum system appropriately. For example, considering a quantum system A and a composite system B together with its environment E as a closed system, an operation $\mathcal{N}$ can be defined with the following effect:

$$\mathcal{N}(\rho) = \mathrm{tr}_E(K\rho K^\dagger). \tag{2.6}$$

Here, an isometry K maps the system A on to the space $B \otimes E$ with density matrix ρ, then the partial trace tr_E traces out the environment, E. It is noteworthy that all these operations are not applicable in the presence of decoherence, which emphasises the necessity of a *closed* quantum system.

The isometries, K, of the form introduced above can be also written as $K = \sum_e K_e \otimes |e\rangle$, where $\{|e\rangle\}$ is an orthonormal basis and $K_e \in L(A, B)$. This allows equation (2.6) to be written as (discarding the density matrix subscript $_{BE}$ for generalisation purposes):

$$\mathcal{N}(\rho) = \mathrm{tr}_E \left[\sum_{e_1, e_2} K_{e_1} \rho K_{e_2}^\dagger \otimes |e_1\rangle \langle e_2| \right].$$

Evaluating the partial trace yields

$$\mathcal{N}(\rho) = \sum_e K_e \rho K_e^\dagger. \tag{2.7}$$

Being linear, $\mathcal{N}$ acts on every density operator or quantum system.

A linear map of the form of equation (2.7) is called a *quantum channel.* Additionally, the expressed form of equation (2.7) is commonly called the *Kraus decomposition* (see Section 2.1.3), with K_e being the *Kraus operators.* Of course, K is still an isometry. Additionally, $K_e \in L(A, B)$ and thus the size of the Kraus operators is known. These operators are blocks from a larger matrix, which is also an isometry. Thus, considering the isometry condition from Definition 2.3, i.e $K^\dagger K = I$, one can express it as a condition on all of the blocks as

$$I = \left(\sum_{e_1} K_{e_1}^\dagger \otimes \langle e_1| \right) \left(\sum_{e_2} K_{e_2} \otimes |e_2\rangle \right).$$

This can also be written as

$$\sum_e K_e^\dagger K_e = I$$

meaning that the Kraus operators obey the completeness condition. Here, e is finite or countable, and the operators are also they are positive semidefinite.

This relation also works in reverse. If a set of Kraus operators $\{K_e\}$ satisfies the Kraus operator conditions, then the matrix K satisfies the isometry condition. That means the construct is a quantum channel of the form (2.7), which can also be written in the form of (2.6).

A quantum channel maps density operators to density operators; that is, it has the following properties.

1. **Linearity:** $\mathcal{N}(\alpha\rho_1 + \beta\rho_2) = \alpha\mathcal{N}(\rho_1) + \beta\mathcal{N}(\rho_2)$.

2. **Preserves Hermiticity:** $\rho = \rho^\dagger \implies \mathcal{N}(\rho) = \mathcal{N}(\rho^\dagger)$.

3. **Preserves positivity:** $\rho \geq 0 \implies \mathcal{N}(\rho) \geq 0$.

4. **Preserves trace:** $\mathrm{tr}(\mathcal{N}(\rho)) = \mathrm{tr}(\rho)$.

Given a quantum channel, $\mathcal{N}$, acting on a quantum system A with Kraus operators $\{K_e\}$, one may introduce the auxiliary system B with Hilbert space dimensions matching the number of Kraus operators. Thus, the main application of quantum channels in the present research derives exactly from the above argument: by using quantum channels and Kraus operators one can introduce noise in a quantum system as a stochastic unitary evolution that interferes with the original closed system.

Finally it is noteworthy that some quantum channels are characterised as *unital* channels, as shown by the following definition.

Definition 2.4 (Unital quantum channel)**.** *A quantum channel, $\mathcal{N}$, is unital if it preserves the identity operator, i.e., $\mathcal{N}(I) = I$.*

In other words, a quantum channel is unital if all trivial observables remain trivial after applying the quantum channel. In general, if the Kraus operators of a channel, $\mathcal{N}$, satisfy

$$\sum_e K_e^\dagger K_e = I = \sum_e K_e K_e^\dagger,$$

then both the channel $\mathcal{N}$ and its dual $\mathcal{N}^*$ are unital channels.

The following sections introduce some of the most well-known quantum channels that are used extensively throughout this thesis.

Depolarising Channel

The first channel of interest is the *depolarising channel.* This channel essentially models a decohering qubit, and has particularly nice symmetry properties. The effects of the depolarising channel can be described as follows: induce error in the system with probability p or do nothing with probability $1 - p$. Considering an orthonormal basis for a single qubit $\{|0\rangle, |1\rangle\}$, the depolarising channel can introduce three types of error in the system:

1. **Bit-flip error:** $\begin{matrix} |0\rangle \mapsto |1\rangle \\ |1\rangle \mapsto |0\rangle \end{matrix}$ or otherwise $|\psi\rangle \mapsto \sigma_{\mathrm{bf}} |\psi\rangle$, where $\sigma_{\mathrm{bf}} = \sigma_x = X$ is the Pauli-X operator.

2. **Phase-flip error:** $\begin{matrix} |0\rangle \mapsto |0\rangle \\ |1\rangle \mapsto -|1\rangle \end{matrix}$ or otherwise $|\psi\rangle \mapsto \sigma_{\mathrm{pf}} |\psi\rangle$, where $\sigma_{\mathrm{pf}} = \sigma_z = Z$ is the Pauli-Z operator.

3. **Both the above:** $\begin{matrix} |0\rangle \mapsto i|1\rangle \\ |1\rangle \mapsto -i|0\rangle \end{matrix}$ or otherwise $|\psi\rangle \mapsto \sigma_{\mathrm{bpf}} |\psi\rangle$, where $\sigma_{\mathrm{bpf}} = \sigma_y = Y$ is the Pauli-Y operator.

Unitary representation. The depolarising channel can be realised as an isometry mapping a quantum system A to a composite system AE, where E can represent, for example, the immediate environment with orthonormal basis $|k\rangle$ (like in Section 2.1.3). The action can be written as the unitary

$$\begin{aligned} U_{A\mapsto AE} = |\psi\rangle_A \mapsto \sqrt{1-p}\,|\psi\rangle_A \otimes |0\rangle_E + \sqrt{\frac{p}{3}}(\sigma_{\text{bf}}\,|\psi\rangle_A \otimes |1\rangle_E \\ + \sigma_{\text{pf}}\,|\psi\rangle_A \otimes |2\rangle_E + \sigma_{\text{bpf}}\,|\psi\rangle_A \otimes |3\rangle_E). \end{aligned} \tag{2.8}$$

The environment evolves to one of the four mutually orthogonal states $|k\rangle$, for $k \in 0,\dots,3$. It is noteworthy that here the states $|k\rangle$ are not qubit states but they represent physical states of the environment. For example, they could be realised as a ground and multiple excited states of the environment viewed as a quantum system.

Operator-sum representation. The operator-sum representation (see Section 2.1.3), or otherwise called Kraus decomposition, can be used as an alternative way to describe a quantum channel. To obtain a Kraus decomposition of the depolarising channel the partial trace over the environment is evaluated in the $\{|e\rangle_E\}$ basis. Then

$$K_0 = \sqrt{1-p}I, K_1 = \sqrt{\frac{p}{3}}\sigma_{\text{bf}},\ K_2 = \sqrt{\frac{p}{3}}\sigma_{\text{pf}},\ K_3 = \sqrt{\frac{p}{3}}\sigma_{\text{bpf}},$$

where, in essence, K_0 describes the case where no error occurs, K_1 is a bit-flip error, K_2 is a phase-flip error and K_3 is a bit- and phase-flip error.

The next step is to check the normalisation condition. Using $\sigma_i^2 = I$ the following is derived

$$\sum_e K_e^\dagger K_e = \left((1-p) + 3\frac{p}{3}\right) I = I,$$

where e is finite or countable, thus showing that the operators follow the completeness condition and are, indeed, Kraus operators.

An arbitrary initial density matrix, ρ, evolves through the quantum channel, $\mathcal{N}$, as

$$\mathcal{N}(\rho) = \rho \mapsto \rho' = (1-p)\rho + \frac{p}{3}(\sigma_{\text{bf}}\rho\sigma_{\text{bf}} + \sigma_{\text{pf}}\rho\sigma_{\text{pf}} + \sigma_{\text{bpf}}\rho\sigma_{\text{bpf}}) \tag{2.9}$$

where the state exists in a superposition of the four possible final states of the composite quantum system and environment.

Finally, a note about the reversibility of the quantum channel. Evidently, one can reverse a uniform contraction of the Bloch sphere with a uniform expansion. But the trouble is that the expansion of the Bloch sphere when the system is not closed, i.e., under the effects of decoherence, is not a channel, because it is not a positive, unitary process. This means that decoherence can shrink the sphere, but no physical process can expand it again. In other words, a channel running backwards in time when the quantum system is not closed is not a quantum channel.

Phase-damping Channel

The next channel of interest is the *phase-damping channel.* As the name suggests, it models the noise that affects the phase of a quantum system. This channel is quite interesting because it reveals a very instructive picture of decoherence in realistic physical situations.

Unitary representation. An isometric representation of the phase-damping channel applied on a quantum system A with an environment E can be given as

$$\begin{aligned} |0\rangle_A &\mapsto \sqrt{1-p}\,|0\rangle_A \otimes |0\rangle_E + \sqrt{p}|\,|0\rangle_A \otimes |1\rangle_E\,, \\ |1\rangle_A &\mapsto \sqrt{1-p}\,|1\rangle_A \otimes |0\rangle_E + \sqrt{p}|\,|1\rangle_A \otimes |2\rangle_E\,. \end{aligned} \tag{2.10}$$

In this case, unlike the depolarising channel, assuming a system A, there are no transitions in the $\{|0\rangle\,, |1\rangle\}$ qubit basis. Instead, with probability p the environment "scatters" off of the qubit, ending up in the state $|1\rangle_E$ if A was in state $|0\rangle_A$ and on state $|2\rangle_E$ if it was in state $|1\rangle_A$.

Operator-sum representation. Evaluating the partial trace over E in the basis $\{|0\rangle_E\,, |1\rangle_E\,, |2\rangle_E\}$ one can obtain the Kraus operators

$$K_0 = \sqrt{1-p}I,\ K_1 = \sqrt{p}\begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix},\ K_2 = \sqrt{p}\begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix}.$$

It is very easy to confirm the operators satisfy completeness condition $K_0^2 + K_1^2 + K_2^2 = I$.

According to [34] there is no need for three Kraus operators and the Kraus decomposition can be done as

$$K_1 = \frac{\sqrt{p}}{2}(I + \sigma_{\text{pf}}),\ K_2 = \frac{\sqrt{p}}{2}(I - \sigma_{\text{pf}}). \tag{2.11}$$

Hence, the channel can be expressed as

$$\mathcal{N}(\rho) = \sum_e K_e \rho K_e = \left(1 - \frac{p}{2}\right)\rho + \frac{1}{2}p\sigma_{\text{pf}}\rho\sigma_{\text{pf}}, \tag{2.12}$$

with e finite or countable. Evidently, the channel can be simply described as a phase-flip, σ_{pf}, happening with probability $p/2$ and nothing happening with probability $1 - p/2$. The initial density matrix ρ evolves through the quantum channel, $\mathcal{N}$, to

$$\mathcal{N}(\rho) = \begin{pmatrix} \rho_{00} & (1-p)\rho_{01} \\ (1-p)\rho_{10} & \rho_{11} \end{pmatrix}$$

Decoherence. Preskill provides a very nice interpretation of the dephasing channel in his lecture notes [34]. In essence, the phase-damping channel can be described as the interaction between a classical (heavy/macrorealistic) particle with a background gas of light (quantum/microrealistic) particles, i.e., photons. The quantum state of the particle can be described by a density matrix ρ obtained by tracing over degrees of freedom of the photons.

The analysis of the dephasing channel indicates that if the photons are scattered by the heavy particle at a rate Γ, then an exponential decay can be observed of the off-diagonal terms in ρ, expressed as $e^{-\Gamma t}$, rendering them obsolete for $t >> \Gamma^{-1}$. At that point the quantum state is subjected to pure decoherence, completely losing the coherent superposition of the position eigenstates of the particle.

As the classical particle is very heavy, its motion is affected very little by the scattered photons due to its large inertia. Thus, there are two disparate time scales relative to its dynamics. The first one is the damping time scale which is the time needed for a significant amount of the particle's momentum to be transferred into the photons. This is a long time due to the large inertia of such a heavy particle, as explained before. The second is the decoherence time scale. In this model the time scale for decoherence is of order Γ, the time for a single photon to be scattered by the large particle, which is far shorter than the damping time scale. Thus, it can be concluded that for a macroscopic object, the decoherence is extremely fast.

The above viewpoint expresses in a nice way why the universe is full of decoherence. A lot of philosophical or non-philosophical views can be explained or argued in this manner, like the "multiverse", why space is so decoherent or why is it so hard to maintain quantum superposition in the classical scale. A coherent superposition of macroscopically distinguishable states of a "heavy" classical object decoheres incredibly rapidly compared to its damping rate. Of course, a much larger philosophical discussion can follow here including macrorealism versus microrealism and what can be considered a macroscopic or a quantum system or how damping time scales versus decoherence time scales can fit into this discussion. Unfortunately, this falls outside the scope of this research.

Amplitude-damping Channel

The last quantum channel discussed is called the *amplitude-damping channel.* This can be used to model the decay rate of an excited qubit due to fluctuations of its energy (for example, thermal exchange).

Unitary representation. Assume the atomic ground state of a quantum system A is $|0\rangle_A$ and the excited state is $|1\rangle_A$. Also assume an environment, E, like for example an electromagnetic field initially in its vacuum state, $|0\rangle_E$. If the initial state of the quantum system is the excited state, $|1\rangle_A$, then, after some time, there is a probability p_e that the excited state has decayed to the ground state, $|0\rangle_A$, by photon emission. After this effect, the environment will be in the state $|1\rangle_E$. This evolution can be described by a unitary transformation according to

$$\begin{aligned} |0\rangle_A \otimes |0\rangle_E &\mapsto |0\rangle_A \otimes |0\rangle_E\,, \\ |1\rangle_A \otimes |0\rangle_E &\mapsto \sqrt{1-p_\mathrm{e}}\,|1\rangle_A \otimes |0\rangle_E + p_\mathrm{e}\,|0\rangle_A \otimes |1\rangle_E\,. \end{aligned} \tag{2.13}$$

Note that, if the qubit starts out in its ground state then no transition occurs between the system A and the environment E as the qubit has no photon to emit.

In the contrary case, the system is initially in the ground state, $|0\rangle_A$, and will spontaneously absorb a photon from its environment (if it has one) with probability p_{a}, which will lead the quantum system to be in the excited state, $|1\rangle_A$.

$$\begin{aligned} &|0\rangle_A \otimes |0\rangle_E \mapsto |0\rangle_A \otimes |0\rangle_E\,, \\ &|0\rangle_A \otimes |1\rangle_E \mapsto \sqrt{1-p_{\mathrm{a}}}\,|0\rangle_A \otimes |1\rangle_E + p_{\mathrm{a}}\,|1\rangle_A \otimes |1\rangle_0\,. \end{aligned} \tag{2.14}$$

As will be shown in later Chapters, within a superconducting quantum computer, spontaneous absorption occurs very rarely compared to excitation within the quantum computer due to the low temperatures achieved by the dilution refrigerator. Hence, for simplicity within this introduction, it is assumed that the composite quantum system AE is fully isolated and the environment, E, has no photon for the system A to absorb. Nevertheless, it is relatively easy to derive the relevant analysis.

Kraus operators representation. By evaluating the partial trace over the environment E in the orthonormal basis $\{|0\rangle_E\,, |1\rangle_E\}$, one can find the Kraus operators

$$K_0 = \begin{pmatrix} 1 & 0 \\ 0 & \sqrt{1-p} \end{pmatrix}, \; K_1 = \begin{pmatrix} 0 & \sqrt{p} \\ 0 & 0 \end{pmatrix}. \tag{2.15}$$

It is again easy to verify the completeness condition: $K_0^\dagger K_0 + K_1^\dagger K_1 = I$. The operator K_1 induces a quantum jump, i.e., the decay from the excited state $|1\rangle_A$ to the ground state $|0\rangle_A$ with probability p. The operator K_0 describes the state change when no jump occurs. The density matrix evolves through the quantum channel, $\mathcal{N}$, as

$$\begin{aligned} \rho \mapsto \mathcal{N}(\rho) = K_0 \rho K_0^\dagger + K_1 \rho K_1^\dagger &= \begin{pmatrix} \rho_{00} & \left(\sqrt{1-p}\right)\rho_{01} \\ \left(\sqrt{1-p}\right)\rho_{10} & (1-p)\rho_{11} \end{pmatrix} + \begin{pmatrix} p\rho_{00} & 0 \\ 0 & 0 \end{pmatrix} \\ &= \begin{pmatrix} \rho_{00} + p\rho_{11} & \left(\sqrt{1-p}\right)\rho_{01} \\ \left(\sqrt{1-p}\right)\rho_{10} & (1-p)\rho_{11} \end{pmatrix}. \end{aligned} \tag{2.16}$$

Time dependence. Knowing the spontaneous decay rate per unit time, denoted Γ, then the decay occurs with probability $p = \Gamma \Delta t << 1$ in a small time interval Δt. The density operator after time $t = n\Delta t$ can be found by applying the channel n times in succession. The ρ_{11} matrix element then decays as

$$\rho_{11} \mapsto (1-p)^n \rho_{11},$$

where $(1-p)^n = (1-\Gamma t/n) \to e^{-\Gamma t}$ is the exponential decay law. The off diagonal entries decay by the factor $(1-p)^{n/2} = e^{-\Gamma t/2}$. Hence

$$\rho(t) = \begin{pmatrix} \rho_{00} + (1-e^{-\Gamma t})\rho_{11} & e^{-\Gamma t/2}\rho_{01} \\ e^{-\Gamma t/2}\rho_{10} & e^{-\Gamma t}\rho_{11}. \end{pmatrix} \tag{2.17}$$

2.2 Noisy Intermediate-Scale Quantum Devices

Finally, this section offers a brief reference to the current state of the quantum computing field within the so called *Noisy Intermediate-Scale Quantum* (NISQ) era, as well as a presentation of the quantum devices used for the experiments throughout the research.

2.2.1 The NISQ Era

Currently, the field of quantum computing and quantum technology is in, what is called within the wider quantum community, the *NISQ* era, an acronym attributed to Preskill [35] and stands for *Noisy Intermediate-Scale Quantum era.* In this term, the "intermediate-scale" refers to the size of quantum computers which are available at this stage of the field, with the number of qubits in newer machines ranging from around 20 to a few hundred. A 50-qubit quantum computer represents a significant milestone, as it forms a relative border on the size of quantum systems that can be simulated through brute force using the most powerful existing digital supercomputers [36–38]. "Noisy" emphasises the imperfect control over those qubits that arises either due to the effects of various forms of decoherence and dephasing or due to infidelities of the hardware itself. The quantum noise places serious limitations on what quantum devices can achieve in the near-term, thus rendering relevant research invaluable.

The number of qubits has been often emphasised as a measure of how difficult it is to perform simulations of a quantum computer on a classical device. Of course, this is not the only factor of interest. A potentially more impactful aspect is the quality of the qubits, and in particular, the accuracy with which quantum operations can be performed on them. Currently, even with the best hardware for controlling trapped ions [39] or superconducting circuits [40], the error rate per gate for two-qubit gates is around the 0.1% level. Furthermore, there is high uncertainty on whether such error rates can be maintained in larger devices with many qubits. Naively, with these noisy devices it is expected that a circuit containing more than about a thousand gates could not be executed reliably — that is, a thousand fundamental two-qubit operations — because the noise will overwhelm the result. Of course, in machines with higher levels of noise, this number drops dramatically. Such limitations on circuit size impose a ceiling on the computational power of NISQ technology and highlight the importance of studying the quantum noise and methods to overcome the noise within quantum machines.

Eventually quantum technologies will progress towards universal quantum computing, potentially using quantum error correction to scale up to larger circuits. Additionally, quantum computers will get better as researchers learn how to deal with the noise and noisy qubits more effectively. It is, therefore, essential for the NISQ era and for the future of the field to be able to characterise the quantum noise present within quantum machines, as well as be able to benchmark the quality of a quantum machine, not only at the level of a qubit but, as quantum computers become bigger and better, also at the level of quantum circuits and algorithms.

2.2.2 The NISQ Computers

Within this thesis, a large amount of experiments is undertaken and used to strengthen the theoretical results of the research. To carry out these experiments, four IBM quantum computers are used: the IBMQx15 Melbourne, the IBMQx5 Bogota, the IBMQx5 Santiago and the IBMQx7 Casablanca machines. It is noteworthy that the first quantum computer (the 15-qubits Melbourne) was used during the earlier stages of the research, i.e., from 2018 and with the last experiments carried out in January of 2021, but was retired in the Summer of 2021 and is no longer available.

Before delving further into presenting the aforementioned quantum computers, it is important to introduce a way to describe and depict the basic characteristics of a quantum processor. To this end, the term *architectural graph* is used, defined as follows.

Definition 2.5 (Architectural graph)**.** *The architectural graph of a quantum processing unit (QPU) is the graph that depicts all the qubits and their connectivity within a QPU. The qubits are represented as nodes on the graph, whereas the available connections as the edges between the qubits (nodes).*

In addition to the above, it is essential to discuss the *quantum volume*, as introduced by Moll et al. [41]. This is an architecture-neutral figure of merit that characterises the performance of a quantum computer. It depends on the number of physical qubits in the machine, the number of qubits in the workspace, the depth and width of the circuit, as well as the average effective error rates of the two-qubit gates implemented in the quantum computer.

Definition 2.6 (Quantum volume [41])**.** *The quantum volume, namely V_Q, is a dimensionless metric used to characterise the performance of a quantum computer and can be calculated through the equation*

$$V_Q = \max_{n<N} \left(\min \left[n, \frac{1}{n \times \epsilon_{\mathit{eff}}(n)} \right]^2 \right) \tag{2.18}$$

where n is the size of the workspace necessary for the computation, N is the number of qubits within the quantum computer and $\epsilon_{\mathit{eff}}(n)$ is the average effective two-qubit gate error rates of the qubits that participate in the circuit, following the connectivity shown through the architectural graph of the QPU.

Starting up, the Melbourne machine had an Albatross QPU with 15 qubits. The connectivity of the qubits was provided by a total of 22 coplanar waveguide bus resonators,

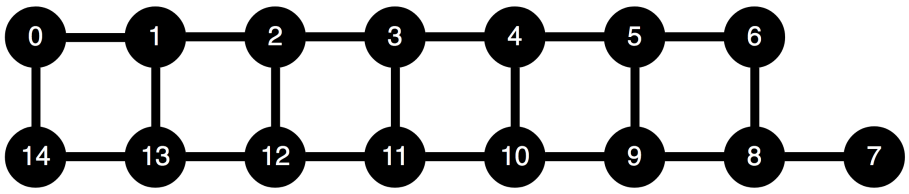
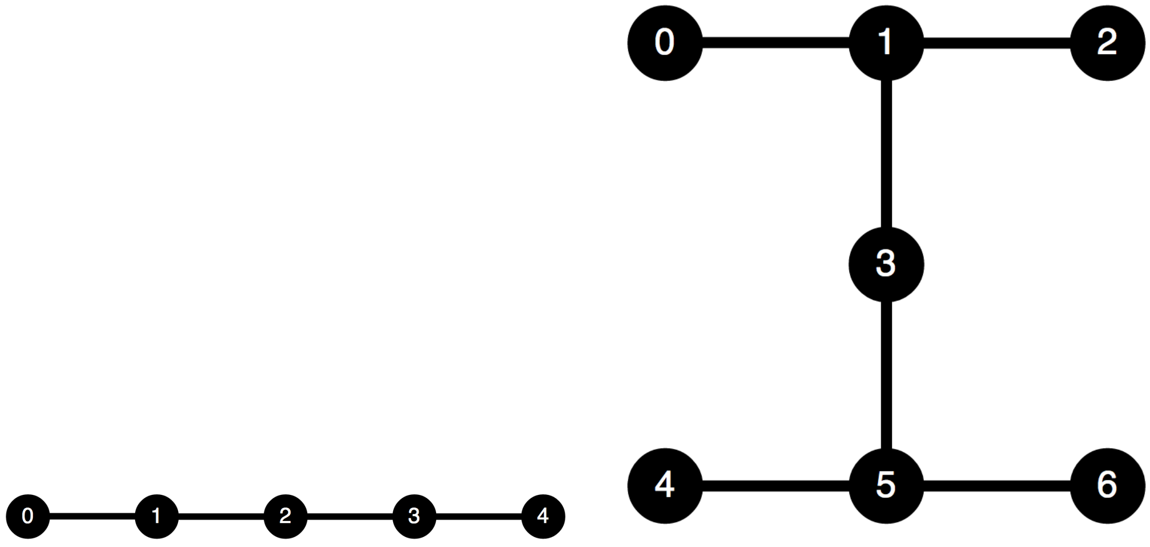


Fig. 2.2 Architectural graph of the IBMQx15 Melbourne computer.



(a) IBMQx5 Bogota and Santiago. (b) IBMQx7 Casablanca.

Fig. 2.3 Architectural graph of the (a) IBMQx5 Bogota and Santiago and (b) the IBMQx7 Casablanca quantum computers.

each connecting two qubits, following the architectural graph seen in the configuration of Figure 2.2. This computer had a quantum volume (QV) of $V_q = 8$. Following up, the Bogota and Santiago machines essentially have equivalent Falcon r4L quantum processors with five qubits and four connections. The architectural graph is depicted on Figure 2.3(a). Finally, the Casablanca computer possesses a Falcon r4H processor with seven qubits and six connections, as shown by the architectural graph of Figure 2.3(b). All three machines exhibit a quantum volume of $V_q = 32$.

Chapter 3

Discrete Time and Space Quantum Walks and Implementations

A quantum walk is the quantum mechanical analogue of a classical random walk [42]. This construct differs greatly to its classical analogue in terms of dynamical evolution. Whereas in a classical random walk the walker occupies definite states and the randomness arises due to stochastic transitions between states, the random behaviour of a quantum walk is governed by (i) quantum superpositions of quantised states, (ii) reversible unitary evolution and (iii) collapse of the wavefunction due to measurements of the quantum state at the end of the evolution.

During a quantum walk, the object which undergoes the evolution, here referred to as a walker and considered to be, for all intents and purposes, a quantum particle, sees its state evolved in a superposition. This gives the walker the ability to “seemingly” follow all possible paths, propagating quadratically further than a walker that is subjected to a classical random walk [8]. This quadratic increase in efficiency is proven mathematically in Section 3.1 and is also shown experimentally through simulations. Unfortunately, due to quantum noise, it is near impossible to demonstrate the speed-up offered by a quantum walk on real devices.

Quantum walks have the potential to speed up classical algorithms that are based on random walks [8, 43, 44]. There have been many systematic studies on this subject area and many of them can lead to further in-depth analysis of more advanced quantum algorithms, such as quantum Metropolis, quantum Markov chains or quantum Monte Carlo methods [45, 46, 15, 8, 47]. An early work from [25] proves that, in the context of quantum walks on graphs, the walker’s propagation in the quantum case is quadratically faster than the classical random walk. The efficiency of quantum walks has been exploited in various cases in order to construct quantum algorithms [48, 12] and speed up classical methods [49, 50], sometimes even exponentially [9, 51]. Quantum walks have also been realised in a number of physical systems, including photons [52–57], cold atoms [58, 59] and trapped ions [60].

This chapter focuses mainly on discrete-time and -space quantum walks (DTQW), as introduced in Section 1.2.1. As a starting point, a complete proof of the advantage provided from the quantisation of discrete-time random walks is shown. Following, two main circuit approaches for implementing discrete-time quantum walks are analysed, along with experiments and in-depth comparison. Finally, a novel work on various attempts to further speed up or alter quantum walk techniques is shown in Section 3.3.

The work outlined in Section 3.2 has been published in Physical Review A in January 2021 [61].

3.1 Quadratic Efficiency of Discrete-time Quantum Walks

This section showcases in a structured proof the quadratic advantage of quantum walks over classical random walks using the variance as a basis for comparison. The variance of a random walk essentially expresses the rate at which the walker propagates (or "spreads") through the graph. Hence, a larger variance means a faster or more efficient walk. The proof provided below is a combination of two different parts, first the Fourier analysis of the DTQW, as shown in [62], which introduces the tools that are then used for the derivation of the variance, shown in [21]. These two parts are put together to construct a theoretical approach for finding the variance of a quantum walk on an infinite, one-dimensional line. An important characteristic of quantum walks is their asymmetry, compared to the classical walk. This asymmetry derives from the effects of interference during the steps of the walk. The interference, in turn, arises from the fact that the amplitudes of the quantum states are complex numbers.

During a quantum walk, the walker has an extra degree of freedom that assists in their motion, called *chirality*. The chirality can take two values, *Left* ($|\mathrm{L}\rangle$) or *Right* ($|\mathrm{R}\rangle$) and describes the predisposition of the quantum walk to interference. Considering a particle that undergoes quantum walk, it will exhibit different behaviour depending on its initial spin (up $|\uparrow\rangle$ or down $|\downarrow\rangle$) or, equivalently, state ($|0\rangle$ or $|1\rangle$). In order to remove this bias, the quantum walk is initialised in an equal superposition of the two states using a Hadamard gate, H. Thus, depending on the initial state of the particle, the walk will be initialised as

$$H|\mathrm{L}\rangle = \frac{1}{\sqrt{2}}(|\mathrm{L}\rangle + |\mathrm{R}\rangle)$$
$$H|\mathrm{R}\rangle = \frac{1}{\sqrt{2}}(|\mathrm{L}\rangle - |\mathrm{R}\rangle)$$

It is noteworthy here how the amplitudes depend on each initial state, i.e, for $|\uparrow\rangle$ (or $|0\rangle$) the superposition undergoes addition whereas for $|\downarrow\rangle$ (or $|1\rangle$) subtraction. This difference in the amplitudes of the initial state comprises the effects of chirality.

The two component statevector of the amplitudes of the particle being at point n at time t is

$$\Psi(n,t) = \begin{pmatrix} \psi_{\mathrm{L}}(n,t) \\ \psi_{\mathrm{R}}(n,t) \end{pmatrix}, \tag{3.1}$$

with the upper component $\psi_{\mathrm{L}}(n,t)$ having chirality *Left* and the lower component $\psi_{\mathrm{R}}(n,t)$ having chirality *Right*. The dynamics of the wave function Ψ are given by the transformation

$$\begin{aligned}\Psi(n,t+1) &= \begin{pmatrix} 0 & 0 \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{pmatrix} \Psi(n-1,t) + \begin{pmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ 0 & 0 \end{pmatrix} \Psi(n+1,t) \\ &= M_+\Psi(n-1,t) + M_-\Psi(n+1,t).\end{aligned}$$

This transformation is unitary, since it is a composition of a unitary operator (namely H) and a reversible move/step to the left or to the right. Assuming that the particle starts at the origin with chirality left, the initial conditions are

$$\Psi(0,0) = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$$

which of course implies

$$\Psi(n,0) = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \quad \text{if } n \neq 0.$$

Fourier Analysis of the Discrete-time Quantum Walk

One of the better ways to map a continuous function to a discrete space is to use *Fourier Transforms* (FT) [62]. The spatial FT $\tilde{\Psi}(k,t)$ for $k \in [-\pi,\pi]$ of the wave function $\Psi(n,t)$ over $\mathbb{Z}$ is given by

$$\tilde{\Psi}(k,t) = \sum_n \Psi(n,t)\mathrm{e}^{\mathrm{i}kn} \tag{3.2}$$

where $\Psi(n,t) : \mathbb{Z} \mapsto \mathbb{C}$ maps from the discrete space $\mathbb{Z}$ to the continuous space $\mathbb{C}$ and is a complex valued function over the integers, with its FT being $\tilde{\Psi}(k,t) : [-\pi,\pi] \mapsto \mathbb{C}$.

From the dynamics of Ψ the following can be deduced about $\tilde{\Psi}$:

$$\begin{aligned}\tilde{\Psi}(k,t+1) &= \sum_n \left(M_+\Psi(n-1,t) + M_-\Psi(n+1,t)\right)\mathrm{e}^{\mathrm{i}kn} \\ &= \mathrm{e}^{\mathrm{i}k}M_+\sum_n \Psi(n-1,t)\mathrm{e}^{\mathrm{i}k(n-1)} + \mathrm{e}^{-\mathrm{i}k}M_-\sum_n \Psi(n+1,t)\mathrm{e}^{\mathrm{i}k(n+1)} \\ &= \left(\mathrm{e}^{\mathrm{i}k}M_+ + \mathrm{e}^{-\mathrm{i}k}M_-\right)\tilde{\Psi}(k,t)\end{aligned}$$

where it is important to prove the following

$$\begin{aligned}\tilde{\Psi}(k-1,t) &= \sum_n \Psi(n-1,t)e^{\mathrm{i}k(n-1)} \\ &= \sum_n \Psi(n-1,t)e^{\mathrm{i}kn}e^{-\mathrm{i}k} \\ &= e^{-\mathrm{i}k}\sum_n \Psi(n-1,t)e^{\mathrm{i}kn} \\ &= e^{-\mathrm{i}k}e^{\mathrm{i}k}\sum_n \Psi(n-1,t)e^{\mathrm{i}k(n-1)} \\ &= \sum_n \Psi(n-1,t)e^{\mathrm{i}kn} = \tilde{\Psi}(k,t)\end{aligned}$$

for the transition between the last two equalities. Thus

$$\tilde{\Psi}(k,t+1) = M_k\tilde{\Psi}(k,t) \tag{3.3}$$

where

$$\begin{aligned} M_k &= e^{\mathrm{i}k}M_+ + e^{-\mathrm{i}k}M_- \\ &= \frac{1}{\sqrt{2}}\begin{pmatrix} e^{-\mathrm{i}k} & e^{-\mathrm{i}k} \\ e^{\mathrm{i}k} & -e^{\mathrm{i}k} \end{pmatrix}. \end{aligned} \tag{3.4}$$

Note that one can also write $M_k = \Lambda_k U^{\intercal}$, where Λ_k is a diagonal matrix and $U^{\intercal}$ is the transpose of the unitary that acts on the chirality state (in the quantum walk case $U = H$, i.e., the Hadamard operator). Thus, M_k is unitary and the recurrence in the Fourier space is of the form

$$\tilde{\Psi}(k,t+1) = M_k\tilde{\Psi}(k,t) = M_k^t\tilde{\Psi}(k,0). \tag{3.5}$$

Additionally, M_k^t can be calculated by diagonalising the matrix M_k, which is readily done being a 2×2 matrix.

The next step is the eigendecomposition of M_k. If M_k has eigenvectors $(|\Phi_k^1\rangle, |\Phi_k^2\rangle)$ and eigenvalues $(\lambda_k^1, \lambda_k^2)$, one can write

$$M_k = \lambda_k^1 |\Phi_k^1\rangle\langle\Phi_k^1| + \lambda_k^2 |\Phi_k^2\rangle\langle\Phi_k^2|,$$

thus obtaining the evolution matrix as

$$M_k^t = \left(\lambda_k^1\right)^t |\Phi_k^1\rangle\langle\Phi_k^1| + \left(\lambda_k^2\right)^t |\Phi_k^2\rangle\langle\Phi_k^2|. \tag{3.6}$$

The eigenvalues of M_k can be found as $\lambda_k^1 = e^{-\mathrm{i}\omega_k}$ and $\lambda_k^2 = e^{\mathrm{i}(\pi+\omega_k)}$, where ω_k is the angle in $[-\pi/2, \pi/2]$ such that $\sin\omega_k = \frac{\sin k}{\sqrt{2}}$. The corresponding eigenvectors are:

$$\begin{aligned} \Phi_k^1 &= \frac{1}{\sqrt{2N(k)}}\begin{pmatrix} e^{-\mathrm{i}k} \\ e^{\mathrm{i}\omega_k} + e^{-\mathrm{i}k} \end{pmatrix} \\ \Phi_k^2 &= \frac{1}{\sqrt{2N(\pi-k)}}\begin{pmatrix} e^{-\mathrm{i}k} \\ -\sqrt{2}e^{-\mathrm{i}\omega_k} + e^{-\mathrm{i}k} \end{pmatrix} \end{aligned} \tag{3.7}$$

where $N(k) = (1+\cos^2 k) + \cos k\sqrt{1+\cos^2 k}$.

Considering the evolution of the particle, starting with chirality *Left* and in the Fourier basis $\tilde{\Psi}(k,0) = |0\rangle\ \forall k$, the two components of the wave function of equation (3.1) at time t are given by

$$\begin{aligned} \tilde{\psi}_{\mathrm{L}}(k,t) &= \frac{1}{2}\left(1 + \frac{\cos k}{\sqrt{1+\cos^2 k}}\right)e^{-\mathrm{i}\omega_k t} + \frac{(-1)^t}{2}\left(1 - \frac{\cos k}{\sqrt{1+\cos^2 k}}\right)e^{\mathrm{i}\omega_k t} \\ \tilde{\psi}_{\mathrm{R}}(k,t) &= \frac{\mathrm{i}e^{\mathrm{i}k}}{2\sqrt{1+\cos^2 k}}\left(e^{-\mathrm{i}\omega_k t} - (-1)^t e^{\mathrm{i}\omega_k t}\right). \end{aligned} \tag{3.8}$$

Inverting the FT allows returning to the real basis. This inversion is given by

$$\Psi(n,t) = \frac{1}{2\pi}\int_{-\pi}^{\pi} \tilde{\Psi}(k,t)e^{\mathrm{i}kn}\,\mathrm{d}k \tag{3.9}$$

Thus, the components of the wave function can now be described as:

$$\begin{aligned}\psi_{\mathrm{L}}(n,t) &= \frac{1+(-1)^{n+t}}{2}\int_{-\pi}^{\pi}\frac{\mathrm{d}k}{2\pi}\left(1+\frac{\cos k}{\sqrt{1+\cos^2 k}}\right)e^{-\mathrm{i}(\omega_k t+kn)}\\ \psi_{\mathrm{R}}(n,t) &= \frac{1+(-1)^{n+t}}{2}\int_{-\pi}^{\pi}\frac{\mathrm{d}k}{2\pi}\frac{e^{\mathrm{i}k}}{\sqrt{1+\cos^2 k}}e^{-\mathrm{i}(\omega_k t+kn)}\end{aligned} \tag{3.10}$$

In order to account for chirality right or $|\downarrow\rangle$, the operator M_k^t acting on the chirality state $|c\rangle = \{\uparrow,\downarrow\}$[1] will now be written as

$$M_k^t\,|c\rangle = \left((\lambda_k^1)^t\,\langle\Phi_k^1|c\rangle\right)|\Phi_k^1\rangle + \left((\lambda_k^2)^t\,\langle\Phi_k^2|c\rangle\right)|\Phi_k^2\rangle\,.$$

The basis in FT can be expressed as

$$|\Psi_{n,t}^c\rangle = U^t|\psi_0^c\rangle = U^t(|0\rangle\otimes|c\rangle) = \frac{1}{2\pi}\int_{-\pi}^{\pi}\tilde{\Psi}(k,t)\otimes M_k^t|c\rangle\mathrm{d}k$$

and consequently

$$|\psi_t^c\rangle = \sum_n |n\rangle\otimes\left[A_c^t(n)|\uparrow\rangle + B_c^t(n)|\downarrow\rangle\right]$$

with the following equalities:

$$\begin{aligned}A_\uparrow^t(n) &= \frac{1+(-1)^{t+n}}{2}\left[\alpha^t(n)+\beta^t(n)\right]\\ A_\downarrow^t(n) &= \frac{1+(-1)^{t+n}}{2}\left[\beta^t(n)-\gamma^t(n)\right]\\ B_\uparrow^t(n) &= \frac{1+(-1)^{t+n}}{2}\left[\beta^t(n)+\gamma^t(n)\right]\\ B_\downarrow^t(n) &= \frac{1+(-1)^{t+n}}{2}\left[\alpha^t(n)-\beta^t(n)\right]\end{aligned} \tag{3.11}$$

and

$$\begin{aligned}\alpha^t(n) &= \int_{\pi}^{\pi}\frac{\mathrm{d}k}{2\pi}\mathrm{e}^{\mathrm{i}(kn-t\omega_k)}\\ \beta^t(n) &= \int_{-\pi}^{\pi}\frac{\mathrm{d}k}{2\pi}\frac{\cos k}{\sqrt{1+\cos^2 k}}\mathrm{e}^{\mathrm{i}(kn-t\omega_k)}\\ \gamma^t(n) &= \int_{-\pi}^{\pi}\frac{\mathrm{d}k}{2\pi}\frac{\sin k}{\sqrt{1+\cos^2 k}}\mathrm{e}^{\mathrm{i}(kn-t\omega_k)}\end{aligned} \tag{3.12}$$

Derivation of the Variance

To find the variance of the walk, the initial state is generalised in a similar fashion to [21]. Thus, starting with initial state:

$$|\psi_0\rangle = |0\rangle\otimes\left[\sqrt{q}\,|\uparrow\rangle + \sqrt{1-q}\mathrm{e}^{\mathrm{i}\sigma}\,|\downarrow\rangle\right] \tag{3.13}$$

with $q \in [0,1]$ and $\sigma \in \mathbb{R}$, the quantum state after t steps will be:

$$|\psi_t\rangle = U^t |\psi_0\rangle = \sum_n |n\rangle \otimes \Big\{ \Big[\sqrt{q}A^t_{\uparrow}(n) + \sqrt{1-q}e^{i\sigma}A^t_{\downarrow}(n)\Big] |{\uparrow}\rangle \\ + \Big[\sqrt{q}B^t_{\uparrow}(n) + \sqrt{1-q}w^{i\sigma}B^t_{\downarrow}(n)\Big] |{\downarrow}\rangle \Big\}$$

The above can be further simplified by considering the modular behaviour (see Definition 1.2) of the quantum walk. The probability for the walker to be at position n after t steps can be calculated as:

$$p^t(n) = \Big|\sqrt{q}A^t_{\uparrow}(n) + \sqrt{1-q}e^{i\sigma}A^t_{\downarrow}(n)\Big|^2 + \Big|\sqrt{q}B^t_{\uparrow}(n) + \sqrt{1-q}e^{i\sigma}B^t_{\downarrow}(n)\Big|^2. \tag{3.14}$$

The integral parts of the matrices $A^n_{\uparrow}, A^n_{\downarrow}, B^n_{\uparrow}, B^n_{\uparrow}$ are mostly concentrated in the interval $[-t/\sqrt{2}, t/\sqrt{2}]$ and they quickly decrease beyond its bounding values. This can be shown by the method of stationary phase, described in Appendix D. Doing the correct approximations can lead to the probability $p^t(n)$ of equation (3.1) as an oscillation around the function

$$P^t(n) = \frac{2t}{\pi(t-n)\sqrt{t^2-2n^2}}$$

where the vanishing part coming from the modularity property can be dropped.

The function $P^t(n)$ allows to approximately evaluate the averages of position n-dependent functions in the t-th step of the quantum walk as

$$\Big\langle f^t(n) \Big\rangle \simeq \frac{1}{2}\int_{-\frac{t}{\sqrt{2}}}^{\frac{t}{\sqrt{2}}} f^t(n)P^t(n)\,\mathrm{d}n \tag{3.15}$$

Thus, from equation (3.15) follows that

$$\begin{aligned} \langle n \rangle &\simeq \frac{1}{2}\int_{\frac{t}{\sqrt{2}}}^{\frac{t}{\sqrt{2}}} n\frac{2t}{\pi(t-n)\sqrt{t^2-2n^2}}\mathrm{d}n = -t\frac{\sqrt{2}-1}{\sqrt{2}} \\ \langle n^2 \rangle &\simeq \int_{-\frac{t}{\sqrt{2}}}^{\frac{t}{\sqrt{2}}} n^2\frac{2t}{\pi(t-n)\sqrt{t^2-2n^2}}\mathrm{d}n = t^2\frac{\sqrt{2}-1}{\sqrt{2}} \end{aligned} \tag{3.16}$$

and finally, using equations (3.16), the variance of the quantum walk, σ^2_{qw} , can be found as

$$\begin{aligned} \sigma^2_{\text{qw}} &= \langle n^2 \rangle - \langle n \rangle^2 = t^2\left(\frac{\sqrt{2}-1}{\sqrt{2}}\right)^2 - t^2\frac{\sqrt{2}-1}{\sqrt{2}} \\ &= t^2\left(\frac{\sqrt{2}-1}{\sqrt{2}}\right)\left[\left(1-\frac{\sqrt{2}-1}{\sqrt{2}}\right)\right] \Rightarrow \\ \sigma^2_{\text{qw}} &\approx 0.207 \times t^2 \end{aligned} \tag{3.17}$$

This shows that the variance of the quantum walk is proportional to t^2 (i.e., $\sigma^2_{\text{qw}} \sim t^2$) or otherwise the standard deviation is $\sigma_{\text{qw}} \sim t$. Thus, the standard deviation of the quantum walk grows near-linearly with time, quadratically faster than a classical random walk, where $\sigma_{\text{cw}} \sim \sqrt{t}$.

3.2 Circuits and Circuit Approaches to Discrete-time Quantum Walks

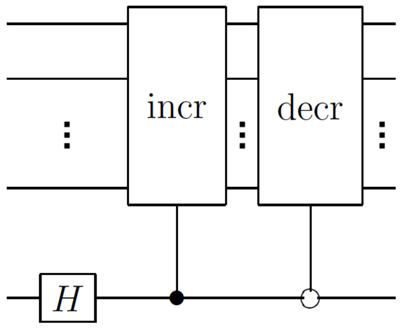
This section discusses two circuit approaches used to implement quantum walks both as simulations and on real quantum computers. Additionally, it offers an analysis and comparison between the two implementations and the associated characteristics (as published in [61]). The quantum walks considered in this section will be run on cycle graphs of size N, or otherwise, N-cycles where for simplicity N is always taken as an exact power of 2. This means that there are N possible positions for the walker and the graph can be encoded on the quantum processing unit using $n = \log N$ qubits. For all the experiments during the next sections, the IBMQ Qiskit tools and quantum computers will be utilised [63, 64]. More specifically, the experiments are executed on the IBMQ 15-qubit Melbourne machine.

Due to the importance and plentiful research pouring into quantum walks and their use, various implementations of this algorithm exist. Within the reviewed research the focus is concentrated towards two approaches to implementing quantum walks. The first one uses generalised inverter gates, i.e., `CNOT` gates controlled by more than one qubits. The second implementation effectively replaces those operations with a sequence of rotation and phase gates.

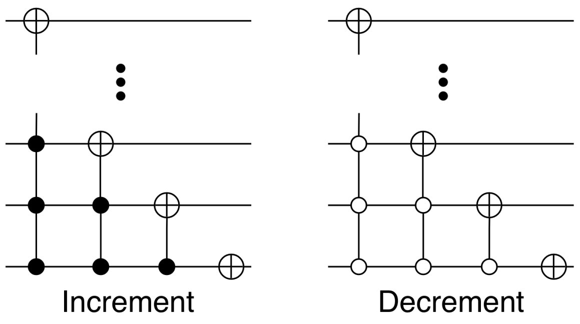
These two approaches offer distinct characteristics and opposite advantages and disadvantages. The generalised inverters approach is very efficient, in terms of necessary resources and quantum volume. The rotational approach is less efficient in terms of gates but more efficient in terms of qubits, while also being universally applicable, meaning that it can be used to decompose arbitrary scaling unitary gates to a sequence of rotation and phase operations. Hence, it is deemed worthwhile to subject the two implementations through extensive analysis.

3.2.1 The Generalised Inverters Approach

The generalised inverters approach to implementing quantum walks is based on the work of [19]. As mentioned above, generalised controlled operations are those whose effect on a *target qubit* is controlled by more than one *control qubit*. In this case, the operation in question will occur if and only if all the control qubits are in state $|1\rangle$. A well known example of a generalised operation, and one that is used extensively for this approach, is the three-qubit Toffoli gate in which the target qubit will be inverted only if both the control qubits are in state $|1\rangle$. Direct implementations of the Toffoli gate do not currently exist on quantum hardware, but they can be easily decomposed to well-known single- and two-qubit operations [65, 66].



(a) Circuit schematic.



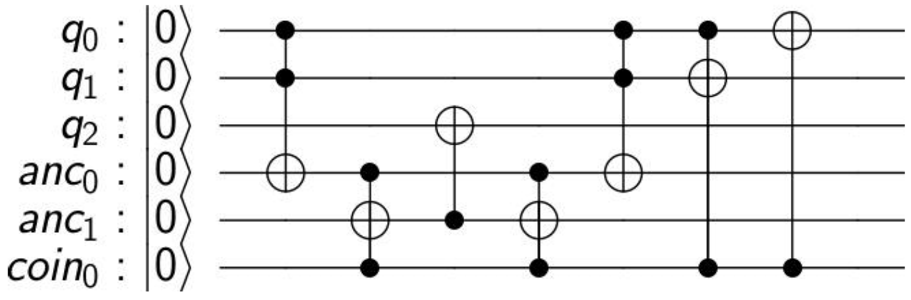
(b) High-level circuit implementation.

Fig. 3.1 (a) Implementation of one step for the quantum walk of a particle. (b) Quantum circuits for increment and decrement operations. A filled control circle means that the control qubits have to be in state $|1\rangle$ in order for the operation to occur. An empty control circle means they have to be in state $|0\rangle$.

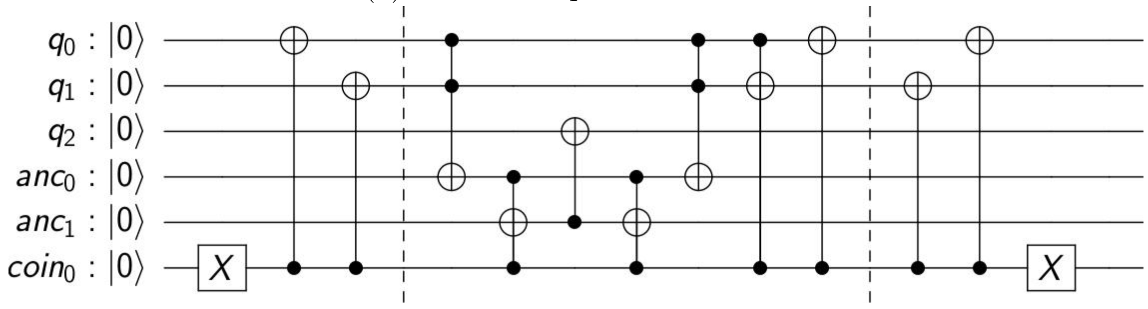
One step of the quantum walk, following equation (1.7), can be broken down to two operations: (i) an *increment* operation (or function) that implements the increase of the quantum state, i.e., S^+, and (ii) a *decrement* function that implements the decrease of the state of the walker, i.e., S^-. These two functions can be executed in a superposition by utilising the Hadamard coin, thus implementing the dynamics of a discrete-time quantum walk.

The generalised `CNOT` gates can be used to construct quantum circuits that implement the increment and decrement functions. As shown by [19], those two operations can be realised with a single quantum circuit, but with opposite control logic. In other words, one function will be controlled by $|1\rangle$ (the increment is chosen here) whereas the other by $|0\rangle$ (decrement). Figure 3.1(a) shows the schematic of the circuit executing one iteration of the quantum walk for a state space of arbitrary size. The general high-level circuit implementation of the increment and decrement circuits is shown in Figure 3.1(b). The realisation of the individual quantum circuits in Figure 3.1(b) for a small quantum walk on a $N = 8$-cycle using elementary and Toffoli gates is shown in Figure 3.2(a-b).

Both the increment and decrement circuits of the last two figures look more complicated than their respective schematic in Figure 3.1(b). The reason is the lack of direct implementation of any generalised inverters other than the Toffoli gate on Qiskit. Any inverter gate with more than two control qubits (i.e., bigger than the Toffoli gate) requires intermediate computations stored in ancilla qubits. This leads to a significant increase of the workspace (i.e., the number of qubits needed for the computation) that grows with the size of the state space. Precisely, a generalised `CNOT` gate with n_c control qubits requires additional n_{c-1} ancilla qubits for the implementation (refer to Appendix C). For example, considering the 15-qubit Melbourne machine (the biggest size of public machine from IBMQ at the time of this research), one can implement a quantum walk on a cycle with at most $N = 2^7 = 128$ states, i.e., a seven-qubit state space. In case of the larger quantum walk on $N = 2^8 = 256$ states, i.e., with an eight-qubit state space ($n = 8$), the largest generalised inverter gate will have eight ($n_c = 8$) control qubits (including



(a) Increment quantum circuit.



(b) Decrement quantum circuit.

Fig. 3.2 (a) Increment circuit. The three qubit $q_n, n = 0, 1, 2$ register is the state space of the quantum walk, the coin register $coin_0$ represents the Hadamard coin. The ancilla qubits used for the computation are anc_0 and anc_1. (b) Decrement circuit. Important here is the need to invert all the control qubits (including the coin) at the start of the computation and uncompute them at the end.

the coin), meaning it will need seven ($n_c - 1 = 7$) ancilla qubits for the computation. Thus, $n + (n_c - 1) = 15$ qubits for the state space and the ancilla, plus one qubit for the Hadamard coin, gives 16 qubits, one more than the capacity of the quantum computer.

For the implementation of a quantum walk using this approach there is the need for a qubit register with $n = \log N$ qubits, an ancilla register with $\log N - 1$ qubits and one additional qubit for the quantum coin. This means that the generalised `CNOT` approach requires $2 \times \log N$ qubits to be implemented. Proposition 3.1 expresses the gate complexity of the quantum circuit in terms of gate requirements.

Proposition 3.1 (Inverters approach gate complexity). *The number of gates that participate in the generalised inverter implementation of the quantum walk circuit increases polylogarithmically with the size of the state space, N, as $\mathcal{O}(\log^2 N)$.*

Proof. For a state space $N \geq 8$, any gate that needs more than two control qubits is expanded to a network with ancilla qubits. The number of gates necessary for this expansion can be expressed as $2\sum_{n_c=3}^{\log N}(2n_c - 1)$, where n_c is the number of control qubits necessary for each operation. The additional gates needed will be the inverters with two or fewer control qubits and the Hadamard gate. For a state space of $N < 8$ there will be no operations with more than two control qubits and the number of gates will be simply calculated by the inverters and the Hadamard gate.

Thus, the number of gates for the generalised inverter implementation can be expressed as

$$\nu_{\mathrm{c}} = \begin{cases} 2\sum_{n_c=3}^{\log N}(2n_c - 1) + 2\log N + 5, & \text{if } N \geq 8 \\ 2\log N + 5, & \text{if } 2 \leq N < 8 \end{cases} \tag{3.18}$$

From the above equation one can see that the sum $\sum_{n_c=3}^{\log N}(2n_c - 1)$ provides the dominant growth rate. It is found that

$$\begin{aligned} \sum_{n_c=3}^{\log N}(2n_c - 1) &= \sum_{n_c=1}^{\log N}(2n_c - 1) - \sum_{n_c=1}^{2}(2n_c - 1) \\ &= \log^2 N - 4, \end{aligned}$$

as the sum $\sum_{n_c=1}^{\log N}(2n_c - 1) = \log^2 N$ is the known sum of the first $\log N$ odd natural numbers and $\sum_{n_c=1}^{2}(2n_c - 1) = 4$. Thus, the number of gates increases with the size of the state space, N, as $\mathcal{O}(\log^2 N)$. □

Finally, it is noteworthy that the above proposition can be formulated in terms of the number of qubits, n, necessary to represent the state space of the quantum walk, as $n = \log N$. With this in mind, the number of gates necessary to implement the circuit for the generalised inverters approach increases quadraticaly with the number of qubits that represent the state space of the quantum walk as $\mathcal{O}(n^2) \equiv \mathcal{O}(\log^2 N)$.

3.2.2 The Rotational Approach

The second approach to implementing quantum walks examined within this thesis uses a set of rotation and phase operations [67, 68] to implement the increment and decrement functions. These operations are more complex than the generalised inverters, but they allow the implementation of a quantum walk without the need for an ancilla register, thus lowering the vast increase in computational resources resulting from adding qubits to the workspace. Referring back to the bounded size of the quantum walk, the 15-qubit Melbourne machine used for the experiments is able to run a quantum walk with rotations on a cycle with $N = 2^{14} = 16{,}384$ positions, i.e., an $n = 14$-qubit state space (plus one qubit for the Hadamard coin), a significant rise compared to the 128 states of the generalised `CNOT` approach.

Another benefit of the rotational approach holds when simulating quantum walks on classical machines. As the size of the state space increases exponentially with the number of qubits, classical computers very quickly start struggling to cope with the size of the workspace. The rotational implementation offers a way around this problem by lowering the number of qubits necessary to implement the quantum walk. Finally, this approach to implementing quantum circuits via rotations around the basis states is universally applicable, i.e., it can be used to translate any arbitrary unitary operator to a sequence of rotation and phase gates, offering the advantages and disadvantages discussed in this section

In order to engineer a quantum circuit that implements the rotational approach to quantum walks, the two following lemmas are used, first introduced and proven by [67].

Lemma 3.1 (Unitary rotational decomposition [67]). *For any unitary operator W there exist operators Φ, A, B and C such that $ABC = I$ and $\Phi AXBXC = W$, where Φ is a phase operator of the form $\Phi = e^{i\delta} \times I$ with $\delta \in \mathbb{R}$, X is the Pauli-X and I the identity matrix.*

What one can learn from Lemma 3.1 is that any unitary operator and, for the case examined here, a `NOT` gate, can be expressed as a sequence of operators $\Phi AXBXC$. This result acts as a stepping stone for generalising the deconstruction of a higher-dimensional inverter gate. The existence of the operator Φ compensates for the fact that the `NOT` gate is not a special SU(2) unitary, i.e., it does not have determinant 1. Thus, efforts can be narrowed down to finding the appropriate Φ, A, B and C operators that suit specific needs of the implementation [61].

The natural first step is to decompose the Toffoli gate, denoted X_{cc}, the simplest generalised `CNOT` gate, that needs two control qubits. The first rotation gate needed can be found through the unitary matrix $R_y(\theta)$ defined as

$$R_y(\theta) = \begin{pmatrix} \cos\theta/2 & -\sin\theta/2 \\ \sin\theta/2 & \cos\theta/2 \end{pmatrix} . \tag{3.19}$$

For the Toffoli case, it is required that $\theta = \pi/2$, giving from equation (3.19) the operator

$$R_y(\pi/2) = \begin{pmatrix} \frac{\sqrt{2}}{2} & -\frac{\sqrt{2}}{2} \\ \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} \end{pmatrix} .$$

The next operation is expressed by the unitary operator $R_z(\phi)$, given as

$$R_z(\phi) = \begin{pmatrix} e^{i\phi/2} & 0 \\ 0 & e^{-i\phi/2} \end{pmatrix} . \tag{3.20}$$

Similarly to the rotation operators, assigning $\phi = \pi/2$, results to the operator

$$R_z(\pi/2) = \begin{pmatrix} \frac{\sqrt{2}}{2} - \frac{\sqrt{2}}{2}i & 0 \\ 0 & \frac{\sqrt{2}}{2} + \frac{\sqrt{2}}{2}i \end{pmatrix} .$$

Finally, since the inverter gate is not a special unitary, there is the need for an additional phase gate $\Phi(\delta)$ defined as

$$\Phi(\delta) = \begin{pmatrix} e^{i\delta} & 0 \\ 0 & e^{i\delta} \end{pmatrix} , \tag{3.21}$$

with $\delta = -\pi/2$ identified for this case. Thus, the relevant operator is

$$\Phi(-\pi/2) = \begin{pmatrix} -i & 0 \\ 0 & -i \end{pmatrix} .$$

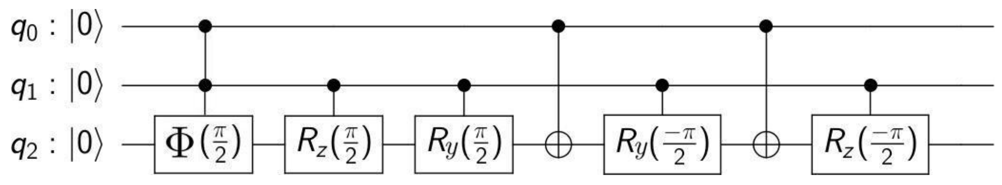


Fig. 3.3 Quantum circuit implementing a Toffoli gate (X_{cc}) using conditioned rotations.

Considering the above, the rotational decomposition of an inverter gate, X, can now be written as

$$X \equiv \Phi(\pi/2)R_z(\pi/2)R_y(\pi/2)XR_y(-\pi/2)XR_z(-\pi/2) \tag{3.22}$$

where $A = R_z(\pi/2)R_y(\pi/2)$, $B = R_y(-\pi/2)$ and $C = R_z(-\pi/2)$. This can be simply proven mathematically, as shown below.

Proof. Decomposing a 2×2 inverter gate, X, to a sequence of rotation and phase gates of the form presented in equation (3.22) is shown to be true via simple matrix operations.

$$\begin{aligned}
X &= \Phi(\pi/2)R_z(\pi/2)R_y(\pi/2)\sigma_x R_y(-\pi/2)R_z(0)\sigma_x R_z(-\pi/2) \\
&= \begin{pmatrix} -i & 0 \\ 0 & -i \end{pmatrix} \begin{pmatrix} \frac{\sqrt{2}}{2} - \frac{\sqrt{2}}{2}i & 0 \\ 0 & \frac{\sqrt{2}}{2} + \frac{\sqrt{2}}{2}i \end{pmatrix} \begin{pmatrix} \frac{\sqrt{2}}{2} & -\frac{\sqrt{2}}{2} \\ \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} \end{pmatrix} \\
&\times \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} \\ -\frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} \frac{\sqrt{2}}{2} + \frac{\sqrt{2}}{2}i & 0 \\ 0 & \frac{\sqrt{2}}{2} - \frac{\sqrt{2}}{2}i \end{pmatrix} \\
&= \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}
\end{aligned}$$

where the matrix expansions for $\theta, \phi = -\pi/2$ are easy to deduce from the relevant equations. □

By modifying the operators to accommodate for the right matrix dimensions, the Toffoli gate can be decomposed to a sequence of controlled rotation, phase and `NOT` gates, as shown in Figure 3.3.

The next step is to generalise this quantum circuit so that it can accommodate more than two control qubits, i.e., create a generalised `CNOT` using rotations. In order to do this, another lemma from [67] can be used. In this context, the need arises to introduce the notation $\wedge_{n-1}(U)$ as used by [67, 68]. For any unitary matrix $U = \begin{pmatrix} u_{00} & u_{01} \\ u_{10} & u_{11} \end{pmatrix}$ and $m \in \{0, 1, 2, \dots\}$, the $(m+1)$-bit (2^{m+1}-dimensional) operator $\wedge_m(U)$ is defined as

$$\wedge_m(U)(|x_1, \dots, x_m, y\rangle) = \begin{cases} u_{y0}\,|x_1, \dots x_m, 0\rangle + u_{y1}\,|x_1, \dots, x_n, 1\rangle & \text{if } \wedge_{k=1}^{m} x_k = 1 \\ |x_1, \dots, x_m, y\rangle & \text{if } \wedge_{k=1}^{m} x_k = 0 \end{cases}$$

where $\wedge_k$ denotes the `AND` operation of the relevant k values. Thus, the second lemma necessary to define the rotational approach is as follows.

Lemma 3.2 (Generalised unitary rotational decomposition [67]). *For any unitary W, a $\wedge_{n-1}(W)$ gate can be simulated by a network of rotation and phase operators, as shown in Figure 3.4, with Φ, A, B and C as in Lemma 3.1.*

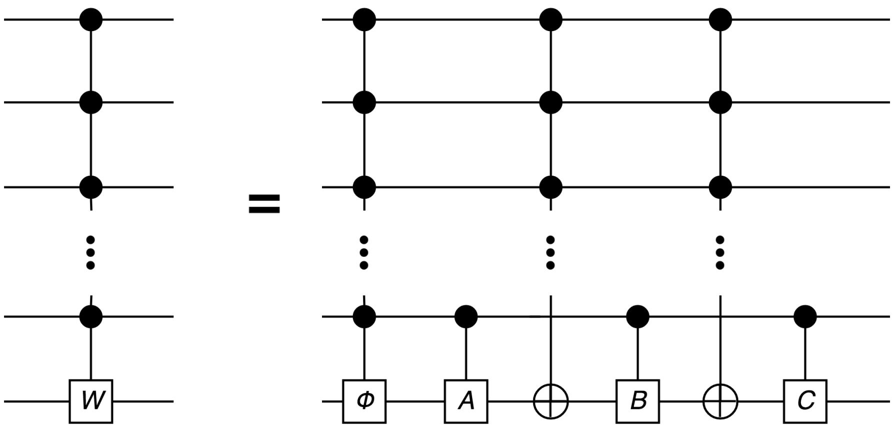


Fig. 3.4 Generalised rotational network that implements a unitary controlled by an arbitrary number of control qubits.

Lemma 3.2 describes a way to expand any generalised unitary with an arbitrary number m of control qubits to a network of controlled rotations and generalised `CNOT` gates of the form $\Phi AXBXC$. It is easy to see that, if $W = X$, one can iteratively expand each one of the generalised `CNOT` gates to such a network. The expansion will stop when the generalised inverter gates end up being regular Toffoli gates. After the transformation of the initial approach to rotation operations, the 2×2 operator W applied to the target qubit is the regular inverter

$$W = X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \tag{3.23}$$

with the dimensions of the matrix representation adjusted according to the dimensionality of the workspace.

The above work derives a way to produce a quantum circuit that implements a generalised `CNOT` gate with an arbitrary number of control qubits without depending on the use of any ancilla qubits. This logic can be applied to any unitary operator [67, 68].

The next step is to integrate this implementation to the increment and decrement circuits that constitute the quantum walk. The generalised `CNOT` gates are substituted with a network of the form described in Lemma 3.2. Any `CNOT` or inverter `X` gates remain the same, as do the Toffoli gates, due to well known and easily implementable decompositions, as aforementioned. A visualisation of an increment quantum circuit on four qubits is shown in Figure 3.5. The decrement circuit will follow similar logic with the difference that all the control qubits have to be inverted at the start of the function. The two sub-circuits implementing the increment and decrement functions can then be sequentially applied to implement one step of the quantum walk, while controlled by the Hadamard coin.

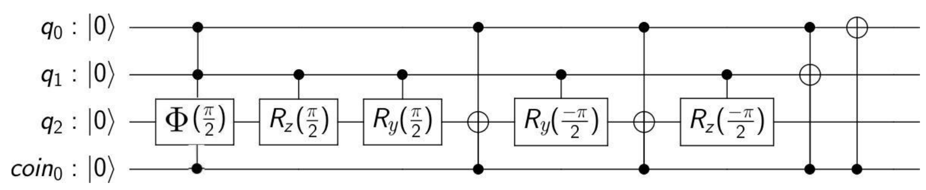


Fig. 3.5 Rotational implementation of an increment circuit for a three qubit state space and one qubit coin.

Unlike the generalised inverter implementation, the rotational approach eliminates the need for an ancilla register to carry out the quantum walk. Thus, it only requires an $n = \log N$ qubit register to represent the state space and one additional qubit for the quantum coin, i.e., $\log N + 1$ qubits. The complexity of this approach in terms of quantum gates is defined by the following proposition.

Proposition 3.2 (Rotational approach gate complexity)**.** *The number of gates that participate in the rotational implementation of the quantum walk circuit increases linearly with the size of the state space, N, as $\mathcal{O}(N)$.*

Proof. For $N \geq 8$ any operation with more than three control qubits will be expanded according to the network of Lemma 3.2. It is evident that every inverter as in Figure 3.4 will need to be expanded to a rotational network until all gates need only two control qubits. This leads to a number of $\sum_{j=3}^{\log N}\left[10\sum_{n_c=j}^{\log N}(2^{n_c-j})\right]$ gates before the last step as well as $\sum_{j=3}^{\log N}(2^{2-j+\log N})$ gates on the last step of the expansion. For operations with two or less control qubits, there is no need for rotations or expansions.

For $N < 8$ there will be no operations with more than two control qubits and the circuit will not need a rotational approach.

The number of gates for the rotational implementation, ν_r, can be expressed as

$$\nu_r = \begin{cases} \sum_{j=3}^{\log N}\left[(2^{2-j+\log N}) + 10\sum_{n_c=j}^{\log N}(2^{n_c-j})\right] + 2\log N + 5, & \text{if } N \geq 8 \\ 2\log N + 5, & \text{if } 2 \leq N < 8 \end{cases} \tag{3.24}$$

Equation (3.24) shows that again the sum provides the dominant growth. In this case it represents the well known sum of a geometric progression, where the largest growth would be given by $2^{\log N} = N$. Thus, the number of gates increases linearly with the size of the state space, N, as $\mathcal{O}(N)$. □

Thus, it is evident that, whereas the rotational approach offers a reduction to the number of qubits necessary for the computation, with all the advantages this entails, it is far less efficient in terms of necessary gates, as the number of gates needed in the quantum circuit increases exponentially faster than in the generalised inverters case. This is also visible when considering the complexity of the rotational circuit in terms of the number of qubits, $n = \log N$, necessary to represent the state space of the quantum walk. In a similar fashion to the inverters circuit, the number of gates needed for the rotational implementation increases with the number of qubits, n, as $\mathcal{O}(2^n) \equiv \mathcal{O}(N)$. As a reminder, the complexity of the generalised inverters circuit is $\mathcal{O}(n^2)$.

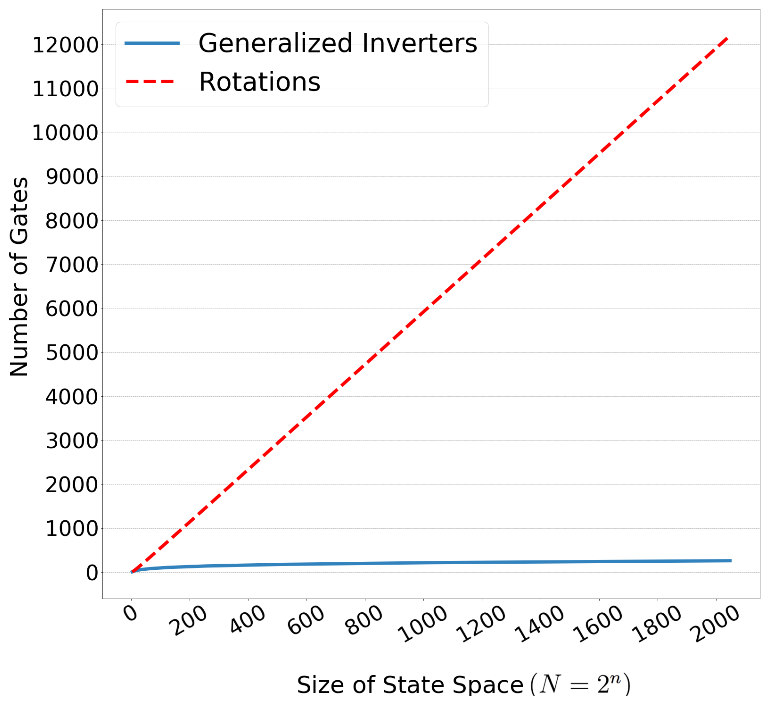


Fig. 3.6 Comparison between the generalised inverter and rotational circuits complexity. The size of the state space, N, is the number of states in the graph, or otherwise, the number of states that can be represented by the qubits n as $n = \log N$.

A visual comparison between the increase in the number of gates necessary for the implementation of both the inverters and rotational approaches as a function of the size of the state space is given in Figure 3.6.

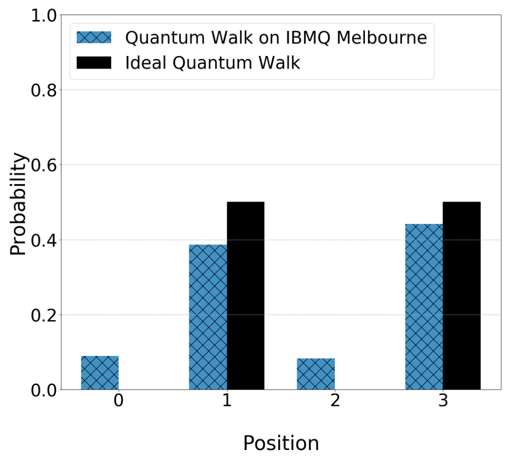
3.2.3 Results and Comparison of Quantum Walk Implementations

Following the analysis of the two approaches defined in Sections 3.2.1 and 3.2.2, this section presents the experimental results acquired from running quantum walks with the engineered quantum circuits, both as an ideal simulation (i.e., in the absence of noise) and on the IBM quantum hardware, more specifically, the IBMQ 15-qubit Melbourne machine.

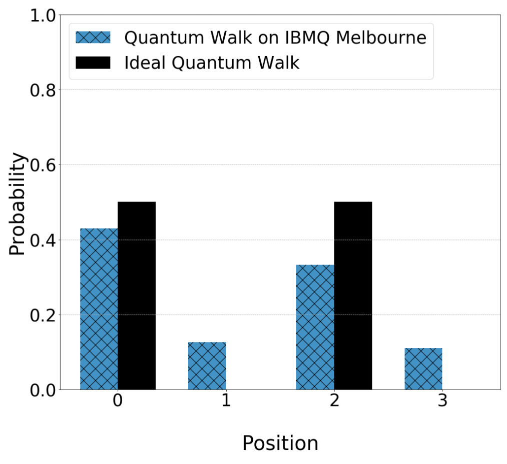
Throughout the experimental process, the quantum walk is initialised on state $|0\rangle^{\otimes n}$, where $n = \log N$ is the number of qubits representing the state space of the quantum walk of size N. The state space of the quantum walk is taken to be $N = 4$ and $N = 8$. The results of the simulations and experiments for both approaches on two and three qubit states are given together (for better comparison) in Figures 3.7 and 3.8 respectively.

Experiments and Results on Noise-free Simulator

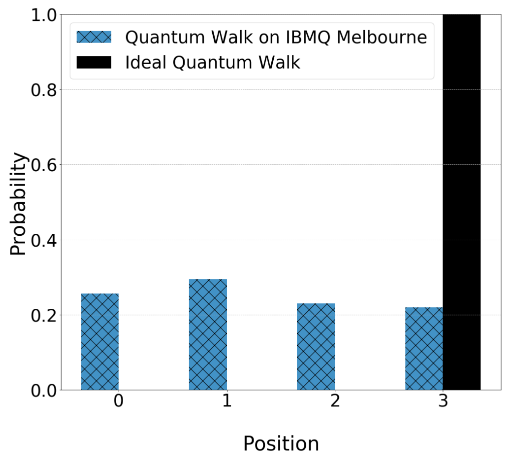
The Qiskit simulator [63] is used to implement the quantum circuits and execute the noise-free simulations. The simulations are run on a MacBook Pro 2017 computer with a 2.3 GHz Intel Core `i5` processor and 16 GB of memory.



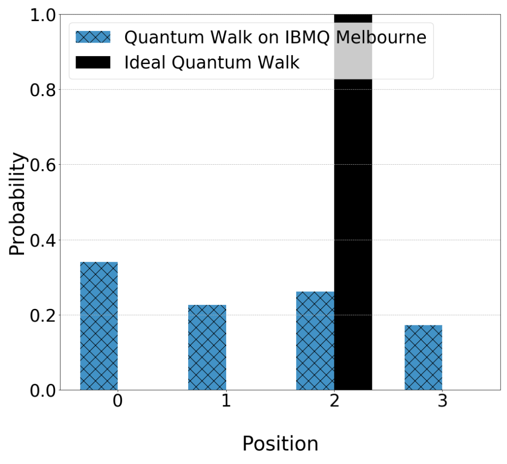
(a) One-step of the quantum walk.



(b) Two-steps of the quantum walk.



(c) Three-steps of the quantum walk.

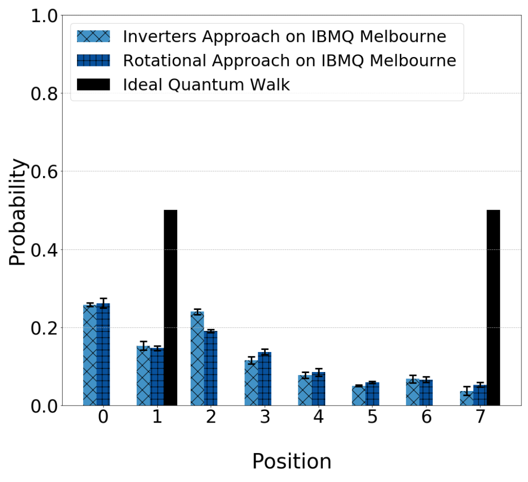


(d) Four-steps of the quantum walk.

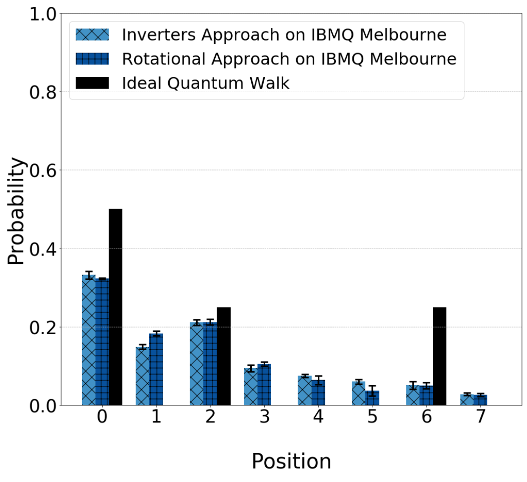
Fig. 3.7 Probability distributions of two-qubit quantum walks on the IBMQ Melbourne computer (crossed bar) and on an ideal simulator (solid bar) for (a) one step, (b) two steps, (c) three steps and (d) four steps, with generalised inverters approach. Experiments with the rotational approach are not needed, as the circuits are identical. Errors calculated with 95% confidence intervals are smaller than 10^{-3}, hence are not displayed.

There are three important points to observe here: (i) the asymmetry of the resulting probability distributions, (ii) the modular behaviour and (iii) the variance of the quantum walk. Those represent properties of discrete-time quantum walks, as showcased in Section 1.2.1. First of all, the effects of the asymmetry can be seen as the imbalance in the ideal probability distribution. A good example comes from examining the ideal distribution resulting from three-steps of the quantum walk, as showcased in Figure 3.8(c) (black bar), where state $|7\rangle$ appears with higher probability than the rest of the states.

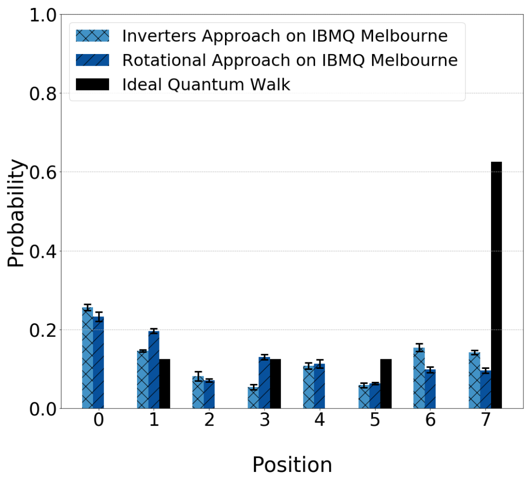
The second point involves the modularity of the quantum walk, as this property was introduced in Definition 1.2. Having initialised the walker on state $|0\rangle$ (considered an even state) and evolving for an odd or even number of steps, it is predicted that the measurement outcome will be an odd or even one respectively. Indeed that is the case,



(a) One-step of the quantum walk.



(b) Two-steps of the quantum walk.



(c) Three-steps of the quantum walk.

Fig. 3.8 Probability distributions of quantum walks on three qubits for (a) one step, (b) two steps and (c) three steps using the IBMQ Melbourne machine for the generalised inverter (crossed bar) and rotational (tiled bar) approach and an ideal simulator (solid bar), on which both approaches are identical. Error bars are calculated with 95% confidence intervals.

with the outcomes measured also satisfying that only $N/2$ states are observed. This stands true for both approaches.

The final point refers to the variance of the quantum walk. It has been proven that Markov chains show a quadratic increase in efficiency compared to their classical analogues [8]. As also theoretically shown in Section 3.1, the variance, σ^2_{qw}, as a function of the coin flips, t, can be calculated from equation (3.17). By computing the simulated quantum walk variance one can verify this quadratic tendency for both implementations, as depicted in Figure 1.5, concluding that both the quantum circuits are likely to be correct implementations of a quantum walk.

Finally, the simulation runtimes for different number of qubits on an 8-cycle for the two approaches are presented in Figure 3.9. It is visible that the runtime of the quantum walk circuit increases exponentially with the number of qubits. For the generalised inverters approach, the classical machine is unable to simulate the circuit for $n > 16$ due to lack

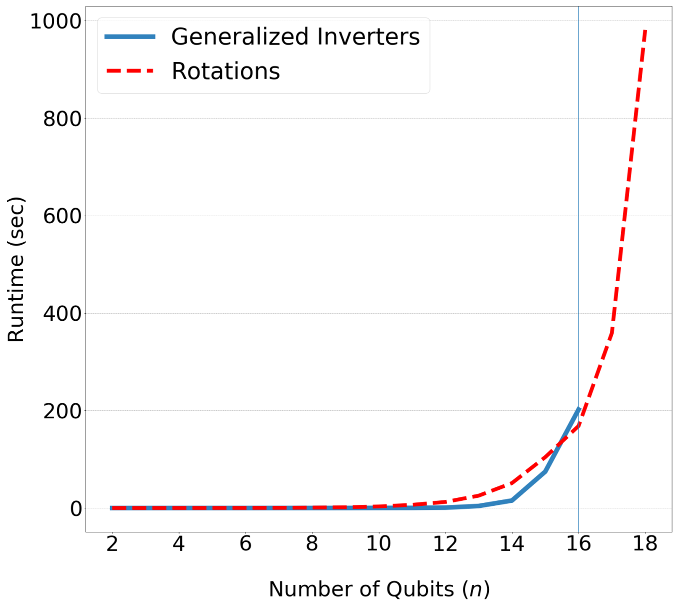


Fig. 3.9 The simulation runtimes for the generalised inverters (solid line) and rotational (dashed line) implementations. For $n > 18$ the classical machine is unable to simulate the generalised inverter implementation. Simulations are run for one step. The number of qubits n only refers to the state space of the walk, i.e., does not include ancilla qubits or the coin.

of memory. On the other hand, the rotational approach is able to simulate the quantum walk for state spaces of up to 18 qubits.

Experiments and Results on the Quantum Computer

The results obtained by executing the experiments on the quantum processor are quite different. Both the quantum walk experiments are executed on the IBMQ 15-qubit Melbourne machine. Due to limitations on the number of iterations permitted on the machine, the quantum walk is repeated 1,000 times in what constitutes a batch of trials. Each batch of trials is repeated 100 times and the probability distributions are mustered from the average results of the cumulative 100,000 repetitions of the experiment. The resulting probability distributions for the three first steps of the quantum walk on a 4- or 8-cycle can be seen in Figure 3.7 and Figure 3.8 respectively. It is noteworthy that, since for a quantum walk on a 4-cycle there are no inverters with more than two control qubits, a rotational implementation is not needed.

As evident from the results, the empirical distributions differ greatly from the simulations. None of the expected properties of a quantum walk are present. More specifically, the process no longer exhibits modular behaviour as there exist measured states that should not occur, especially on the walks with larger state space or number of steps. Unfortunately, the resulting probability distributions are completely different to the theoretical

Quantum Gate	Symbol	Execution Time (in ns)
Hadamard	H	$t_h = 53.4$
Inverter (`NOT`)	X	$t_x = 107$
Controlled-inverter (`CNOT`)	X_c	$t_{cx}^{\text{inv}} = 722.5$, $t_{cx}^{\text{rot}} = 695.7$
Toffoli (`CCNOT`)	X_{cc}	$t_{ccx} = 3{,}665.9$
Phase	R_z	$t_{rz} = 1{,}445$
Rotation	R_y	$t_{ry} = 1{,}605$

Table 3.1 Execution time of the various quantum gates used to construct the quantum walk circuits. As a controlled-inverter exhibits different execution time on each pair of qubits, the execution time presented here is computed as the average of the execution times for all the qubit pairs that participate in each circuit; t_{cx}^{inv} corresponds to the generalised inverter circuit and t_{cx}^{rot} to the rotational circuit.

ones, making it difficult to correctly calculate the variance in a comparable way to the simulations. Thus, no remarks regarding the variance can be made with certainty.

Taking into account the nominal execution time of the different gates that participate in the circuits, showcased in Table 3.1 as given by IBMQ, the overall runtime of the quantum computations can be calculated. The quantum circuit for one step of the walk on a three-qubits state space contains the following number of gates: one Hadamard, two inverter gates, eight `CNOT` gates and 10 Toffoli gates. Additionally, as the execution time of a `CNOT` depends on which qubit pair the gate is applied, one has to consider the qubits that participate in the circuit according to the architecture of the QPU. Here, the average execution time of all the relevant qubit pairs is found, as shown in Table 3.1. For example, following the architectural graph of the IBMQx15 Melbourne computer, as shown in Figure 2.2 and knowing that qubits 0 to 5 participate in the generalised inverter circuit, the qubit pairs can be found as $0-1$, $1-2$, $2-3$, $3-4$ and $4-5$. Thus, following the aforementioned table, the execution time can be calculated as

$$t_{\text{inv}} = t_h + 2 \times t_x + 8 \times t_{cx}^{\text{inv}} + 10 \times t_{ccx} \approx 42\,\mu\text{s}.$$

Similarly, the execution time of the rotational circuit can be calculated after taking into account the number of gates of each type that participate in the circuit, i.e., one Hadamard for the quantum coin, 12 of each of the phase (R_z) and rotation gates, two inverters, six controlled-inverters and 10 Toffoli gates. Additionally, since there are different pairs of qubits in the rotational circuit, the average execution time of the controlled-inverter differs from the generalised `CNOT` case. More specifically, the rotational approach utilises qubits 0 to 3, with qubit pairs $0-1$, $1-2$ and $2-3$. Thus, following again Table 3.1 the execution time is:

$$t_{\text{rot}} = t_h + 12 \times t_{rz} + 12 \times t_{ry} + 2 \times t_x + 6 \times t_{cx}^{\text{rot}} + 10 \times t_{ccx} \approx 76\,\mu\text{s}.$$

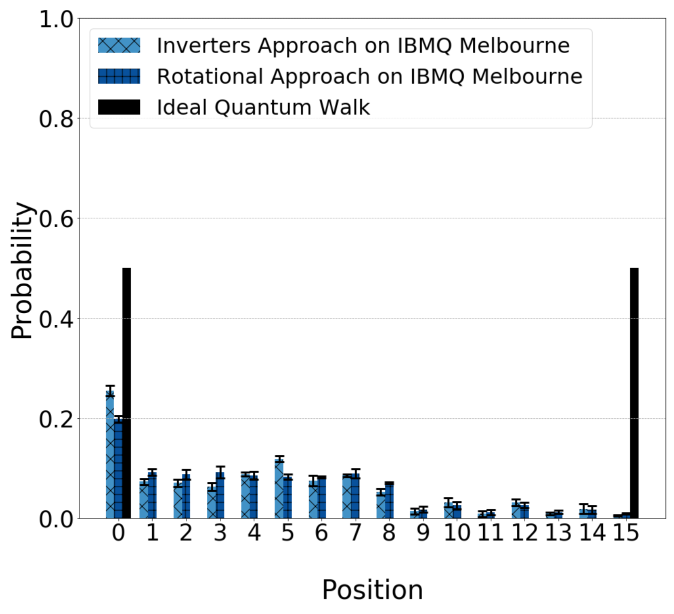


Fig. 3.10 Probability distributions of generalised inverter (crossed bar), rotational (tiled bar) and ideal (solid bar) quantum walks on four qubits for one step. Error bars are calculated with 95% confidence intervals.

For one and two steps of the walk on two qubits, the execution times are 10.5 and 21 µs respectively. It is important to mention here that these execution times are purely for the operations themselves, meaning additional time factors are not taken into account, like state preparation (if it occurs), measurement or buffer time between operations on the same qubit. Since there is no major idle session for the quantum circuit, for the purpose of this analysis these times account for a much smaller period of the execution time than the gates themselves.

In general, shorter quantum walk circuits, for example one or two steps of a two-qubit quantum walk, can generally provide results closer to the expectations, as shown in Figure 3.7(a-b). This is due to the low execution time of the quantum circuit, as calculated above, compared to the average coherence time of the three qubits participating in the circuit, calculated at 64.25 µs, and the smallest decoherence time among these qubits, i.e., 56 µs, measured through IBMQ experience [64] on the date of the experiments. For more than two steps the distribution starts to deviate from the expected (Figure 3.7(c-d)).

As shown in Figure 3.8(a-c), the effects of the noise remain intense for both approaches on an 8-cycle. The average coherence of the qubits participating in the circuit is found to be 58.6 µs for the six qubits in the inverters approach and 63.6 µs for the four qubits of the rotational approach. Those coherence times were, again, provided through IBMQ experience on the date of the experiments.

Finally, similar results are obtained on larger state spaces. One example is one step of the walk on a four qubit state space, shown in Figure 3.10. Thus, it is safe to conclude that, for one or two steps on a two qubits state space, the quantum walk behaves relatively close to the expectation, whereas for more than three steps or for state spaces larger than three qubits, where the runtime of the circuit is longer and approaches the coherence time, the effects of the noise are overwhelming.

Comparison via Quantum Volume

An elegant way to compare the two approaches also arises when using the quantum volume, as introduced in Definition 2.6. This section tries to provide an insight on such a comparison for the two approaches when executing a three-qubit quantum walk. In this context, the quantum volume is considered with opposite logic: used as a value that describes the expected resources necessary for a circuit to run on a quantum computer. In other words, it attaches an additional metric that shows the required resources of each quantum walk approach. For a meaningful comparison, it is required that both circuits are run on the same quantum architecture, a condition met here as they are both executed on the IBMQx15 Melbourne machine.

The quantum volume can be calculated via equation (2.18). It is important here that, due to automatic circuit optimisation of the connectivity applied by the backend compiler before execution on the quantum computer, additional qubits that store intermediate quantum states have to be taken into account when calculating the quantum volume. Subsequently, the size of the workspace for the generalised inverters approach is $n_{\text{inv}} = 8$ and for the rotational $n_{\text{rot}} = 6$. For the qubits that participate in the workspace of each approach, the average effective error rates can be easily computed as the average of the two-qubit error rates provided by the machine calibrations as $\epsilon_{\text{eff}}^{\text{inv}} = 2.74 \times 10^{-2}$ and $\epsilon_{\text{eff}}^{\text{rot}} = 3.10 \times 10^{-2}$ for the inverters and rotational approach respectively, as of the day of the experiment.

Substituting the above values in equation 2.18, the quantum volume for the generalised inverters approach is computed as $V_Q^{\text{inv}} = 20.812$ and for the rotational approach as $V_Q^{\text{rot}} = 28.905$. In a similar fashion, the quantum volume for a quantum walk of arbitrary size can be calculated.

Thus, the safe conclusion is that the generalised inverters approach would require the smaller quantum volume of the two implementations due to the much smaller circuit depth and subsequently reduced execution time compared to the rotational approach, as well as lower the cumulative effective error due to hardware infidelities.

3.3 Quantum Walks in Superposition

This section presents a different approach to executing discrete-time quantum walks. The novel idea examined in this thesis is to construct a quantum circuit which essentially executes two steps of the quantum walk in superposition. This construct, named *quantum-quantum walk* (QQW), uses multiple quantum coins to progress the evolution of a discrete-time quantum walk, or in other terms, a multidimensional quantum coin. In the simplest case, one can use a two-dimensional coin, whereas in the general case a *d*-dimensional coin is needed.

In order to present the logic behind this construct it is easier to start by discussing the hierarchy of a circuit using two quantum coins, i.e., a two-coin QQW, in the simplest two-dimensional case. In order to distinguish between the two coins, one is named the walk control coin (WCC), represented by the operator C_W, and the other entanglement control coin (ECC), as operator C_E. The main element of this hierarchy, and the one that differentiates the quantum-quantum walk from other algorithms, is the fact that the WCC is controlled by a superposition induced by the ECC. In other words, the two coins are entangled in a special way. Each coin is a map to a two-dimensional Hilbert space, $\mathcal{H}^2$ and both are balanced Hadamard coins, represented by the familiar Hadamard operator, H. Essentially, post-entanglement, the individual coins act as a four-dimensional coin, denoted C_{qq}, which is a map in $\mathcal{H}^2 \otimes \mathcal{H}^2$. The quantum-quantum coin in turn is used like a quantum switch to operate the increment and decrement functions of the quantum walk, as they were defined in Section 3.2.1.

The four-dimensional quantum-quantum walk coin, C_{qq}, can be expressed as

$$C_{qq} = (X \otimes I) \cdot H_c \cdot (X \otimes I) \cdot H_c \cdot (H \otimes I), \tag{3.25}$$

where H is the Hadamard operator with matrix representation given in equation (1.6), X is the Pauli-X matrix, or a simple inverter gate, and H_c is a four-dimensional controlled Hadamard operator, defined as

$$H_c = \frac{1}{\sqrt{2}} \begin{pmatrix} \sqrt{2} & 0 & 0 & 0 \\ 0 & \sqrt{2} & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & -1 \end{pmatrix}.$$

Carrying out the matrix expression, the QQW coin can be expressed as the operator

$$C_{qq} = \frac{1}{2} \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{pmatrix}.$$

This operation, essentially, takes a two-qubit state and puts it into an equal superposition of all its possible states, i.e $C_{qq} |00\rangle = \frac{1}{4} (|00\rangle + |01\rangle + |10\rangle + |11\rangle)$.

The next step is to define the increment and decrement functions for the QQW. As discussed in Section 3.2.1, the operator that describes these two functions for a simple Hadamard quantum walk can be expressed via equation (1.7). Following a similar logic, in the case of the QQW and since the coin is four-dimensional, i.e., it takes up two qubits within the quantum circuit, the new shift operator, namely S_{qq}, can be defined as

$$S_{qq} = S^- \otimes |00\rangle \langle 00| + S^+ \otimes |01\rangle \langle 01| + S^- \otimes |10\rangle \langle 10| + S^+ \otimes |11\rangle \langle 11|. \tag{3.26}$$

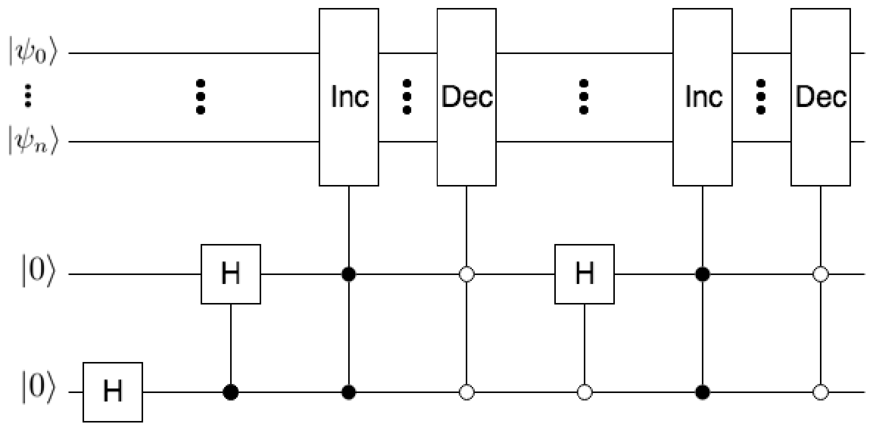


Fig. 3.11 Two-coin quantum-quantum walk circuit for 1 iteration (coin flip).

where S^+ is the Increment function which increases the quantum state by 1 and S^- is the Decrement function, which decreases the quantum state by 1. In other words, $S^+ : |x\rangle \to |x+1\rangle$ and $S^- : |x\rangle \to |x-1\rangle$, the same as in the simple QW case.

Finally, the quantum-quantum walk operator can now be defined following the same logic as the simple quantum walk operator of equation (1.5):

$$U_{qq} = S_{qq} \cdot (C_{qq} \otimes I). \tag{3.27}$$

The quantum circuit schematic that describes the one step of the quantum-quantum walk is shown in Figure 3.11.

Quantum State Vector

For more clarity, this paragraph offers an example of the evolution of an arbitrary quantum state, $|x\rangle$, undergoing one step of the quantum-quantum walk. Assume a cycle graph with eight quantum states, needing $n = 3$ qubits to be represented, or otherwise, a three-qubit quantum register. The quantum coin C_{qq} is four-dimensional, i.e., it needs a two-qubit register and is initialised in state $|00\rangle$. The evolution of a QQW, defined by U_{qq}, applied on a quantum system initialised on an arbitrary quantum state $|x\rangle|00\rangle$ will have the following effect

$$\begin{aligned}
U_{qq}|x\rangle|00\rangle &= S_{qq} \cdot (\mathbb{I} \otimes C_{qq})|x\rangle|00\rangle \\
&= S_{qq} \cdot \left[|x\rangle \otimes \frac{1}{4}(|00\rangle + |01\rangle + |10\rangle + |11\rangle)\right] \\
&= \frac{1}{4}S_{qq}|x\rangle|00\rangle + \frac{1}{4}S_{qq}|x\rangle|01\rangle + \frac{1}{4}S_{qq}|x\rangle|10\rangle + \frac{1}{4}S_{qq}|x\rangle|11\rangle \\
&= \frac{1}{4}|x-1\rangle|00\rangle + \frac{1}{4}|x+1\rangle|01\rangle + \frac{1}{4}|x+1\rangle|10\rangle + \frac{1}{4}|x+1\rangle|11\rangle
\end{aligned}$$

It is visible that after one flip of the four-dimensional coin, the particle initialised in a quantum state $|x\rangle$ will have undergone a transformation where it gets increased and decreased twice, seemingly at the same time. Intuitively, one can treat the quantum-

quantum walk as two normal Hadamard quantum walks running in a superposition. For each coin flip the walker will propagate twice, once for each of the increment/decrement circuit pairs. The effects of this can be better shown on the variance of the quantum-quantum walk, as discussed later.

Quantum-Quantum Walk Algorithm

Algorithm 3 better showcases the philosophy behind the QQW.

Algorithm 3: Quantum-quantum walk algorithm

```
1 Initialisation. Initialise the quantum registers. The state register will be
    initialised in the preferred quantum state, i.e $|x\rangle$. The coin register will be
    initialised in the usual basis state $|00\rangle$.
2 Coin flip. Flip the coin, or otherwise, apply the transformation $C_{qq}$ as defined in
    equation (3.25).
3 Evolution. Evolve the particle's position according to the operator $S_{qq}$, as defined
    in equation (3.26). This completes the current step of the quantum-quantum
    walk, expressed by the operator $U_{qq}|x\rangle|00\rangle = S_{qq} \cdot (C_{qq} \otimes \mathbb{I})|x\rangle|00\rangle$.
4 Iteration. If $t$ coin flips have occurred, move to step 5. This concludes the
    quantum-quantum walk. Otherwise, go to step 2.
5 Measurement. Measure the quantum states register.
```

Experimental Variance and Circuit Runtime

Figure 3.12 shows the variance of a quantum-quantum walk compared with the simple quantum walk. It is visible from the showcased figure that the QQW shows an increase in the variance, i.e., the walker propagates further than in the simple quantum walk. This is an expected result as, for each coin-flip, the walker will evolve twice (i.e., go on double the distance).

Even though the above figure shows increase in the speed of a quantum walk via the QQW construct, the major downside is the execution time. In order to better realise the execution runtime of the constructs presented in this section, three main times of interest are defined: (i) the coin-flip time, t_c, i.e., the time it takes for the coin-flip (Hadamard gate) to be executed; (ii) the increment time, t_inc, i.e., the time it takes for the increment subcircuit to be implemented; and (iii) the decrement time, t_dcr, with obvious definition.

Thus, it is easy to realise that the execution time of one step of the simple quantum walk will be:

$$t_\mathrm{qw} = t_\mathrm{c} + t_\mathrm{inc} + t_\mathrm{dcr}.$$

On the other hand, the execution time for the quantum-quantum walk can be calculated as

$$t_\mathrm{qqw} = 2 \times (t_c + t_\mathrm{inc} + t_\mathrm{dcr}) = 2 \times t_\mathrm{qw}, \tag{3.28}$$

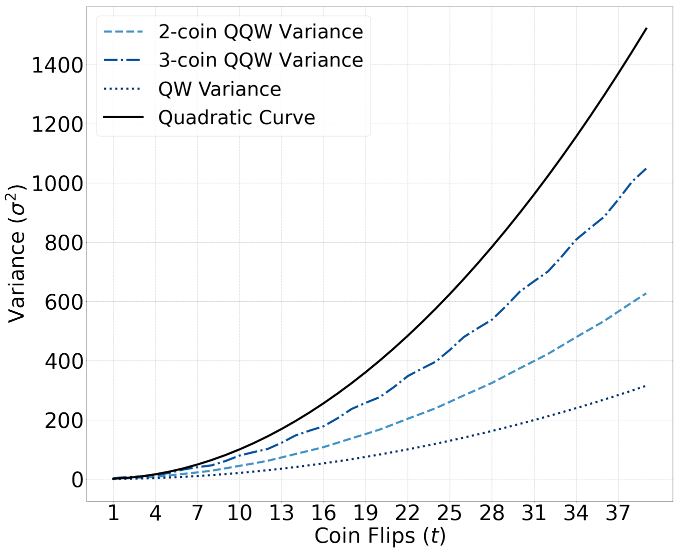


Fig. 3.12 Variance of the two- and three-coin quantum-quantum walk, as compared to the variance of the simple Hadamard walk and the theoretical variance for 39 coin flips.

as there are two coin executions, two increment and two decrement functions in the same step. This relationship between QW and QQW intuitively makes sense as two quantum walk circuits are running in sequential superposition.

In conclusion, even though the quantum-quantum walk shows an increase in the variance of the simple quantum walk, it also takes twice as long to run. In other words, the advantage given by the variance when considering a four-dimensional coin disappears when considering the execution time. This collapses the quantum-quantum walk down to the same complexity as running two steps of a single Hadamard quantum walk, as shown in equation (3.28). Thus, it is safe to conclude that, via the quantum-quantum walk construct presented in this section, there can be no major advantage gained over the simple quantum walk.

Finally, this implementation can be extended to accommodate larger number of quantum walks running in superposition by adding more quantum coins that control each one. The implementation follows a natural extension of the schematic of Figure 3.11. The advantages and disadvantages are again the ones described above, i.e., an increase in the variance negated by the increase in the size of the circuit (or runtime). Figure 3.12 also shows the variance for a quantum-quantum walk controlled by three quantum coins.

3.4 Discussion

As mentioned in the introduction, quantum walks can form the base for many algorithms providing clear advantage, as comprehensively shown in Section 3.1. With the increasing attention that areas like quantum machine learning receive from within the field of quantum computing, it is easy to spot the many advantages offered by quantum walks and the reasons why this process leads to alluring real-world applications. Thus, the value of research on quantum walks is evident.

Following the theoretical presentation of a discrete-time quantum walk in Section 1.2.1, two approaches are used to implement it: (i) the generalised inverters [19], and (ii) the rotational approach. The experiments show that using generalised inverters keeps the implementation simple and the circuit shallow but requires an ancilla register. The number of ancilla qubits increases linearly with the number of control qubits quickly leading to a large workspace, limiting the capabilities of the experiments. The rotational approach deals with this limitation by rendering the ancilla register obsolete, allowing a much larger state space for the quantum walks. The disadvantage of the rotational approach is the larger complexity of the resulting circuit.

It is evident that the two implementations of quantum walks offer opposite advantages and disadvantages. Implementing a quantum walk with generalised inverters shows smaller execution time and requires exponentially less gates as a function of the size of the state space, or in other words, smaller circuit depth. On the other hand, the rotational circuit requires less qubits to be used in the workspace, as there is no need for an ancilla register, but the circuit is much deeper than the former approach.

The dependency of the operations between the qubits within the architectures does not allow for any gates to be run in parallel, restricting the width of the circuit at each time step. Thus, the runtime of the quantum circuit quickly surpasses the coherence time of the qubits, leading to immensely noisy distributions. For very small state spaces (i.e., two qubits) the distributions form closer to the theory with lower level of noise. This does not hold for walks with a three-qubit state space or larger. The execution time of the circuit with relation to the coherence time of the qubits greatly affects the resulting distribution of the quantum walk.

Due to the stochastic nature of the noise, it is very difficult to draw safe conclusions from the comparison of the two approaches on the quantum computer. However, one can point out that since the rotational circuit is deeper, the cumulative error due to hardware infidelities will be more extensive. Additionally, since the computation also takes longer in the case of the rotational circuit, the active qubits have a higher chance to decohere. Thus, it is safe to claim that the rotational circuit will be the noisier of the two.

It is noteworthy that, considering the state of the art, no optimal implementation of quantum walks currently exists. The generalised inverters approach proposed by [19] constitutes an efficient circuit. Other implementations might exist that could perhaps prove to be more efficient and could be the subject of further comparison. Moreover, the analysis and results presented in this chapter involve quantum walks on a discrete space. Similar experiments could be undertaken for continuous-time or space quantum walks as, in the end, any algorithm and the corresponding quantum circuit will be decomposed in a sequence of universal quantum gates, which can then be examined with the methodology and analysis presented in this chapter. Nevertheless, some interesting analysis could arise from examining the behaviour of continuous-space and/or time quantum walks and their characteristics when implemented on NISQ machines.

Due to large presence of noise in the quantum machine, no firm conclusions can be drawn regarding today's quantum hardware. This chapter presents a resource theory and compares two different implementations of quantum walks. Therefore, considering the limitations and hardware constraints of NISQ machines, this analysis can assist in the realisation of quantum walks on near-term quantum computers in an efficient way. For example, knowing the levels of noise within a quantum computer can help in choosing a more suitable implementation for executing quantum walks. The research finds that the generalised inverters approach exhibits less error if the decomposition of the generalised gates is not excessively long, making it more suitable for noisier machines. On the other hand, the rotational approach appears to be the only option when quantum walks, or their simulations, are executed on a state space that is too large to be executed on a quantum computer or be simulated on a classical machine.

Furthermore, the methodology followed in this chapter can be extended to compare any number of implementations of arbitrary quantum algorithms, including circuits implementing a quantum Hamiltonian. To understand this, it is important to remember that any algorithm or Hamiltonian, in order to be implemented on a *digital* (discrete) quantum computer, will be decomposed in a sequence of quantum gates that can be carried out on said quantum computer. The above methodology can then be used to compare the end circuits.

Finally, an alternative construction that evolves quantum walks in superposition is presented. This implementation shows an increase in the variance for each step of the superposed quantum walk with the disadvantage of a larger execution time. More specifically, one step of the quantum-quantum walk takes twice as long as one step of a simple quantum walk, an expected result as the circuit is twice the size. Thus, it is evident that no advantages in the performance of a quantum walk can be gained by superposing multiple steps with a higher dimensional quantum coin, at least not with the configuration presented within this chapter.

3.5 Future Work

The work done in this chapter outlines a coherent way to compare circuit approaches to quantum walks using an in-depth theoretical and experimental analysis, as well as the quantum volume. This can constitute a framework for further analysis, either between other approaches to executing quantum walks or for finding suitable quantum circuits, considering available quantum computers. This research can be used or further developed in the future in order to obtain valuable information about the efficiency of circuit implementations when designing more complex algorithms that are based on quantum walk circuits (like, for example, in quantum machine learning as discussed above).

Furthermore, alternative configurations of quantum walk circuits or clever designs that could lead to additional speed-ups are also some open areas of research. The example work done here could lead to further research on superposed quantum walks, quantum-quantum walks or constructs that achieve quantum circuits running in parallel. Alas, such research falls outside of the scope of this thesis, and thus no additional research is pursued here.

Chapter 4

Quantum Noise Modelling and Simulations

It has been proven that arbitrarily long quantum computation is possible given constraints on the error rates and the error locality [69, 3, 70]. Quantum noise is one of the main characteristics of near-term (i.e., NISQ) devices and, so far, there are no existing quantum computers that can operate noise-free, regardless of technology or architecture. Thus, considering that limiting the noise within a QPU is a necessary step towards the goal of commercial and universal quantum computation, detailed studies and characterisation of said noise in quantum systems is valuable.

One method for mitigating quantum noise is through quantum error correction [3, 70–79], which relies on knowing what the most likely error sources are. It has been found that error correcting methods optimised for specific noise in a system can dramatically outperform generic ones [80, 81]. Thus, identifying, characterising and simulating the noise in quantum computers is important and can lead to much more efficient calibration and error correction, which are necessary for large-scale quantum computing [82].

The most common error model is a depolarising Pauli channel. Effectively, a Pauli operator is chosen to be applied to operations that have a probability to produce an erroneous result [73, 76, 83]. This unital channel (see Definition 2.4) is generally a good approximation of most error processes that lead to a maximally mixed noisy state. Alternative depolarising channels can use Clifford operations to effectively approximate quantum errors [84].

A large portion of error also derives from non-unital interactions between the quantum system and the environment. Noise of this type causes decoherence during the computation in various forms. The most common one is thermal exchange, i.e., the exchange of thermal energy between the quantum system (for example, the QPU) and its environment, which plays a central role in the noise modelling within this thesis. The non-unital nature of such quantum noise makes it difficult to simulate it with Pauli or Clifford operations, leading to a more complicated approach. Other forms of decoherence, like electromagnetic or gravitational, are not considered within this work. It is nevertheless believed that their study can also prove valuable to understanding the effects of noise within a quantum machine.

This chapter provides the theoretical foundation and a detailed analysis of a new model that can be used to approximate the noise during the evolution in a quantum computer. First of all, a comprehensive presentation of the relevant sources of error in a quantum computer is provided. Next, the thesis introduces a set of parameters that express the levels of noise within the quantum computer and are used by the models during simulations. Section 4.3 then offers an introduction, analysis and coherent review of the new model, before showcasing the experiments with a real quantum computer in Section 4.4. Finally, a discussion of the results and possible future extensions to the work is offered.

The work outlined in this chapter has been published in Physical Review A in December 2021 [85].

4.1 Sources of Noise within NISQ Machines

As described in Chapter 2, within the NISQ era, quantum noise is a central obstacle in the progress of quantum computation, currently preventing the construction of large-scale quantum computers and the execution of long quantum computations. Noise mainly occurs either due to infidelities of the quantum hardware, or otherwise, faults of the operations that take place during the computation (i.e., quantum gates, measurement or state preparation devices), or due to unwanted interactions of the system with its environment, a process that is commonly called decoherence (for more details on the mechanism of decoherence see Section 2.1.4). There exist various forms of decoherence, for example thermal which occurs, as mentioned above, due to the exchange of thermal energy between the system and the environment, electromagnetic which is caused by interactions of the qubits with an electromagnetic field, or gravitational decoherence [86–89]. The latter form of decoherence is mostly theoretical and can occur due to time dilation caused by quantum particles being in different locations of spacetime [90].

This thesis is mainly concerned with three sources of error: (i) hardware and control infidelities, (ii) decoherence in the form of thermal exchange and (iii) dephasing of the qubits within the QPU. The first source of error refers to noise that arises in the form of hardware errors during the execution of a quantum circuit. They are often called control infidelities as they appear during operations that attempt to control the quantum computation. This type of noise affects operations that target a qubit (or pair of qubits) aiming to control and/or perform various actions on the qubit(s). The thesis follows a simple categorisation of the type of hardware infidelities according to which part of the hardware they originiate from or which operation they affect:

- **Initialisation errors.** These are also known as state preparation errors and refer to the failure of the control hardware to initialise a qubit (or a qubit register) on a desired state.

- **Measurement errors.** Also known as readout errors, they refer to the erroneous measurement of the state of a qubit by the readout apparatus at the end of the computation.

- **Qubit operation (gate) errors.** These are errors induced by infidelities of the control operations that constitute a quantum gate. They can be further categorised according to the type of gate to (i) single-qubit errors that correspond to gates applied on a single qubit (for example Hadamard or Pauli-X), and (ii) two-qubit errors, or errors that occur during the execution of two-qubit operations (like a controlled-inverter, or `CNOT`).

The hardware infidelities are a form of *coherent error*, i.e., in theory, they do not induce decoherence during the execution of the quantum circuit, but they simply make undesired changes to the state of the qubit. They are purely systematic and represent the most simple type of noise from a modelling perspective. Nevertheless, one might connect hardware infidelities with the other noise groups. For example, one might argue that decoherence also occurs due to hardware infidelities, as current technology fails to isolate the quantum processor from the environment, or that electromagnetic noise also identifies a part of its source to the electrical components of the machine (including gates). That said, this thesis considers the infidelities of quantum hardware as a distinct source of coherent noise due to simplicity and the advantages such a view offers on the efficiency of modelling and simulating the noise in quantum computers.

The second form of error originates from interactions between the quantum system (i.e., the QPU in this case) and its environment. Such interactions induce decoherence during the evolution of the quantum circuit (a more comprehensive review of the idea of decoherence is provided in Section 2.1.4). As aforementioned, this thesis focuses on the exchange of thermal energy between QPU and environment as spontaneous emission or absorption of photons. This exchange occurs with varying intensity as time progresses during the quantum computation (more information on the thermal decoherence is given in Section 2.1.6).

The last type of noise also represents a different form of decoherence, one that, over time, drives the quantum system towards classical behaviour. This effect takes place as a loss of phase of the qubit as time progresses during the evolution of the quantum circuit in a similar manner to thermal decoherence. Dephasing is a more complex concept than the thermal exchange induced decoherence from a theoretical perspective, but both share a similar modelling approach, as will be shown in the following sections. Further information on the dephasing of the qubits can be found in Section 2.1.7.

A more detailed analysis of the three sources of noise this thesis works with, as well as their mathematical description, is provided in the following sections.

Noise Source	Type of Parameter	Number of Parameters
Hardware Infidelities	Error Rates	$r + m + s$
Thermal Decoherence	Time T_1	n
Dephasing	Time T_2	n

Table 4.1 Type and number of parameters for each of the three error groups; n is the number of qubits in the system, m is the number of qubits that are measured, s is the number of qubits that undergo state preparation and r is the number of distinct types of gates implemented in the architecture, each considered once per qubit or pair of qubits.

4.2 Quantum Noise Parameters

In order to better present the main ideas of the quantum noise models engineered within this chapter, it is important to first discuss an important functionality of the noise models. For the simulations to work and provide an accurate approximation of the quantum evolution, it is necessary to obtain an indication of the levels of noise within the quantum computer during the execution of the circuit. To that end, the engineered models make use of a set of *quantum noise parameters*. These parameters are usually given by experimental calibration of the quantum computer. There are a few techniques used to calibrate the error rates and decoherence times of quantum computers, like cross-entropy benchmarking [91, 92], process tomography [93–95] or randomised benchmarking [96–103]. Randomised benchmarking specifically is the most used and most prominent technique, with a few recent alterations like cycle benchmarking [104] or dihedral benchmarking [105].

For the hardware and control infidelities, the noise parameters come in the form of operation error rates: they represent the probability that an operation, be it a gate, measurement or state preparation, when applied in the quantum circuit, produces an erroneous outcome. Each individual type of gates implemented within the architecture (Pauli gates, Clifford gates, `CNOT`, etc.) is associated with a specific error rate. Additionally, each type of gate has different error rates for each qubit(s) that they are applied on. Similarly, the state preparation and measurement noise parameters are a selection of error rates that represent the probability that the preparation of the initial quantum state or the outcome of a measurement will be erroneous. Each qubit in the system yields different error rates when prepared or measured. Finally, for the thermal decoherence, the parameters are represented by the thermal decoherence times per qubit q, i.e., $T_1(q)$ and, similarly, the dephasing noise parameters are the times $T_2(q)$. As explained in the previous section, decoherence and dephasing occur with varying intensity over time as the computation progress. The T_1 and T_2 times essentially represent the time it takes for each qubit within the QPU to decohere or dephase, respectively.

Table 4.1 shows the type of parameters with respect to each quantum noise channel, as well as how many parameters are associated with each error source. Finally, it is important to mention here that for the remainder of the chapter, it is assumed that the calibrated quantum noise parameters remain invariable during the execution of the quantum circuit.

4.3 The Unified Noise Model

This section aims to introduce and analyse a comprehensive model for approximating the noisy evolution of a circuit within NISQ computers. The experiments are focused on four of the IBMQ computers, but the engineered models can be applied on any quantum computer. The framework for modelling the quantum noise is guided by the sources of error identified in Section 4.1, which are subsequently categorised in three error groups: (i) hardware infidelities in the form of depolarising Pauli noise, (ii) state preparation and measurement (SPAM) errors and (iii) decoherence in the form of thermal relaxation and dephasing. To avoid confusion, it is important to understand that there is no one-to-one correspondence between these three error groups and the sources of error discussed in Section 4.1. The two sets represent different categorisation of the noise: the sources of noise show where the quantum noise comes from whereas the error groups arrange the noise in a sensible way for modelling, as will be made clear later on.

Each error group is modelled independently from the other two. In order to form a mathematical and mechanical foundation for the effect of each error group within its respective model, this work uses quantum channels (more on quantum channels can be found in Section 2.1.8). The quantum channels themselves do not represent a novel idea but are well-known from the bibliography. The main contribution of this section is the specific combination, or unification of the channels in one single model, as well as the implementation, experiments and results. The three error groups, along with the respective channels used for their modelling, are discussed in the following subsections.

4.3.1 Error Group 1: Modelling Hardware and Control Infidelities

The first error group contains the errors that stem from hardware infidelities during quantum operations (i.e., quantum gates). The channel used to model this error group is known as a *symmetric depolarising channel*, a term which within this thesis will be used interchangeably with the terms *gate infidelities*, or simply, *depolarising channel.* These errors take the form of either a phase-flip (i.e., an inversion of the phase of the qubit during the computation) or a bit-flip (i.e., an inversion of the state of the qubit), or both. The corresponding quantum channel essentially simulates the bit-flip and phase-flip errors due to gate infidelities within the circuit as a depolarising channel [30, 106–108]. It is assumed that an error of this group occurs with probability p_1, and the bit-flip and phase-flip errors are defined through the Pauli-X and Z operations. In the case that both a bit- and phase-flip happen, the operation is defined through Pauli-Y. All three types of Pauli errors have the same probability of occurring. The depolarising channel can be represented by the following operators (an introduction to Kraus operators and the operator-sum representation is provided in Section 2.1.3):

$$\begin{aligned} K_{D_0} &= \sqrt{1-p_1}I, \\ K_{D_1} &= \sqrt{\frac{p_1}{3}}X, \\ K_{D_2} &= \sqrt{\frac{p_1}{3}}Z, \\ K_{D_3} &= \sqrt{\frac{p_1}{3}}Y. \end{aligned} \tag{4.1}$$

The effect of the depolarising channel on a quantum system can be expressed via the operator-sum representation, as

$$\rho \mapsto \mathcal{D}(\rho) = \sum_{i=0}^{3} K_{D_i} \rho K_{D_i}^{\dagger} = \begin{pmatrix} \left(1-\frac{4p_1}{3}\right)\rho_{00} - \frac{2p_1}{3}\rho_{11} & \left(1-\frac{2p_1}{3}\right)\rho_{01} \\ \left(1-\frac{2p_1}{3}\right)\rho_{10} & \left(1-\frac{4p_1}{3}\right)\rho_{11} - \frac{2p_1}{3}\rho_{00} \end{pmatrix}$$

where $\rho = \begin{pmatrix} \rho_{00} & \rho_{01} \\ \rho_{10} & \rho_{11} \end{pmatrix}$ is the density matrix for a qubit. It is noteworthy that, as $K_{D_i} = K_{D_i}^{\dagger}$, one can do the relative replacement in the above representation.

4.3.2 Error Group 2: Modelling State Preparation and Measurement Errors

The second error group contains the errors that occur during the measurement (otherwise readout errors) at the end of the computation and the state preparation wherever it takes place (at the start or any point during the computation). These two sources of noise have been grouped together and are separate from the other hardware infidelities as their effects are quite similar and simpler than the effects of gate infidelities on the quantum computation.

The channel that models this error group is essentially a simple Pauli-X error. Thus, the state preparation and measurement (SPAM) quantum channel for the measurement errors can be defined via the following operators

$$\begin{aligned} K_{M_0} &= \sqrt{1-p_2}I, \\ K_{M_1} &= \sqrt{p_2}X \end{aligned} \tag{4.2}$$

where p_2 is the probability that the measurement is incorrect.

The effect of the SPAM channel for measurement errors can be expressed through the density matrix, ρ, as

$$\rho \mapsto \mathcal{S}(\rho) = K_{M_0}\rho K_{M_0} + K_{M_1}\rho K_{M_1} = \begin{pmatrix} (1-p_2)\rho_{00} - p_2\rho_{11} & (1-p_2)\rho_{01} - p_2\rho_{10} \\ (1-p_2)\rho_{10} - p_2\rho_{01} & (1-p_2)\rho_{11} - p_2\rho_{00} \end{pmatrix}.$$

In the case that state preparation takes place in the computation, the error channel (i.e., $\rho \mapsto \mathcal{S}'(\rho)$) is of similar form to the measurement case, with the qubit failing to be prepared at the desired state, resulting in the inverted state by X with probability p_2'.

4.3.3 Error Group 3: Modelling Thermal Decoherence and Dephasing

The third error group models the interaction between the physical qubits and the environment. There are two types of noise within this error group: (i) the thermal decoherence that occurs over time in the form of excitation/relaxation, or otherwise, the emission or absorption of a photon by the quantum system (here the QPU), and (ii) the dephasing of the qubits over time.

Thermal decoherence, as examined in Section 2.1.6 is a non-unital (i.e., irreversible) process that describes the thermalisation of the qubit spins towards an equilibrium state at the temperature of their environment. This process involves the exchange of energy between the QPU and the environment, which drives the qubits either towards the ground state, $|0\rangle$ (relaxation or de-excitation or reset to $|0\rangle$, which corresponds to photon emission) or the excited state, $|1\rangle$ (excitation or reset to $|1\rangle$, which corresponds to photon absorption).

On the other hand, dephasing, as introduced in Section 2.1.7, refers to the ways in which the coherence of the qubits within the QPU decays over time during the execution of the quantum circuit. This mechanism, which can also be characterised as the absolute form of decoherence, describes the transition of the quantum system towards classical behaviour.

In order to engineer a quantum channel that models these two processes, one needs to take into account:

- the average execution time of each type of quantum gates g implemented, denoted T_g;
- the time it takes for each qubit q to relax and dephase, commonly denoted $T_1(q)$ and $T_2(q)$ respectively, with $q \in [0, n-1]$, where n represents the number of qubits in the quantum computer.

In other words, $T_1(q)$ describes an evolution towards equilibrium as a perturbation orthogonal to the quantisation axis (x, y-component of the Bloch vector) and $T_2(q)$ describes a slow perturbation along the quantisation axis (z-component of the Bloch vector), or otherwise, the behaviour of the off diagonal elements over time for each qubit. These two times follow a well-known relation expressed as $T_2(q) \leq 2T_1(q)$. For the purposes of creating a program that replicates the above behaviour, there already exists a function implementing this error group as a quantum channel within Qiskit[1] and further details of said implementation can also be found in [109].

Considering the thermal decoherence and dephasing times $T_1(q)$ and $T_2(q)$, as well as the (known) gate execution times T_g, the probability for each qubit q to decohere and dephase after a gate of type g is applied to it can be defined as $p_{T_1}(q) = e^{-T_g/T_1(q)}$ and $p_{T_2}(q) = e^{-T_g/T_2(q)}$ respectively. Then, the probability for a qubit to reset to an equilibrium state can be defined as

$$p_{\text{reset}}(q) = 1 - p_{T_1}(q). \tag{4.3}$$

Taking into account the thermal decoherence transition picture as described earlier (excitation and de-excitation), there is a specific way to express the tendency of the quantum system to reach each of the two equilibrium states (i.e., the ground or the excited state). This is defined in the form of a weight that dictates towards which of the two equilibrium states ($|0\rangle$ or $|1\rangle$) this noise (or reset error) drives each qubit, q, and can be calculated as [109, 110]

$$w_e(q) = \frac{1}{1 + e^{2hf_q/k_B\Theta}}, \tag{4.4}$$

where Θ is the QPU temperature, h is Planck's constant, k_{B} is Boltzmann's constant and f_q is the frequency of the qubit. In essence, the qubit frequency is the difference in energy between the ground and excited states. This frequency is crucial from a qubit engineering perspective, for creating pulses which enact particular quantum operations on the qubit – a topic that falls outside the context of this thesis.

Important here is that the model assumes that the decoherence and dephasing noise occurs for each qubit in the system independently. Thus, for better presentation of the equations hence forth, it is preferred to selectively omit the presence of the qubit identifier, q (i.e., p_{reset} instead of $p_{\text{reset}}(q)$, etc.). Thus, knowing the time parameters T_1, T_2 and T_g, the weight w_e as in equation (4.4) and the probability of a reset to an equilibrium state p_{reset} as in equation (4.3), the following forms of noise can be identified for the case $T_2 \leq T_1$:

- **Dephasing:** a phase-flip which occurs with probability $p_Z = (1-p_{\text{reset}})(1-p_{T_2}p_{T_1}^{-1})/2$.
- **Reset to $|0\rangle$:** this represents thermal relaxation (photon emission), or a jump to the ground state $|0\rangle$, and occurs with probability $p_{\text{reset}_0} = (1 - w_e)p_{\text{reset}}$.
- **Reset to $|1\rangle$:** this represents a thermal excitation (photon absorption), or a jump to the excited state $|1\rangle$, and occurs with probability $p_{\text{reset}_1} = w_e p_{\text{reset}}$.
- **Identity:** this signifies that no noise occurs and nothing happens to the state, or in quantum mechanical terms, the identity, I, is applied with probability $p_I = 1 - p_Z - p_{\text{reset}_0} - p_{\text{reset}_1}$.

[1]Thermal relaxation and dephasing channel in Qiskit: https://qiskit.org/documentation/stubs/qiskit.providers.aer.noise.thermal_relaxation_error.html

The operators for this case follow simply from the forms of noise described above as

$$\begin{aligned} K_I &= \sqrt{p_I} I, \\ K_Z &= \sqrt{p_Z} Z, \\ K_{\text{reset}_0} &= \sqrt{p_{\text{reset}_0}} \left|0\right\rangle \left\langle 0\right|, \\ K_{\text{reset}_1} &= \sqrt{p_{\text{reset}_1}} \left|1\right\rangle \left\langle 1\right|. \end{aligned} \tag{4.5}$$

and the operator-sum representation describing the quantum channel can be expressed as

$$\rho \mapsto \mathcal{N}(\rho) = \sum_{k \in \{I, Z, \text{reset}_0, \text{reset}_1\}} K_k \rho K_k^\dagger.$$

This far, the temperature, Θ, of the QPU has been taken into consideration as in equation (4.4). In general, according to IBMQ, the mixing chamber within which the QPU resides at the lowest part of the dilution refrigerator brings the quantum processor and associated components down to a temperature of $\Theta \approx 15\,\text{mK}$. Thus, knowing the temperature one can calculate the weight w_e from the aforementioned equation. As an example, considering the average frequency of the qubits within the IBMQx15 Melbourne machine calculated at $\overline{f}_q \approx 4.9801 \times 10^9$Hz as of the time of the experiments, the average weight can be calculated from equation (4.4) as $\overline{w}_e \approx 1.44532 \times 10^{-14}$. Thus, an excitation occurring with probability $p_{\text{reset}_1} = \overline{w}_e(1 - p_{T_1})$ can be easily considered a rare event and, subsequently, can be omitted from the model, a proof that reiterates the theoretical point in Section 2.1.6. Therefore, it can be effectively assumed that the reset error takes the form of only reset to the ground state, $|0\rangle$, or in other words, that the device temperature is $\Theta = 0$. For simplicity in the following analysis, thermal decoherence will be referred to simply as (thermal) relaxation.

Considering the above, the quantum channel can now be simplified by disregarding spontaneous excitation. If $T_2 \leq T_1$ for every qubit, then the relaxation and dephasing noise can be expressed as a mixed reset and unital quantum channel [109]. Assuming a device temperature $\Theta = 0$, the following forms of noise can be identified:

- **Dephasing:** a phase-flip which occurs with probability $p_Z = (1 - p_{\text{reset}})(1 - p_{T_2} p_{T_1}^{-1})/2$.
- **Identity:** the identity, I, occurs with probability $p_I = 1 - p_Z - p_{\text{reset}}$.
- **Reset to $|0\rangle$:** a jump to the ground state occurring with probability $p_{\text{reset}} = 1 - p_{T_1}$.

Having omitted the thermal excitation, the remaining relaxation and dephasing channel can be represented with the following operators:

$$\begin{aligned} K_I &= \sqrt{p_I} I, \\ K_Z &= \sqrt{p_Z} Z, \\ K_{\text{reset}} &= \sqrt{p_{\text{reset}}} \left|0\right\rangle \left\langle 0\right|. \end{aligned} \tag{4.6}$$

Thus the effect of the relaxation channel when $T_2(q) \leq T_1(q)$ can be expressed via its operator-sum representation as

$$\rho \mapsto \mathcal{N}(\rho) = \sum_{k \in \{I, Z, \text{reset}\}} K_k \rho K_k^\dagger =$$
$$\begin{pmatrix} \frac{2p_I\rho_{00} - 2p_Z\rho_{00} - ip_{\text{reset}}\rho_{01}}{2} & \frac{2p_I\rho_{01} + 2p_Z\rho_{01} + p_{\text{reset}}\rho_{01}}{2} \\ \frac{2p_I\rho_{10} + 2p_Z\rho_{10} + p_{\text{reset}}\rho_{01}}{2} & \frac{2p_I\rho_{11} - 2p_Z\rho_{11} + ip_{\text{reset}}\rho_{01}}{2} \end{pmatrix}.$$

If $2T_1 \geq T_2 > T_1$ the Qiskit model implementation uses a Choi-matrix representation [111, 112]. In general, a Choi matrix is defined as

$$C = \sum_{i,j} |i\rangle \langle j| \otimes \mathcal{E}\left(|i\rangle \langle j|\right),$$

with $\mathcal{E}(\cdot)$ an arbitrary quantum channel. For a single-qubit case it stands that $i, j \in \{0, 1\}$. The Choi matrix for the thermal relaxation and dephasing model can be written as [109]

$$C = \begin{pmatrix} 1 & 0 & 0 & p_{T_2} \\ 0 & 0 & 0 & 0 \\ 0 & 0 & p_{\text{reset}} & 0 \\ p_{T_2} & 0 & 0 & 1 - p_{\text{reset}} \end{pmatrix} \tag{4.7}$$

with the various probabilities as defined above. The evolution of the density matrix ρ with respect to the Choi matrix C can be described as

$$\rho \mapsto \mathcal{N}(\rho) = \text{tr}_1 \left[C(\rho^T \otimes I) \right],$$

where tr_1 is the trace over the main system in which the density matrix ρ resides.

The transition from Choi-matrix to operator-sum representation can be realised via the spectral theorem as

$$C = \sum_{j=1}^{r} |v_j\rangle \langle v_j|$$

for vectors $v_1, \ldots, v_r$ and $r = \text{rank}(C)$. It is then a matter of deducing the Kraus operators to be $K_1, \ldots, K_r$ such that $\text{vec}(K_j) = v_j$, for $j \in \{1, \ldots, r\}$, where $\text{vec}(\cdot)$ represents the vectorisation function for a matrix, i.e., a linear transformation which converts the matrix into a column vector. If the Choi matrix is Hermitian, then, given an isomorphism Φ from $\mathbb{C}^{n^2}$ to $\mathbb{C}^{n \times n}$ with column-major order mapping, i.e., $\Phi(x)_{i,j} = (x_{i+n(j-1)})$, $i, j = 1 \ldots n$ and $x \in \mathbb{C}^{n^2}$, the Kraus operators can be expressed as

$$K_\lambda = \sqrt{\lambda}\Phi(v_\lambda),$$

where λ are the eigenvalues and v_λ the eigenvectors of C.

If the Choi matrix is not Hermitian, or if its eigenvalues are negative, then singular value decomposition (SVD) is applied. Let the SVD of the Choi matrix be

$$C = U\Sigma V^{\dagger},$$

where $\Sigma = \operatorname{diag}(\sigma_1, \ldots, \sigma_n)$, $\sigma_i \geq 0$, $U = (u_1|\ldots|u_n)$ the left singular vectors and $V = (v_1|\ldots|v_n)$ the right singular vectors. This leads to two sets of Kraus operators, one for the left and one for the right map, which can be expressed as

$$\begin{aligned} K_i^l &= \sqrt{\sigma_i}\Phi(u_i) \\ K_i^r &= \sqrt{\sigma_i}\Phi(v_i). \end{aligned}$$

If the left and right Kraus operators are not equal, i.e. $u_i \neq v_i$ for some $i \in \{1, 2, \ldots, n\}$, then they do not represent a completely positive trace preserving map, triggering an error in the thermal relaxation model.

As a final remark, it is noteworthy that, in general, the thermal decoherence (or relaxation, if simplified as presented above) and dephasing model does not account for the overall effects of decoherence and dephasing during idle times of the qubits in the execution. The main reason is that these effects are attributed mainly to electromagnetic interference and cross-talk between the qubits. What the present model accounts for is only thermal decoherence and dephasing on idle qubits over the duration of the execution on the quantum computer. More specifically, as the execution time of every quantum gate is known (i.e., T_g) and used as a parameter within the model itself, it is relatively easy to compute the total execution time of the quantum circuit after every operation takes place. Thus, when the probability of decoherence or dephasing of a qubit is calculated by the channel following the analysis presented in this section, the time elapsed from the start of the execution and up to the current point in the circuit is taken into account by the implemented function.

4.3.4 Unifying the Quantum Channels

After the above detailed expression of the individual quantum channels that model each source of noise, the next step is to define a noise model that combines these quantum channels in a single construct. This new model is dubbed the *unified noise model* (UNM), as, aiming to approximate the noise within the quantum computer as closely as possible, it combines (or unifies) the three distinct and independent error groups. This model is, therefore, able to simulate the evolution of a quantum system within a NISQ machine while considering all the aforementioned sources of error. Furthermore, as shown by the experiments outlined in Section 4.4, the unified model is able to recreate the behaviour of a quantum computer better than other state-of-the-art models.

The main characteristic of the UNM is its architecture awareness. The model is implemented with the ability to take into account the connectivity of the qubits within the architectural graph of the computer, as well as the specific properties of the qubits (i.e., decoherence time) and the gates (i.e., execution time, error rates) that participate in the system in the form of noise parameters, as defined in Section 4.2. In other words, one can encode within the UNM essential knowledge about the position of the qubits in relation to each other and the levels of noise during the execution of a quantum circuit.

Constructing the Depolarising Model

The logic followed to construct the depolarising model defined earlier in this chapter is pretty straightforward. In general, a circuit executed directly on a quantum computer includes either single- or two-qubit gates. Larger constructs like, for example, the three-qubit Toffoli gate, might be supported within Qiskit, but are implemented as a decomposition of the larger gate to smaller, single- or two-qubit gates. Considering this, a depolarising model that can be applied to any sort of gate within the circuit can be constructed according to the following rules:

1. Single-qubit errors occur after a single-qubit gate in compliance with the single-qubit error rates calibrated at the time of the experiment on the real quantum computer. In other words, after every single-qubit gate the depolarising channel is applied and induces an error according to the rules of error group 1.

2. Two-qubit errors occur after a two-qubit gate according to the two-qubit error rates, calibrated along with the single-qubit error rates. Like for the single-qubit gates, after a two-qubit gate is applied the depolarising channel follows, introducing quantum noise according to the rules of error group 1, but extended to a 4×4 operator dimensionality. Here, in the context of an architecture-aware model, the knowledge of the qubit connectivity within the QPU is encoded within the model. This means that the model "knows" which qubits form pairs, as well as the error rates between the pairs.

Constructing the SPAM Model

This model is essentially a simplified version of the depolarising model. Essentially, if a state preparation occurs at any point during the computation, it is followed by the SPAM channel modelling state preparation errors. Likewise, the SPAM channel is applied before the measurement, modelling the readout errors. The likelihood of each respective error is defined by the respective error rates per qubit, calibrated at the time of the experiment.

It is important to emphasise a bit further on the main idea behind separating the state preparation and measurement operations from the rest of the quantum circuit. On IBMQ, the state of a qubit (or, in more general terms, a qubit register) is prepared by injecting the standard initial state $|0\rangle$ to the qubit (or $|0^{\otimes n}\rangle$ for all the n-qubits in the register). Of course, arbitrary quantum computations might start with a different initial state, which would require alternative operations for its preparation. As a simple example, consider a quantum walk on four qubits initialised on state $|0110\rangle = |6\rangle$. Thus, it appears to be a logical decision within the framework for modelling adopted during this research, that the operations preparing the qubits (or quantum registers) should not be part of the main body of the quantum circuit that executes the algorithm. Similarly, the reason for choosing measurement as a separate quantum operation is self-explanatory. Finally, grouping the two operations together in the same quantum channel, and equivalently, the same model, comes naturally, as is evident from the above, both processes are described and modelled in the same way.

Constructing the Relaxation and Dephasing Model

Following the above trend of architecture awareness, it is essential for the relaxation and dephasing times to be considered for each individual qubit that participates in the circuit. The thermal relaxation and dephasing model can be constructed according to the following rules:

1. Incorporate a function that introduces thermal relaxation to each of the qubits in the quantum system. This is implemented after each gate is applied and occurs according to the relaxation time of each qubit in the system, as well as the duration of each type of quantum gate within the system and the overall duration of the computation up to this point.

2. Incorporate a function that introduces dephasing in the quantum circuit in a similar manner as the thermal relaxation.

Constructing the Unified Noise Model

Now that there is a clear framework that allows the creation of the individual noise models from their respective quantum channels, the unified quantum noise model can be constructed as the combination of the three individual noise models. The application of every quantum channel is independent and their combination is simply computed by composing the error operators with the circuit gates. In the simplest case of an one-qubit system, assuming an arbitrary number t of unitary, single-qubit quantum gates U_t, and an initial quantum state ρ_0, the effect of the unified noise model on the evolution can be expressed as

$$\mathcal{V} = M \cdot \mathcal{S} \cdot \prod_t \Big(\mathcal{N} \cdot \mathcal{D} \cdot \mathcal{U}_t \Big) \cdot \mathcal{N} \cdot \mathcal{S}' \cdot P(\rho_0) \tag{4.8}$$

where $\mathcal{U}_t(\rho) = U_t \rho U_t^\dagger$, $\mathcal{D}$, $\mathcal{S}$, $\mathcal{S}'$ and $\mathcal{N}$ are the depolarising, measurement, state preparation, and relaxation and dephasing channels respectively, M is a measurement superoperator and P is a state preparation superoperator. This definition can be readily expanded to account for higher dimensional operators (e.g., for two-qubit operations) on larger quantum circuits. Figure 4.1 visualises the unified quantum noise model for the single-qubit case.

Finally, it is noteworthy how the model treats single- and two-qubit gates differently when the depolarising and relaxation and dephasing channels are applied. After each gate in the circuit, the two channels occur independently of each other and can be combined by composition. Figure 4.2 visualises the effect of the channels on each type of gate. More

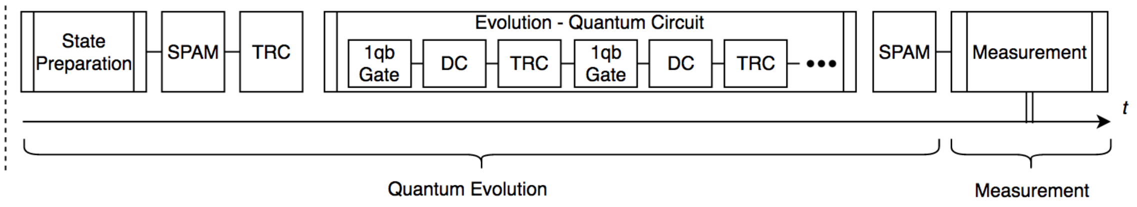


Fig. 4.1 The unified quantum noise model on a single-qubit circuit. The SPAM model is applied at the start, after the state preparation (if that occurs) and at the end before the measurement. The depolarising channel (DC) is applied after every gate during the evolution of the circuit. The relaxation and dephasing channel (TRC) is applied after every gate and after the depolarising channel.

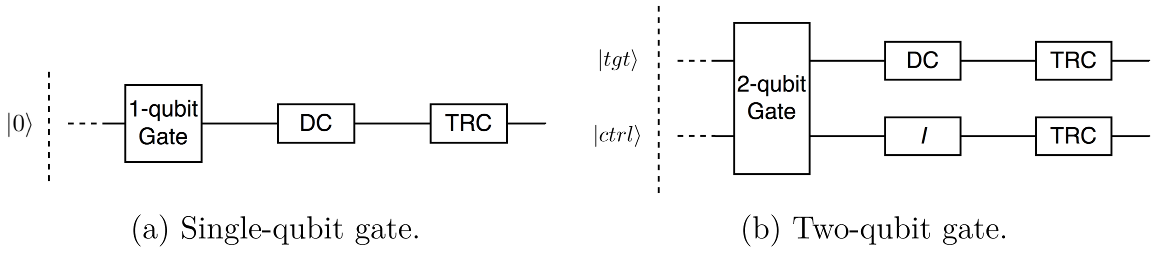


Fig. 4.2 (a) Application of depolarising (DC) and relaxation and dephasing (TRC) channels on a single-qubit gate. (b) Application of the two channels on a two-qubit gate; $|ctrl\rangle$ and $|tgt\rangle$ are control and target qubits; the channels are applied independently on each qubit and I is the identity, which is considered a virtual gate with zero execution time.

specifically, in the two-qubit gate, it can be observed that only the target qubit is affected by the depolarising channel. This happens as, within the depolarising part of the UNM, the part of the operation that has a chance to go wrong is the “state change”. In other words, the control qubit acts just as a driver of the quantum gate, and thus it is assumed that the gate has no effect on its state, either willingly or through the effects of noise.

4.4 Simulating the Noisy Behaviour of NISQ Machines

This section showcases the various experiments and results from the simulation of the noise within some of the IBMQ quantum computers. An introduction to the machines themselves and the architecture of their QPUs can be found in Section 2.2.2. The unified quantum noise model and associated methods are implemented using Python and the circuits are simulated using the IBMQ Qiskit simulators [63]. Of course, this does not induce any difficulties in applying the model in different architectures that use QASM or QASM-type implementation for the low level quantum circuits. In other words, the noise model is not limited to modelling the IBMQ computers. Its architecture awareness and low-level circuit approach allows for it to be easily attached on any QASM-based implementation. The code is available on GitHub [113].

Preliminary Methods

Before moving on to showcasing the experiments, it is necessary to discuss some preliminary methods that assist in the experimental process and interpreting the results. First of all, the discrete-time and space quantum walk has been chosen as a method to carry out the experiments on the quantum computer, as introduced in Section 1.2.1. There are two main reasons that led to this choice: (i) the predictable behaviour of this algorithm that derives from the modularity property (see Definition 1.2), and (ii) the susceptibility of the quantum walk to noise and the fact that the effects of the noise are easily visible through the violation of the modularity property. For the implementation of the quantum walk circuit, the gate-efficient approach that uses generalised inverters is utilised, as was analysed in Section 3.2.1. The number of gates in the aforementioned quantum circuit increases with the size of the circuit, N, as $\mathcal{O}(\log^2 N)$, proven through equation (3.18).

Moreover, in order to carry out a quantitative analysis of the noisy behaviour of the quantum machine and test the performance of the models (UNM or other models it is compared to), it is necessary to use a metric that is able to compare the results of the simulated noisy evolution and the execution on the quantum computer. For this purpose, the Hellinger distance is used, defined as follows [114].

Definition 4.1 (Hellinger distance)**.** *For probability distributions $P = (p_1, \ldots, p_k)$, $Q = (q_1, \ldots, q_k)$ and k finite or countable, the Hellinger distance between them is defined as*

$$h(P, Q) = \frac{1}{\sqrt{2}} \sqrt{\sum_{i=1}^{k} (\sqrt{p_i} - \sqrt{q_i})^2}. \tag{4.9}$$

The Hellinger distance is a metric satisfying the triangle inequality. It takes values between 0 and 1 (i.e. $h(P, Q) \in [0, 1]$) with 0 meaning that the two distributions are equal. On the other hand, a Hellinger distance of 1 would indicate the maximum possible distance between the two distributions, which could occur when, for example, they have disjoint support. In other words, the maximum Hellinger distance can be achieved when P assigns probability zero to every set to which Q assigns a positive probability, and vice versa. Additionally, the Hellinger distance is easy to compute, easy to read and it does not depend on the probability distributions having the same support. The last property is particularly useful since in many ideal output distributions of quantum circuits the probability mass is concentrated on a few states.

The Hellinger distance is derived from a family of statistical distances called the f-divergence that can calculate the distance between probability distributions. Other metrics do exist that could be used in this context, for example the Kullback–Leibler divergence or the total variation distance. Nevertheless, the Hellinger distance is the one preferred as the metric of the distance between two probability distributions for the rest of this thesis.

Finally, as described in Section 4.2, there is a number of parameters corresponding to each of the error groups simulated. In order for the architecture-awareness of the UNM to work, it is necessary to take into account the individual error rates and decoherence times

1-qubit Gate Errors	2-qubit Gate Errors	Measurement Errors	T_1 (µs)	T_2 (µs)
11.68×10^{-4}	3.17×10^{-2}	7.61×10^{-2}	56.15	56.01

Table 4.2 Average noise parameters for all the qubits of the IBMQx15 Melbourne machine used on the date of the experiments. 1qb Errors are the single-qubit gate errors, 2qb Errors the two-qubit gate errors, Meas. Errors the readout errors and T_1 and T_2 the average relaxation and dephasing times of all 15 qubits.

for each qubit participating in the quantum circuit, as well as for each pair of qubits. In other words, it is essential for the UNM to "know" the connectivity of the qubits within the QPU. Thus, the architectural graph (see Definition 2.5) of each QPU is given to the UNM at the start of the simulation. Furthermore, the error parameters that describe the error rates, as well as the decoherence and dephasing times, are provided to the UNM prior to the simulation. Table 4.2 shows the average for each category of error rates used during the modelling of the quantum noise, calculated on the date of the experiments.

Experimental Methodology

Having made a well-informed decision on the preliminary methods and metrics necessary for the following analysis, this section presents the experimental methodology. For both the experiments and the simulations trying to approximate their evolution, it is deemed that only one step of the quantum walk (i.e., one coin-flip) will be implemented, as the previous work done in Section 3.2.3 (published [61]) makes it clear that this duration is satisfactory both for errors to take place and for the behaviour of the quantum walk to evolve in a predictable manner (i.e., see Figure 3.8). In general, the quantum walks are initialised on the state $|0^{\otimes n}\rangle$, where $n = \log N$ is the number of qubits necessary to represent the size of the state space, N, of the walk. It is evident that, since the state of the system is initialised on $|0\rangle$, there will be no need for state preparation. This simplifies the experimental methodology and lowers the complexity of the models for the experiments.

The experimental methodology consists of 100,000 runs of the quantum walk, with the configurations described above, on the quantum computer and as a simulation. At the time of the experiments, the IBMQ systems had a reduced capacity on the number of circuit iterations within the machines (capped at 8,000). Thus, the circuit runs are broken down to 20 batches of 5,000 iterations each. The end result is simply the cumulative distribution arising from the individual results of each batch of iterations.

In order to better identify the performance of the UNM, an experimental evaluation follows of four additional noise models, with the simulations treated in the same way, i.e., try to approximate the same quantum walk experiment, with the same initial configuration and duration as the UNM. A brief introduction of the four models is shown below:

- **QiskitCM.** This is the most complex of the four noise models as it implements a simple combination of three sources of noise: a depolarising error modelling the gate/control and measurement infidelities and a thermal relaxation and dephasing error modelling decoherence and dephasing. The QiskitCM model is implemented within Qiskit[2].

- **DSPAM.** A simpler version of the UNM that includes the depolarising model for the gate infidelities and the SPAM model for the measurement and state preparation errors. In other words, it is the UNM without thermal relaxation and dephasing.
- **TRM.** The opposite of the DSPAM model, the TRM is a standalone thermal relaxation and dephasing noise model implemented within Qiskit that follows the main principles of Error Group 3 (Section 4.3.3). This model is also implemented within Qiskit.
- **SDM.** A simple depolarising model that is not architecture-aware, i.e., it does not take into account the qubit connectivity within the QPU. This model is also implemented within Qiskit, along with TRM and QiskitCM.

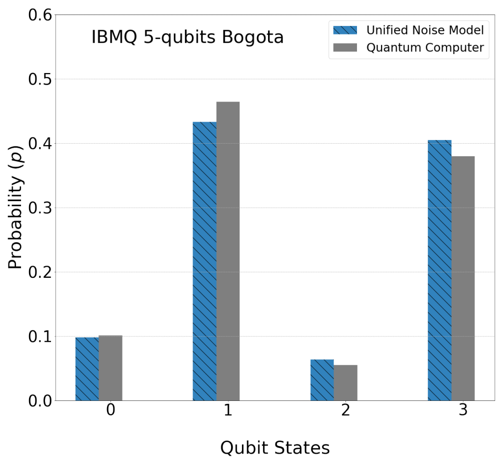
The QiskitCM model is clearly the more complex of the IBMQ noise simulators and shares similarities with the UNM on the way it computes the error. On the other hand, this model does not take into account the noise parameters for each qubit separately, but calculates and utilises their averages, a fact that is reflected through a larger deviation from the quantum computer distribution than the UNM (see Table 5.4). The DSPAM and TRM models are, essentially, a separate and simple implementation of Error Groups 1 and 2, respectively. Finally, the SDM model is just a simple depolarising model that is completely architecture-unaware, i.e., it does not take into account the connectivity of the qubits within the QPU, but instead, computes the noise through a simple probabilistic application of Pauli errors during the computation.

The result of simulating the noisy evolution of the quantum walk within the Melbourne computer using the UNM is then compared with the outcome of simulating the noise using each of the other noise models. A more detailed analysis of this process and the results are presented in the following section.

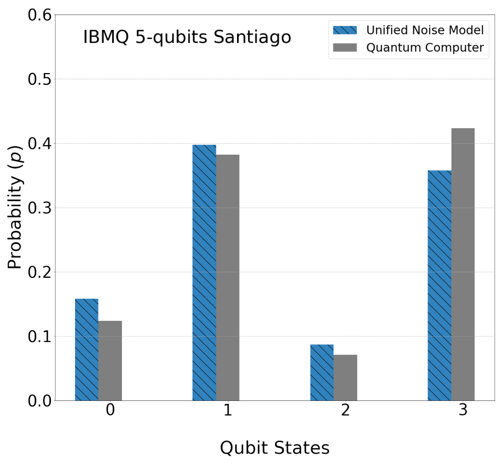
Experiments and Results

A number of experiments has been carried out using four of the IBMQ machines: the 5-qubit Bogota, 5-qubit Santiago, 7-qubit Casablanca and 15-qubit Melbourne computers (more information in Section 2.2.2). The Melbourne machine, with an architectural graph shown in Figure 2.2, is the largest publicly available quantum computer from IBMQ, showing a quantum volume of $V_Q = 8$. Unfortunately, this computer has been decommissioned as of August 2021. The former three computers, with architectures shown in Figure 2.3, are dedicated machines part of the IBMQ Researcher program. They are using the Falcon r4 QPU and exhibit a quantum volume $V_Q = 32$ [64, 41]. The evolution of the quantum circuit is also simulated using the UNM according to the characteristics of each quantum computer.

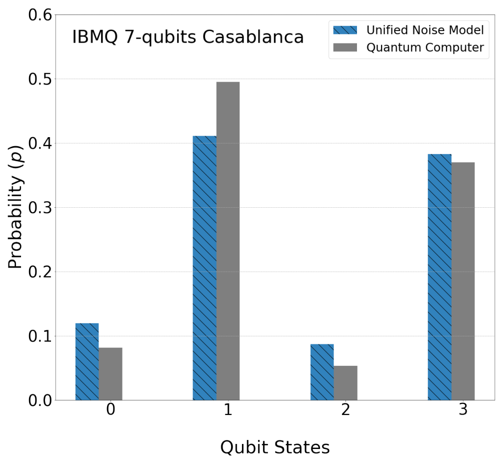
[2]More concrete description in Qiskit documentation: https://qiskit.org/documentation/apidoc/aer_noise.html



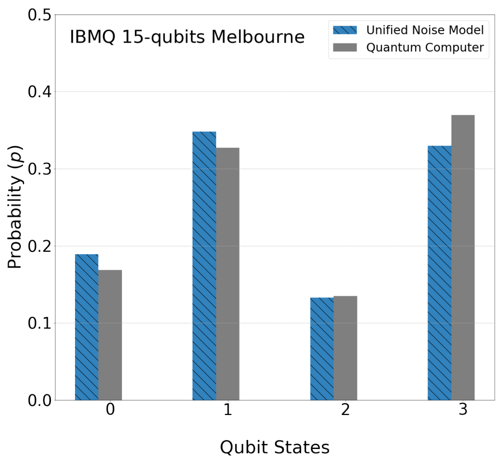
(a) IBMQx5 Bogota.



(b) IBMQx5 Santiago.



(c) IBMQx7 Casablanca.



(d) IBMQx15 Melbourne.

Fig. 4.3 Comparison of the probability distributions between the UNM simulations and the various quantum computers for a quantum walk on two qubits (i.e., $n = 2$).

After running the quantum walk experiments on the four quantum computers, the UNM is used to simulate the evolution within each machine. Figure 4.3 showcases the performance of the UNM when simulating the noise of each IBMQ computer. The executed circuit implements a two-qubit quantum walk (i.e., with a state space $N = 4$) in order to accommodate the restricted number of qubits on the smaller quantum computers. Table 4.3 showcases the Hellinger distances between the probability distributions resulting from each quantum computer and the corresponding UNM simulations. Evidently, all simulated evolutions are quite close to the quantum computers, with the Bogota machine exhibiting the smallest Hellinger distance, followed closely by Melbourne (0.025 and 0.033 respectively). The Santiago and Casablanca machines showcase a larger divergence from the simulated evolution via the UNM.

From this point forward, the experiments are focused on evaluating the performance of the UNM in comparison with the four noise models introduced above: QiskitCM, DSPAM, TRM and SDM. Due to the limited size of the architectures of the smaller computers (i.e., 5

Machine	Hellinger Distance	Runtime (sec)
IBMQx5 Bogota	0.025	240.7
IBMQx5 Santiago	0.054	234.2
IBMQx7 Casablanca	0.076	254.5
IBMQx15 Melbourne	0.033	262.3

Table 4.3 Hellinger distances between the various quantum computers used in the experiments and the UNM. Runtime is the execution time for 100,000 iterations of the quantum walk circuit on the corresponding quantum machine

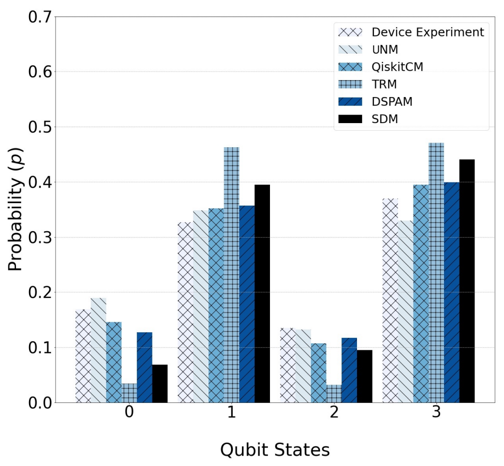
No. States	No. Qubits	UNM	QiskitCM	DSPAM	TRM	SDM	Ideal	Uniform
4	2	0.033	0.040	0.049	0.229	0.126	0.264	0.218
8	3	0.127	0.152	0.224	0.438	0.280	0.771	0.186
16	4	0.224	0.262	0.369	0.476	0.329	0.834	0.150
32	5	0.393	0.421	0.482	0.505	0.467	0.862	0.434
64	6	0.457	0.509	0.525	0.587	0.576	0.891	0.579

Table 4.4 Hellinger distance between the probability distributions of the quantum computer and the various noise models, as well as the ideal and uniform distributions. For ease of presentation, the acronyms are ascribed as UNM: unified noise model; QiskitCM: the Qiskit composite model; DSPAM: depolarising and SPAM; TRM: relaxation and dephasing model; SDM: simple depolarising model; Ideal: the theoretical distribution from an ideal (noise-free) quantum walk; Uniform: uniform distribution, for the maximum-entropy guess.

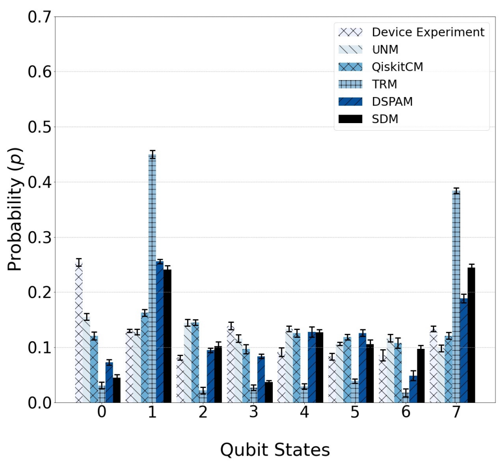
and 7 qubits), increasing the size of the state space of the walk causes the quantum circuits to quickly become larger than the computers can accommodate. Thus, for the sake of a more comprehensive experimental review, the following analysis focuses on experiments carried out on the 15-qubits Melbourne machine. The interest here lies in quantifying how close the evolution of each of the five noise models is to the quantum computer. This comparison will offer a comparative idea of the performance of the UNM in approximating the noise. For this purpose, after a number of experiments with increasing state space are executed, the Hellinger distance (HD) is computed between the distribution of each simulated noise model and the quantum computer.

As a reminder, the smaller the Hellinger distance between a noise model and the computer, the better the approximation of its behaviour and, hence, the more accurate is the model. In terms of a visual comparison, the better model is the one producing a distribution that is closer to the resulting distribution of the computer. The experimental results are visualised in Figure 4.4(a) which shows that, on a two-qubit system, the UNM provides a better approximation of the quantum computer's distribution than the other noise models. A numerical comparison of this result using the computed Hellinger distances is shown in Table 4.4, providing a clearer indication of the model performance.

The subsequent experiment repeats the above methodology for a quantum walk on three qubits. The respective results are shown in Figure 4.4(b). Again, the results showcase



(a) Two-qubits quantum walk.



(b) Three-qubits quantum walk.

Fig. 4.4 Comparison between the probability distributions of a quantum walk on (a) a two-qubit system and (b) a three-qubit system, simulated with the UNM (light-coloured vertically lined bar), QiskitCM (dark-coloured crossed bar), TRM (tiled bar), DSPAM (dark-coloured vertically lined bar), SDM (solid bar) and run on the actual quantum computer (light-coloured crossed bar). The quantum walk propagates for one coin-flip; error bars with 95% confidence intervals are shown for the three-qubit system; for the two-qubit system they are smaller than 10^{-3}, hence are not displayed.

the superiority of the UNM to the rest, with the smallest HD of 0.12749 from the computer. This value, and indeed the distances of all the models from the computer, are much higher than the corresponding figures for the smaller, two-qubits quantum walk. This is an indication that, in general, the models perform worse in approximating the noisy evolution of bigger quantum circuits. This is an expected result as, for longer circuits, there will be an increase in the errors and decoherence within the computer as there are a lot more operations and measurements that can produce an error and the execution time of the circuit is much longer, creating much higher chances of decoherence and dephasing. Such results bear deeper meaning and reflect the performance of the quantum computers, a topic that this thesis addresses in Chapter 6.

Providing a figure for a visual comparison on experiments on a quantum walk with a state space larger than three qubits has little information value, as the bar plots become very messy due to the volume of information (for example, there are 16 states on a 4-qubits state space and the simulations are done using five noise models – i.e., 80 bars in the plot). Thus, additional results from further experiments are only numerically shown in Table 4.4. It is clearly visible that the distance from the quantum computer's distribution is increasing with the number of qubits in the system, an expected result as explained above. This is true for all the models. Nevertheless, it is easy to realise that, according the Hellinger distances, the UNM is the best out of all the models, followed closely by QiskitCM.

As a final remark, it becomes apparent by the experimental results that the quantum computer, for quantum walks of size bigger than $N = 8$, produces probability distributions that are closer to the uniform distribution due to excessive noise. Such results can often be viewed as “garbage” with no real meaning as the evolution has simply become too distorted to bear any resemblance to the original algorithm. Nevertheless, these results are driven by the effects and intensity of the noise within the QPU, an indivisible part of what the NISQ era means for quantum computers, and thus, the results are deemed worthy to be included in this analysis.

A number of experiments using the UNM in the context of benchmarking have been carried out in Chapter 6. The results further showcase the effectiveness of the new model at approximating the behaviour of the respective quantum computers, sometimes with even higher accuracy. For better presentation, the relevant probability distributions are showcased in Appendix E. The code implementing the experiments can be found on Github [113].

4.5 Placing the Unified Noise Model in the State-of-the-art

The UNM is a model trying to offer an alternative approach to noise modelling and simulating in an era where such research is a priority for many fields. Thus, it is important to present a review of the contributions of this thesis in these terms, as well as place the UNM within the vast fields of quantum error correction (QEC) and benchmarking quantum operations.

4.5.1 Unified Noise Model vs Gate Set Tomography

One of the prominent protocols for characterising quantum operations is *gate set tomography* (GST) [115]. GST has been used in a large number of experiments [116–122] and implemented in open-source software [123, 124]. The basic aim of GST is to characterise quantum operations performed by hardware and allow one to estimate the performance of a system with a relatively small number of qubits. Additionally, it reconstructs or estimates not a single logic operation, but an entire set of logic operations (hence, gate set).

The characteristics of GST give rise to a meaningful comparison to the UNM. As is evident by the above sections, the unified noise model aims to reconstruct (or simulate) the entire quantum evolution of a circuit, from the starting point to measurement. It follows the quantum circuit execution at runtime and simulates the effects of noise on three levels, gate infidelities, state preparation and measurement and decoherence and dephasing of the qubits. On the other hand, the GST is aimed to predictive characterisation of the quantum gates of the circuit within the QPU, i.e., how the logic operations affect the qubits they act upon. The quantum gates need to be specified before the GST reconstructs the gate-driven evolution.

Furthermore, as mentioned earlier, the GST protocol works well with *small* quantum systems [124, 125], i.e., with a smaller number of qubits. On the other hand, the UNM aims to approximate the noisy evolution within a QPU irrespective of the size of the quantum system. Thus, whereas GST works well for two- or three-qubit systems, the UNM is designed to scale with the size of the quantum circuit and the number of qubits. Of course, there is an upper bound on the scaling capabilities of the UNM tied to the increasing difficulty of classical machines to simulate increasing number of qubits. Nevertheless, within the NISQ era, most of the computers that the UNM will be used on are relatively small (for example, within this thesis, the biggest available quantum computer is 15 qubits and can easily be simulated on a standard laptop).

Finally, one characteristic of the GST protocol is that it is calibration-free [125]. When GST reconstructs a model of a quantum system, it does not depend on any prior description of the measurements used or the states that can be prepared. The UNM, as is evident from the analysis above, depends on a set of quantum noise parameters, which reflect the levels of noise within the QPU during the execution of the quantum circuit. Thus, calibration of these noise parameters (or error rates) is essential for the success of the unified noise model, a reason that leads to the development of the noise parameter optimisation technique showcased in the following Chapter. Of course, such calibrations in no way limit the capabilities of the model or the runtime of the simulations as the noise parameters are readily available from the QPU providers (in this case, IBMQ).

4.5.2 Markovian vs Non-Markovian Noise

In a recent study [126], the authors simulate the relaxation of stationary states at different frequencies and on different quantum computers in order to study the spectroscopic behaviour of their noise. In other words, they aim to study the absorption or emission of light or other radiation that leads to decoherence during the computation. The study of non-Markovian quantum system dynamics attempts to provide a solution to fundamental problems on how to define and quantify memory effects in the quantum domain or how to exploit and develop applications based on them. Further studies examine its use in order to find the ultimate limits for controlling open system dynamics. Hence, the simulations of the non-Markovian noise effects are based on the concept of quantum memory, i.e., the idea that the time evolution of the conditional probability of the environmental noise is governed by a generalized master equation depending on the environmental memory effect [127].

The results find that the noise follows non-Markovian behaviour. They also suggest that quantum computers can be modelled as non-Markovian noise baths and analysed through simulations, thus providing interesting potential applications on error-mitigation. Within this thesis, the unified noise model describes the noisy evolution of the quantum circuit through a probability space of possible states resulting from the effects of the noise during the computation. In other words, a highly Markovian approach is employed to simulate the noise and decoherence for all three quantum channels. As is evident by the results, this memoryless approach produces a satisfactory performance, especially on small quantum systems.

4.5.3 Noise Modelling in Quantum Error Correction

One of the most prominent fields in the development of quantum computing is *quantum error correction* (QEC). Seeing how the noise is the major crippling part of near-term quantum computers and the prime obstacle of universal quantum computing, research around error correction aims to fight noise by, as the name suggests, correcting errors that appear during the evolution of the quantum circuit. Within QEC, as is expected, there is a large amount of literature and research along the lines of error analysis with NISQ systems. Furthermore, modelling the noise appears to be an essential part of successful error correction. This leads to the aforementioned evidence that a more accurate noise model is valuable for the field of QEC.

Similarly to the research presented in this thesis, noise within a quantum computer is categorised in coherent systematic gate errors, environmental decoherence and models of loss, leakage, measurement and initialisation errors [128, 129]. Systematic noise contains any errors caused by faults of the quantum gates themselves, much like the gate infidelities within the UNM. Environmental decoherence tries to highlight how QEC relates to environmental effects. An elegant model for characterising decoherence on open quantum systems is the Lindblad formalism [130, 30, 131], accompanied by several assumptions that may not hold in some cases [132–135]. Particularly in superconducting systems where cross-talk and fluctuating charges can cause decoherence, the need arises for more specific decoherence models. One way to construct such models is via more general mappings [129], or alternatively, a combination of models like the UNM presented in this work.

A recent paper [136] discusses a structure for QEC which relies, amongst others, on evaluating noise modelling techniques or combinations of them. According to the authors, such models have the potential to provide a much more accurate and efficient approximation of the noise within quantum computer, a result which would benefit QEC. The UNM is a perfect fit for such an approach, as by default, it combines the major sources of error that play a significant role in quantum error mitigation.

Finally, it is evident that more complex or expanded quantum channels have the ability to better recreate the quantum noise before any type of QEC is applied [137]. Hence, future incorporation of further types of systematic noise, like Clifford errors, or environmental decoherence, like electromagnetic noise, to the UNM, can lead to an even more accurate model for quantum noise.

4.6 Discussion

This chapter introduced a new model for simulating the noisy behaviour of quantum computers, named *unified noise model* (UNM). The UNM combines three quantum channels that model three main error groups: (i) gate infidelities, (ii) state preparation and measurement (SPAM) errors and (iii) thermal decoherence and dephasing of the physical qubits. Each noise source adheres to different aspects of the hardware and of the interaction between the system and its environment and incorporates some of the most prominent sources of noise during a computation. The noise model is architecture-aware, meaning that it takes into account the connectivity of the qubits and the individual characteristics and levels of noise (i.e., noise parameters) of each qubit within the QPU.

Throughout the experimental methodology, the performance of the UNM is compared with the evaluated efficiency of four other models. The analysis shows that the UNM offers more accurate approximations of an IBM quantum computer's evolution and significant improvements in the accuracy of the noise simulations. Furthermore, the UNM is not limited to only modelling IBMQ computers. Its architecture awareness and low-level circuit approach allows for it to be easily attached on any QASM-based implementation.

Noise is one of the main challenges preventing universal and scalable quantum computation. This work has shown that unifying noise sources on a single model results in a better approximation of the noisy evolution of a quantum computer. Finally, this approach to noise modelling can assist with the understanding of noise within quantum computers and consequently be utilised during the design or testing of error correcting methods or calibration techniques and attempts to minimise the noise in near-term quantum computers.

Surprisingly, there are only a few works addressing the topic of modelling noise in quantum computers [138–140]. Notably in [141], the authors also attempt to generate a composite model for noisy quantum circuits by dividing the quantum circuit to subcircuits, according to desired characteristics. This decomposition allows for iterative adjustment of the models through minimisation of the total variation distance between simulation and experimental results, until sufficient accuracy is obtained. Thus, the work done in this chapter represents a valuable addition to the field of quantum computation.

In conclusion, having shown the importance of being able to understand and model the noise within a quantum computer, it is apparent that the work presented in this chapter is invaluable and with a wide application space.

4.7 Future Work

Whereas the UNM excels at simulating the noisy behaviour of near-term quantum computers compared to other state-of-the-art noise models, the elevated Hellinger distance between the UNM and quantum computer distributions shows that there might still be elements missing from the model. One factor is that further known and/or potential sources of noise are not taken into account, for example electromagnetic or gravitational decoherence. Additionally, it is known that the calibrated noise parameters fluctuate, potentially with significant effects on the results, even throughout the duration of the experiments.

The above points, as well as the deviation of the HD is an additional indicator of the lack of understanding of quantum noise within the field. In the remainder of this thesis, a chapter is dedicated to engineering a way that brings the simulated noisy evolution of the UNM and the real evolution within a quantum computer much closer.

Following the above, some clear future additions to the UNM could include the consideration of further sources of error. One example is the incorporation of electromagnetic decoherence as an important noise factor. It is believed that this source of decoherence will make up for a large portion of the deviation between the UNM and the quantum computer distributions. Finally, further errors from the Clifford group [84] could be taken into account as an additional interpretation of gate infidelities during the evolution of the quantum system.

Chapter 5

Quantum Noise Parameters Optimisation

The previous chapter presented the unified noise model (UNM) which combines some of the major sources of noise during a quantum computation in order to simulate the noisy evolution of NISQ systems. The sources of error modelled within the UNM derive from operation errors during the circuit execution, decoherence (of the thermal form) and dephasing of the physical qubits. As discussed in the same chapter, a crucial role for the efficiency and functionality of the UNM is played by the noise parameters. These parameters represent numerically the various error rates and decoherence and dephasing times that result from calibrations of the actual quantum computer whose behaviour is being modelled (more information in Section 4.2). In other words, they reflect the levels of noise within the quantum computer during the execution of a circuit.

The parameters used for the experiments implemented in Chapter 4 are calibrated near the time of the execution of the circuit on the quantum computer using benchmarking techniques. As evident from the results of Chapter 2, the approximated evolutions resulting from the noisy simulations of the UNM deviate from the actual evolution of the quantum computer. Motivated by said deviation, this chapter showcases and implements a novel methodology that can be used to optimise a subset of the calibrated noise parameters that take part in the modelling and simulating of a circuit's evolution. The techniques used for this purpose are classical optimisation routines, and more specifically, genetic algorithms [142]. The simulations that utilise the optimised parameters resulting from this process ("post-optimisation noise parameters" or "optimised noise parameters") show a significant increase in the efficacy of the simulated evolution that can reach up to 84% in small systems.

The first section presents a comprehensive method that uses a classical genetic algorithm routine in order to optimise the noise parameters used by the UNM. Following the establishment of the framework, the various experiments and results from new simulations with the optimised parameters are presented, showcasing the effectiveness of the optimisation method and the increase in efficiency of the UNM. Finally, the findings of the work outlined in this chapter are discussed and future additions are also presented.

The work outlined in this chapter has been jointly published with Chapter 4 in Physical Review A in December 2021 [85].

5.1 Genetic Algorithms for Quantum Parameter Optimisation

As aforementioned, the noise parameters used at the time of the noisy simulations of the unified noise model are calibrated from the quantum computer itself during the execution of the quantum circuits whose noisy evolution is examined. This can be done using various techniques, as showcased in Section 4.2. Of course, putting fluctuations of the noise parameters over time aside, these techniques are not always accurate themselves, and errors on the resulting calibrated parameters can be expected. This section discusses the classical methodology that allows to optimise the hardware-calibrated noise parameters for the UNM and mimic the evolution of the quantum computer much more closely, rectifying the deviations between the UNM evolution and the real quantum computer.

Notably, such an optimisation procedure is possible for the UNM as the structure and implementation of the model itself is independent from real-time execution of the circuit, thus allowing the altered parameters to be fed to the model before a simulation. This is not possible when using other models implemented in the Qiskit provider API, like for example the QiskitCM model, as such methods automatically draw the hardware-calibrated parameters from the quantum computer itself and then construct the model offering no user control over the procedure. To avoid confusion, it is noteworthy that this is not a mathematical or mechanical discrepancy of the model design itself, but rather a limitation of Qiskit at the time of the creation of the UNM.

As shown in Chapter 4, there is a large number of noise parameters associated with each of the error groups within the UNM, and their number grows with the size of the state space of the implemented quantum process. For example, a quantum walk on a two qubit state space (using the generalised-inverter approach for the implemented circuit) will only use three qubits in the QPU, whereas a three qubit quantum walk, accounting for the necessary ancilla, will need six qubits. Within the scope of this thesis the parameters considered are the ones that correspond to the control and operation error rates, i.e., the gate, measurement and state preparation errors that occur during the execution of the circuit. The size of this set of parameters (i.e., the number of parameters) can be calculated as $r + m + s$, where m is the number of qubits in the system that are measured, s the number of qubits that undergo state preparation and r the number of different types of gates that participate in the circuit. More information on the types and numbers of parameters can be found in Table 4.1.

The parameters optimised correspond to the ones used within error groups 1 and 2 (shown in Sections 4.3.1 and 4.3.2 respectively). Importantly, the decoherence and dephasing parameters used within the model corresponding to error group 3, i.e., the decoherence and dephasing times $T_1(q)$ and $T_2(q)$ of each qubit q, are excluded from the optimisation routine. This decision is taken for two main reasons. First of all, in order for the simulation to be remotely close to the quantum computer evolution, these time parameters have to be considered individually per qubit q in the system. This means that, with the increasing number of qubits necessary for the workspace, the number of parameters for optimisation grows very fast, rendering the routines exceedingly taxing on the classical computer. Secondly, the relaxation and dephasing of the qubits are parameters tied with the physical implementation of the qubits themselves within each quantum computer. Thus, an optimisation of those parameters would produce better results when implemented on a qubit engineering modelling level that takes into account further environmental and/or physical factors, for example, cross-talk between the qubits or the effects of active qubits to idle qubits, conditions that are not taken into account by the UNM.

Furthermore, as shown from the results, optimising the operation error rates produces an improved approximation of the quantum computer evolution, especially on small systems. On the other hand, including the decoherence parameters in the optimisation does not offer enough of an advantage to justify the increased complexity of the routine (see Section 5.2). Of course, there is nothing preventing the optimisation of the relaxation and dephasing parameters in the future and specifically on larger quantum system, especially if a sufficiently large classical system is available to accommodate the increased complexity and runtime of the genetic algorithm.

For the additional experiments with the optimised parameters, the same quantum walk circuits are used as in Chapter 4. Of course, any quantum circuit of arbitrary quantum algorithms can be easily simulated and optimised with the implemented methods (both the UNM and the optimisation routines). The quantum walk implementation is the generalised-inverters circuit (Section 3.2.1) which includes Hadamard, `NOT` and `CNOT` gates. Important here is that every gate will be considered for the parameter count only once per qubit or pair of qubits, no matter how many times it is used in the circuit. The reason for this follows from the assumption that the noise parameters remain static during the execution of the quantum circuit (see Section 4.2). Thus, the number of parameters that need optimising in the case of the quantum walk circuit is $(1 + r_c + r_t) + m + s$, where m is the number of qubits measured, s the number of state preparations, r_c is the number of inverter gates, r_t the number of `CNOT` gates and $+1$ for the Hadamard gate. It is noteworthy that, due to tiny differences in the error rates of single-qubit gates, the differentiation between Hadamard and inverter gates can be omitted without needing to optimise both types. Thus, the number of parameters can be calculated as $r_s + r_t + m + s$, with r_s being the single-qubit gates in the circuit.

In order to obtain a set of better parameters, the methodology is based on *genetic algorithms* (GA) [142]. This method relies on iterative generations of new parameters, followed by simulations using said new parameters and comparison of the simulated results with the original distribution of the quantum computer. In each iteration, the best parameters of the current batch are chosen as those which bring the simulated evolution closer to the quantum computer, and are consequently preferred and reused in the next iteration. In order to keep the execution time small and the results presentable, a quantum walk with a small state space of $N = 4$ and a three-qubit system for its execution is used for the analysis. In the circuit, the third qubit is necessary for the quantum coin. The coin is never measured, meaning the results of its error rate optimisation will not be directly visible, but will nonetheless affect the overall computation. The following sections offer a concrete presentation of the optimisation methods.

5.1.1 Preparing the optimisation Routine

It is instructive to first offer a more detailed view of the methodology used to achieve the aims. Genetic algorithms (GA) are evolutionary algorithms in which random changes are applied to a current set of parameters in order to get a new, optimised set that performs better on a certain task. The current set of parameters is called a *solution*. These random changes are applied recursively until the best (set of) solution(s) is obtained. It is important to present here a brief introduction to the language and terminology associated with the study of GAs and that will be used during this chapter.

- *Parameter:* a single variable in the system of interest; in the context of this study, a parameter is a single error rate, i.e., a single-qubit or two-qubit error rate.
- *Solution:* otherwise called an *individual*, this is a set of parameters that assemble the items that are of optimisation interest; for the studied case, a solution contains all the error rates, single-qubit plus two-qubit plus measurement plus state preparation ($r_s + r_t + m + s$).
- *Fitness value:* a metric that encompasses the performance of a solution or set of solutions on the initial task, i.e., the simulation of the noisy evolution through the UNM.
- *Generation:* the current set of parameters for which there is sufficient information regarding their performance and fitness values.
- *Population:* the number of solutions within the current generation.
- *Chromosome:* a distinct, individual encoded item within a set of solutions, or in other words, a solution that bears a specific encoding; a chromosome encodes within itself all the parameters that are of optimisation interest. The process used to encode a solution and create a chromosome is discussed later in this section.
- *Gene:* an individual characteristic of interest (i.e., a singular error rate) encoded within a chromosome; in other words, a gene is the encoded form of a parameter.

Each chromosome is accompanied by a fitness value which shows how good this chromosome or solution and, in extent, the genes or parameters associated with it are for their intended purpose. The fitness value is the result of a fitness function and it is a measure of the performance of the solution when executing the task that it has been

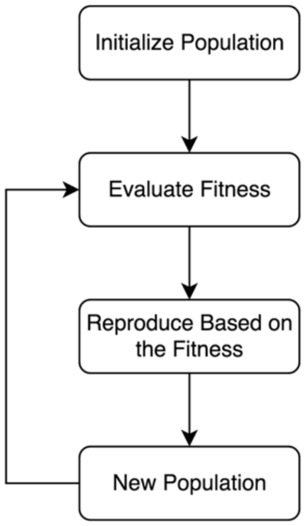


Fig. 5.1 General flow of a genetic algorithm. The initialisation process contains preparation steps like parameter encoding or producing the initial population. The reproduction step summarises various key processes, like crossover and mutation.

optimised for, i.e., during the UNM simulation. Following the evaluation and assignment of the fitness value, the best individual of a population is selected according to their fitness, in order to generate what is called a mating pool, where the higher quality individual has a higher probability to be selected. Thus, as time goes on, selecting the best candidates and eliminating the bad will lead to an optimal or acceptable solution.

The general flow of a GA routine is shown in Figure 5.1. The process is initialised by encoding the initial population in a suitable format. The next step is to evaluate the performance of the initial population, which will be the initial metric or, in a sense, the fitness value to which every solution in the new generation will be compared in order for a better individual to be picked. When the initial fitness is computed, the current generation will "reproduce" using processes that will be examined below, and from the resulting individuals, the new population will be selected.

Parameters Encoding

The first part of the GA routine refers to selecting a suitable way to represent and encode the parameters during the optimisation, also called *parameter encoding*. This is a process of great importance for the efficiency and success of the algorithm. The result of the parameter encoding is a chromosome which holds the desired characteristic and all the information about the parameters that are necessary for the optimisation routine. One of the better ways to encode the parameters, as well as provide convenience for the future steps of the algorithm (for example, during the reproduction process), is to use binary representation for the genes of the individual.

For the specific needs of the quantum noise parameter optimisation implemented within this thesis, the chromosome is encoded as a binary string that contains the binary

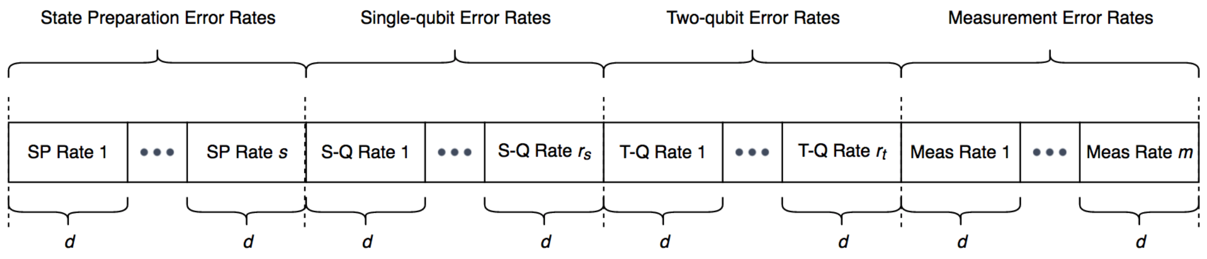


Fig. 5.2 The format of the encoded error rates, or genes, that constitute the chromosome. All the rates are translated into their binary representation and have the same length d, which is the number of bits needed to represent the largest error rate. The number of state preparation, single-qubit gate, two-qubit gate and measurement error rates is s, r_s, r_t and m respectively.

representation of all the error rates. These are, essentially, probabilities of an error occurring, i.e., numbers between 0 and 1. Each parameter within a solution is encoded to reflect the binary form of an error rate (creating the encoded gene) and is then appended to the chromosome until all the binary encoded genes are added. The genes within the chromosome appear sequentially in a specific order, which is crucially kept the same throughout the entire process: first are the state preparation error rates (if any), followed by the single-qubit error rates (if any), then the two-qubit error rates (if any) and finally the measurement error rates. Knowledge of the sequence of genes within the chromosome is necessary for the next steps of the algorithm, i.e., the fitness evaluation, as will be made clear later on. A visual representation of the parameter encoding process is shown in Figure 5.2.

As mentioned before, all the parameters represent probabilities and they are real numbers between 0 and 1. This means that their binary representation could very quickly become troublesome to work with, as real numbers can potentially be hard to visualise and manipulate in binary compared to integers. In order to deal with this problem, all the error rates are multiplied by a sufficiently large scalar, which remains the same for the entire procedure. Thus, the parameters become integers which are much easier to work with. The most commonly used scalars, depending on the magnitude of the smallest error rate, are 10^5, 10^6 or 10^7. Wherever the parameters need to be decoded during the genetic algorithm routine (for example during the fitness evaluation or when returned at the end of the optimisation routine), they are divided by the relevant scalar and, thus, return to their original decimal form.

As a brief example, consider the set of parameters whose average is presented in Table 4.2 – these are the average noise parameters during the execution of the noise experiments in Chapter 4. Here, the smallest single-qubit error rate (which are usually the operations that occur with the lowest probability) was 9.82×10^{-4}. Thus, in order for this number to become an integer, the scalar chosen is 10^6: $9.82 \times 10^{-4} \times 10^6 = 982$. Multiplying all the error rates (not including T_1 and T_2) with this scalar will result to all of them being integers, similarly as the above. When the need arises for the parameters to be decoded, they are all divided by 10^6, thus returning to their original form. Alternative approaches to solve this issue could be used, like for example in the form of an inversion of the parameter instead of multiplication with a scalar.

Finally, for convenience during the decoding of the parameters, all the genes are forced to have the same length, namely d, i.e., they are all encoded using the same number of bits. This is achieved by enforcing all genes to adopt the length of the largest value within the set. This transformation also means that the length of the chromosome is equivalent to the number of bits of the biggest binary representation multiplied by the number of parameters for optimisation, i.e., $(s + r_s + r_t + m) \times d$, where s, r_s, r_t and m are the number of state preparation, single-qubit gate, two-qubit gate and measurement error rates respectively.

Initial Population

The next step of the optimisation process is to create the initial population, which is the first population provided as input to the genetic algorithm. The initial population contains the hardware-calibrated operation error rates, measured at the time of the experiment, encoded as a binary chromosome in the way discussed above. The size of the population, n_p, corresponds to how many individuals (or chromosomes) the population contains and it may vary between experiments. It is also noteworthy that the usual practice is for the population size to stay the same during the evolution of the genetic algorithm that optimises the parameters for a quantum circuit of a static size. Thus, in the end, the best set of parameters will be picked out of a pool of size equal to the initial population size.

Alongside the chromosome containing the calibrated parameters, the population will have an additional $n_p - 1$ chromosomes generated via the addition of small random values to the calibrated parameters. The role of these “ancilla” chromosomes is to simply fill in the initial population set and, after evaluation, provide an indicative fitness for those randomly generated parameters that will aid in the next steps of the GA. If these randomly changed initial chromosomes prove to be better than the hardware-calibrated parameters, they have a higher chance of being chosen by the GA in the following steps. It is crucial for these random values to be of the same magnitude of the respective error rates they will affect, otherwise they might change the error rates dramatically. For example, for the single-qubit error rate discussed above (9.82×10^{-4}), the random number could be generated through a stochastic process with mean 5×10^{-5}. On the other hand, for the two qubit error rate it would be a process with mean 5×10^{-3}.

Finally, there is the option to set the range of values within which the algorithm will look for new parameters. This is not a necessary utility, but it helps with execution time if there are indications of an approximate range within which the parameters might be optimised. Another issue that this option resolves is, in the case of a physical system, there might be physical boundaries on the parameter values. For example, one equation that helps pick the range for the parameters is

$$P = \frac{c}{2^b - 1} \left(P_{\max} - P_{\min}\right) + P_{\min}$$

where c is the decimal representation of the binary string encoding of the chromosome (initial population) and b is the number of bits in the chromosome.

Fitness Evaluation

Perhaps the most crucial operation within the GA routine is the *fitness evaluation.* This process essentially provides a meaningful comparison for the efficiency of the evaluated individual in relation with the previous best candidate. It assigns a value to the chromosome, also called the *fitness value*, according to its efficiency for executing the task at hand. This value acts as a metric according to which the best, optimised individual is chosen in each iteration of the GA and also at the end of the optimisation.

The fitness value is usually computed via a fitness function, which matches the needs of the optimisation routine. For the evaluation process within this work, the fitness value is chosen to be the Hellinger distance, as defined in Definition 4.1 and computed by equation (4.9). This metric is calculated between the experimental distribution resulting from the quantum computer experiments (i.e., the evolution whose approximation the GA attempts to optimise) and the distribution resulting from noisy simulations of the UNM using as noise parameters those encoded within the individual that is currently under evaluation. The HD is computed for every individual in each population generated in every iteration of the GA. This means that a number of simulations of the UNM that equals the size of the population need to be run in every fitness evaluation during the optimisation routine, making this process the most taxing part of the GA.

For the present research, it is desirable that the noisy simulation distribution is as close to the quantum computer distribution as possible. In other words, the aim of the optimisation routine is to chose an individual that *minimises* the (Hellinger) distance between simulated and real evolution. This choice is reflected within the fitness evaluation by implementing the process so that the individual with the smallest Hellinger distance is picked in every iteration. Thus, during the course of the GA, successive selections of individuals baring the minimum fitness value (following a set of rules discussed in later sections) leads to the reduction of the distance between simulation and experiments, or in other words, to a better approximation of the quantum evolution within the machine.

Selection

Following the evaluation of the fitness of each chromosome/individual and the assignment of a fitness value as discussed above, the *selection* process refers to choosing which individuals (i.e., sets of error rates) will be used to generate the new population. Those individuals (also called parents) are selected from the current population pool based on their fitness value as assigned to them during the evaluation process. For this case (i.e., for the fitness value representing the Hellinger distance) the smaller the fitness value, the higher the probability of the individual being selected. This process ensures that the most suitable candidates are used to produce the next population.

There is a number of methods that can help pick the best suited parents during the selection procedure, such as roulette wheel, rank selection, tournament selection and more [142], each exhibiting different characteristics. The present work uses a version of the tournament selection method, called k-tournament selection, which randomly selects k individuals from the current population, with $1 < k < n_p$, and then selects the best out of these k individuals to become a parent. The same process is then repeated for selecting the next parent, and so on. The advantage of the tournament selection for the purposes of this research is the fact that it excels at choosing parents for minimising fitness values, making it also extremely popular in the GA literature. Additionally, it is efficient to code.

Recombination

Following selection of the parents, the final step of the GA corresponds to creating the new population for the optimisation routine, or otherwise, creating the individuals that amass the next generation of the GA. This new population will then be fed into the next iteration of the routine for fitness evaluation, and so on, following the processes described above. Ideally, every new generation will contain some chromosomes that perform better than the previous generation.

The main objective of the recombination step is to utilise the parents selected as the best fitted for reproduction in order to create the individuals of the new population. At this point, any individual produced from the recombination process is commonly called a *child*. During the recombination there are two important processes that take place: (i) the crossover and (ii) the mutation of the child. The crossover process generates a new individual by transferring part of the parent's chromosome to the child. The selection of transferable part occurs randomly whereas the size of the transferable part can be chosen as a percentage of the parent's chromosome. The mutation process refers to random changes occurring to the new individual. For the binary encoded chromosomes used during this chapter, the mutation method is essentially a simple bit-flip. For each new child, there is a chance that one or more bits of its binary representation are randomly selected and changed.

Each of the methods of crossover and mutation occur with specific probability during the optimisation routine. There have been several studies for optimising the selection of crossover and mutation probabilities [143–145]. For the algorithm implemented in this thesis the probability of crossover is $p_{\mathrm{co}} = 0.6$ and mutation $p_{\mathrm{mt}} = 0.05$. A figure showcasing the effect of these two operations is shown in Figure 5.3.

5.1.2 The Algorithm

Finally, all the steps described up to this point comprise the genetic algorithm that is used for the optimisation of the quantum noise parameters. A more concrete description of the routine is given in Algorithm 4.

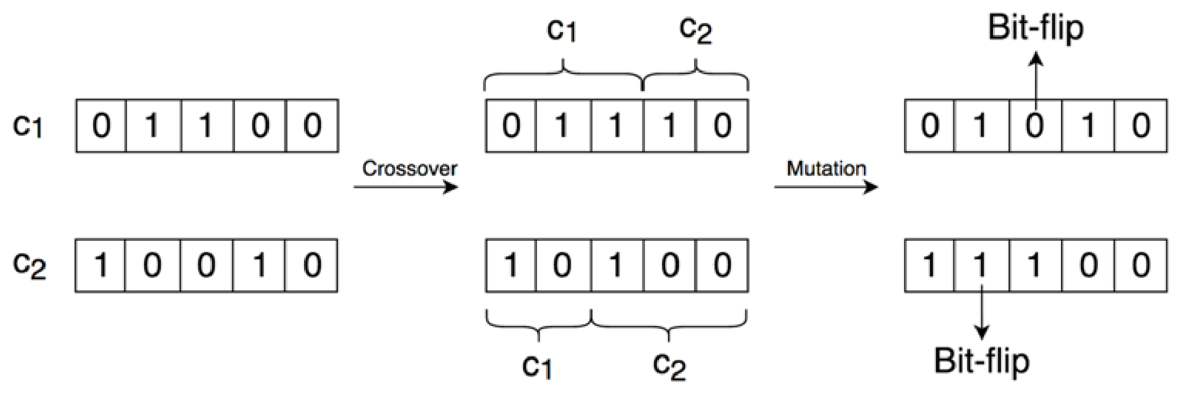


Fig. 5.3 The process of crossover and mutation. During crossover, part of each chromosome c_1 and c_2 is passed to the children with probability $p_{\rm co} = 0.6$. In the mutation, a random bit-flip occurs with probability $p_{\rm mt} = 0.05$.

Algorithm 4: Genetic algorithm for quantum noise parameter optimisation

```
1 Initialisation and parameter encoding. Input the pre-optimisation
    (hardware-calibrated) parameters and encode them to a binary string. Generate
    random chromosomes to match the population size. Define a number of
    generations, i.e., a number of iterations for the optimisation routine.
2 while the predefined number of generations is not met do
3     Define the current population. Create the current population, either from
        the initial or a new generation.
4     Evaluate fitness. Run simulations to compute the probability distribution of
        the evolution produced by each individual within the current population. The
        fitness value will be the Hellinger distance between the simulation output
        distribution and the quantum computer output distribution.
5     Selection. Choose the best individual (i.e., set of error rates) from the
        population to reproduce using tournament selection.
6     Recombination. Use crossover with probability $p_{\rm co} = 0.6$ to produce the
        chromosomes (individuals) that will result from the mating of the current
        population.
7     Mutation. Inflict mutations with probability $p_{\rm mt} = 0.05$ by bit-flip operations
        on the new chromosomes.
8 Result and evaluation. The final generation comprise the optimised population.
    This population is then evaluated and the best individual is picked. This
    individual is then decoded and its genes correspond to the new, optimised set of
    noise parameters.
```

5.2 Experiments and Results

This section describes in detail the experiments carried out using the optimisation routine laid out in the previous section, as well as the relevant results. In order to obtain an optimised set of noise parameters for the simulation of the quantum walk using the unified noise model, the GA optimisation routine of Algorithm 4 is used. The algorithm is set up, initialised and executed as described in Section 5.1. This method relies on iterative generations of new parameters, simulations using said new parameters and comparison of the simulated results with the quantum computer's distribution. In each iteration, the parameters that bring the simulated evolution closer to the quantum computer are kept.

In order for the execution time of the optimisation routine to remain relatively small and the results to be presentable, this section analyses in detail the results of two quantum walks with a small state space of $N = 4$ (two-qubit quantum walk) and $N = 8$ (three-qubit quantum walk) and a three-qubit and six-qubit system for their execution respectively. The quantum circuits use the generalised inverters approach of Section 3.2.1 for their implementation, same as the experiments of Chapter 4. For the two-qubit quantum walk, the quantum circuit requires three qubits as one is also necessary for the quantum coin (and $\log N = 2$ qubits for the state space). For the three-qubit quantum walk, the need arises for a two-qubit ancilla register, driving the size of the system to six qubits overall. The coin itself is never measured, meaning the results of the optimisation of the coin's error rates will not be directly visible through the measured states, but will nevertheless affect the overall computation. Results of larger quantum walk optimisations are given numerically in Table 5.1.

Before moving on to the analysis of the results, it is important here that, due to circuit optimisation methods implemented within Qiskit at the time of the experiments, the two-qubit quantum walk circuit utilised an additional qubit within the quantum computer in order to reduce the size of the circuit. Thus, the experiments and results showcase below follow this template where a four-qubit system is used for the two-qubit quantum walk.

For the optimisation routine, the genetic algorithm is allowed to run first for 50 generations (i.e., 50 repetitions). Considering the two-qubit quantum walk, the number of parameters that undergo optimisation is 9, calculated as follows: $r_s = 4$ single-qubit gate error rates, one for each of the four qubits in the system, $r_t = 3$ two-qubit gates according to the architecture of the computer, one for each pair of connected qubits, and $m = 2$ measurements at the end of the computation. The computation starts at state $|0^{\otimes(\log N)}\rangle$, and thus there is no need to account for state preparation of the qubits (i.e., $s = 0$). The population is chosen to be of size 8, i.e., there are eight individuals (chromosomes or encoded sets of noise parameters) within each generation and is kept static throughout the optimisation routine for the two-qubit quantum walk. The genetic algorithm results show that the Hellinger distance of the distributions between the post-optimisation simulation of the noise during the evolution of the quantum circuit and the noise within the real quantum computer has decreased from ~ 0.033 to ~ 0.005, an approximately 84.85%

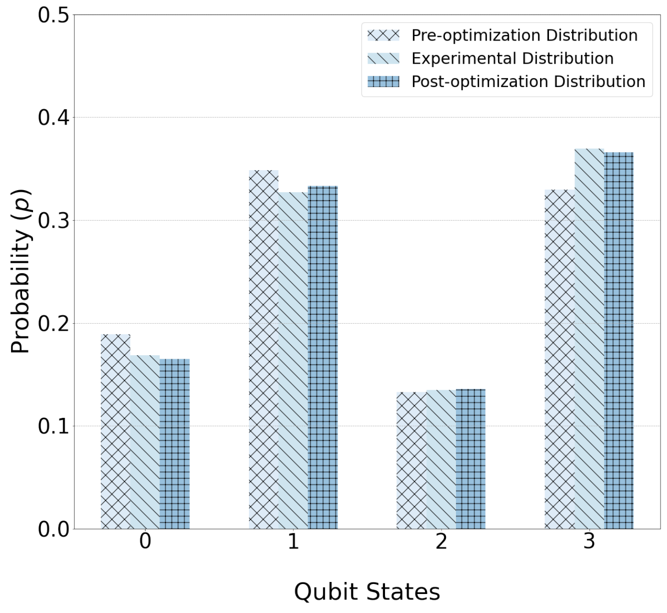


Fig. 5.4 Probability distribution of pre- and post-optimisation (crossed and vertically lined bars respectively) for a two-qubit quantum walk after a single optimisation routine compared with the quantum computer distribution (tiled bar). The GA is run for 50 generations and the optimised set produces a distribution that is 84.85% closer to the quantum computer result.

improvement. Figure 5.4 shows a visual implementation of the approximation improvement that results from the parameter optimisation after 50 generations of the GA.

The same experimental template is used for the three-qubit quantum walk experiments. For consistency, the optimisation routine is also run for 50 generations of the genetic algorithm but with a different population size of 12 individuals. The larger population size is used to account for the increased number of error rates deriving from the larger system used by the three-qubit quantum walk. In this case, the number of parameters required for optimisation is increased to 14, calculated as follows: $r_s = 6$ single-qubit rates, $r_t = 5$ two-qubit rates, $m = 3$ qubits measured – the state space of the quantum walk – and $s = 0$ as the walk is initialised on $|0\rangle$. The increased complexity of the optimisation due to the larger number of parameters is shown through the runtime of the optimisation routine on the classical computer, with the optimisation of the two-qubit quantum walk needing approximately $6,500$ seconds for its execution, whereas the three-qubit routine requires $9,500$ seconds. The results after a single optimisation routine (i.e., 50 generations) are shown in Figure 5.5. The HD between the post-optimisation simulation of the quantum circuit and the quantum computer has decreased from ~ 0.127 to ~ 0.054, an approximately 57% improvement.

In addition to the above, a number of experiments are run on quantum walks of various sizes, with the results following a similar trend. Table 5.1(a) presents the averaged results from three optimisation routines, i.e., three runs of the genetic algorithm routine for 50

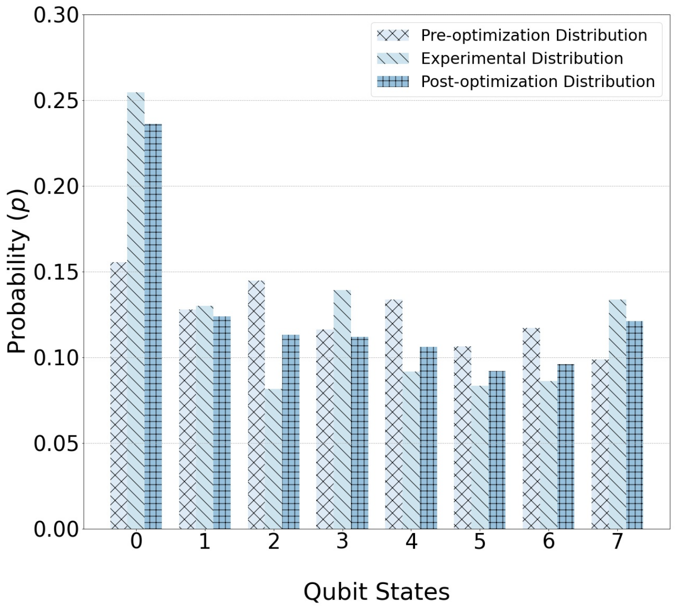


Fig. 5.5 Probability distribution of pre- and post-optimisation (crossed and vertically lined bars respectively) for a three-qubit quantum walk after a single optimisation routine compared with the quantum computer distribution (tiled bar). The GA is run for 50 generations and the optimised set produces a distribution that is 57.48% closer to the quantum computer's.

generations each. Due to the need of ancillary qubits in the computation, the number of parameters that need optimisation becomes large quickly. This means that the GA routine becomes slower with every qubit added in the state-space register.

It is noteworthy that the efficiency of the parameters post-optimisation declines with the size of the state space. In other words, the larger the number of parameters required for optimisation, the smaller the increase in efficiency of the approximated evolution. The main reason for that is the fact that the number of generations of the genetic algorithm remain static to 50 iterations, for comparison and readability purposes. Further experiments show that an increased number of generations of the GA during the experiments provide further improvement on the approximation of the quantum computer's distribution, even on quantum walks with larger state spaces. Table 5.1(b) outlines the optimisation results for 100 generations of the genetic algorithm.

5.2.1 Noise Parameter Analysis

One byproduct of the work carried out on parameter optimisation is the ability to conduct further analysis and draw various conclusions on the nature and accuracy of the hardware-calibrated noise parameters. The comparison between the hardware-calibrated parameters and the optimised parameters provides unique evidence on what the computer's error rates look like in the simulated world, thus allowing further conclusions on the infidelities of the quantum hardware. Here follows a comparison of the model parameters pre- and

State Space	Workspace	HD (Pre)	HD (Post)	% Distance	Runtime ($\times 10^3$ sec)
4	4	0.033	0.005 ± 0.001	84.85 $\downarrow$	6.5
8	6	0.127	0.054 ± 0.003	57.48 $\downarrow$	9.5
16	8	0.224	0.152 ± 0.006	32.14 $\downarrow$	12.1
32	10	0.393	0.301 ± 0.025	23.41 $\downarrow$	46.7
64	12	0.457	0.377 ± 0.016	17.51 $\downarrow$	93.5

(a) optimisation with 50 generations.

State Space	Workspace	HD (Pre)	HD (Post)	% Distance	Runtime ($\times 10^3$ sec)
4	4	0.033	0.003 ± 0.001	88.26 $\downarrow$	8.2
8	6	0.127	0.035 ± 0.004	72.44 $\downarrow$	11.3
16	8	0.224	0.121 ± 0.013	45.98 $\downarrow$	26.4
32	10	0.393	0.246 ± 0.014	37.40 $\downarrow$	87.9
64	12	0.457	0.336 ± 0.029	26.48 $\downarrow$	178.9

(b) optimisation with 100 generations.

Table 5.1 Results averaged from three optimisation routines of quantum noise parameters with our UNM model. Each routine is run for (a) 50 and (b) 100 generations of the genetic algorithm. HD (Pre) and HD (Post) are the HD between the probability distributions of the quantum computer and the simulator pre-optimisation and post-optimisation (along with standard deviation, $\pm$ s.d., rounded up to three decimal points) respectively; $\uparrow$ or $\downarrow$ mean increase or decrease in the distance between the distributions. The size of workspace is the number of qubits necessary for the computation (i.e. ancilla included). The runtime showcased is the average of the three optimisation routines.

post-optimisation for the simplest two-qubit quantum walk. The choice of the smallest system is purely for readability, as it has the smallest number of parameters optimised, but the exact same analysis can be carried out for a system of any size.

Table 5.2 shows the relevant noise parameters for the $N = 4$ state space quantum walk pre- and post-optimisation, after a single optimisation routine (50 generations of the genetic algorithm). Overall, it is evident from the parameters that the UNM operates closer to the computer for different error rates than the ones provided by the computer's calibrations. More specifically, comparing the parameters pre- and post-optimisation from Table 5.2, single-qubit operations and measurements on qubits 0 and 1 of the IBM quantum computer are noisier than the calibrations claim (i.e., their error rates are higher), with the opposite being true for qubits 2 and 3 and the two-qubit operations on all qubit pairs. Such results can be useful as they can potentially showcase an expectation of machine calibration outcomes or give an alternative read of the noisy reality within the quantum computer.

The same analysis applied to the larger systems shows that our model performs closer to the computer when single-qubit operations and measurements are, in their majority, noisier than calibrated, whereas two-qubit operations tend to be less noisy. There are different factors that cause this. First of all, the length of the experiment. For larger experiments, where the computation is much longer than the times T_1 and T_2, it is very

Opt.	Sq(0)	Sq(1)	Sq(2)	Sq(3)	Tq$(0,1)$	Tq$(1,2)$	Tq$(2,3)$	$M(0)$	$M(1)$
Pre-	0.000631	0.000550	0.000550	0.000496	0.015850	0.011410	0.021430	0.036700	0.080900
Post-	0.000685	0.000725	0.000406	0.000479	0.010435	0.010074	0.013219	0.038924	0.090734

Table 5.2 Pre- and post-optimisation noise parameters for an $N = 4$ state space system; Sq(q) are the single-qubit gate error rates, including Hadamard and NOT gates, Tq(q, q') are the two-qubit gate error rates and $M(q)$ are the measurement error rates for each qubit q or pair of qubits (q, q'), according to the architecture of the quantum computer.

difficult to get a concrete conclusion through such an analysis. As shown above, for smaller computations (and not necessarily quantum walks), the above methodology could provide a very good picture of whether the quantum computer calibrations overestimate or underestimate each of the error rates. Secondly, the above findings are the averaged results of three optimisation routines. This means that the claim of this analysis could still be an artefact of the randomness embedded within the parameter optimisation technique. Further optimisation runs for the same experiment could revoke this ambiguity. Unfortunately this endeavour could prove increasingly time consuming, especially for longer computations with a much larger number of noise parameters. Nevertheless, the current thesis remains satisfied with the improvement in precision when approximating the noise of NISQ computers using parameters optimised with the presented techniques.

5.2.2 Parameter optimisation Including the Decoherence and Dephasing Parameters

As a final remark to the study of noise parameter optimisation, the following analysis is presented in order to further support the decision to not include the relaxation and dephasing time parameters in the optimisation routine. It is clear from the above discussion that the number of parameters that need to be optimised increases with the size of the state space of the experiment, or in other words, with the number of qubits in the relevant register. This increase leads to much higher execution times for the GA routine, as showcased in Table 5.1. Including the decoherence and dephasing parameters in the optimisation would result to an even larger number of parameters, i.e., an additional $2 \times s_w$, where s_w is the size of workspace (ancilla and coin included) necessary to implement a quantum walk on a state space of size $N = 2^n$. Table 5.3 showcases the number of parameters for optimisation with respect to the number of qubits necessary to represent the state space of a discrete-time quantum walk.

As expected, the increasing number of parameters in the optimisation routine leads on an increase on the runtime of the genetic algorithm (see Table 5.1, Runtime column). The execution time of the optimisation quickly becomes relatively large for a quantum walk on $n = 6$ qubits (93.5×10^3s for 50 generations and 178.9×10^3s for 100 generations on a Macbook Pro 2017). Thus, it is logical to predict that, including the decoherence and dephasing parameters for optimisation will lead to an even larger execution time.

No. Qubits (n)	No. Param	No. Param with Decoherence
2	9	17
3	14	26
4	25	41
5	34	54
6	38	62

Table 5.3 Comparison of the increase in the number of parameters necessary for optimisation with respect to the number of qubits (No. Qubits) in the state space register. No. Param. shows the number of parameters when the decoherence noise is not included for optimisation.

State Space	Workspace	HD (Pre)	HD (Post)	% Distance	Runtime ($\times 10^{-3}$ sec)
4	4	0.033	0.0048	85.48 $\downarrow$	8.1
8	6	0.127	0.045	64.57 $\downarrow$	11.9

Table 5.4 Results showcasing the efficiency of the UNM when the decoherence and dephasing parameters are also optimised for 100 generations of the genetic algorithm. HD (Pre) and HD (Post) are the HD between the probability distributions of the quantum computer and the simulator pre-optimisation and post-optimisation (along with standard deviation, $\pm$ s.d., rounded up to three decimal points) respectively; $\uparrow$ or $\downarrow$ mean increase or decrease in the distance between the distributions. The size of workspace is the number of qubits necessary for the computation (i.e. ancilla included). The runtime showcased is the average of the three optimisation routines.

In order to examine the effects of including the decoherence and dephasing parameters $T_1(q)$ and $T_2(q)$ in the optimisation routine, a number of experiments were conducted using the methodology in question. As mentioned above, these parameters need to be considered per qubit in the system, including the ancilla and coin qubits. Considering the two optimisation routines analysed in Section 5.2, for a state space of $N = 4$ the number of parameters that need optimising are 17, calculated as follows: 9 parameters that represent the hardware infidelities and SPAM errors, as shown in Section 5.2, plus 8 relaxation and dephasing times $T_1(q)$ and $T_2(q)$, one for each of the four qubits in the workspace (i.e., 2×4). For $N = 8$ the number of parameters is 26: 14 for the hardware and SPAM errors plus 12 relaxation and dephasing times, one for each of the six qubits in the workspace (i.e., 2×6). Table 5.4 shows the results of a GA parameter optimisation routine with 50 generations.

A comparison of the results showcased in Tables 5.4 and 5.1(b) can lead to some very interesting conclusions. As is evident from the percentage of decrease in the distance between the distributions of the simulated quantum walk and the evolution of the quantum computer, the optimisation performs better in both cases when the decoherence parameters $T_1(q)$ and $T_2(q)$ are excluded from the optimisation. There are a couple of reasons for this, the most important of which is the fact that with more parameters to optimise, less space of the potential optimal parameters is searched. Secondly, the thermal relaxation and dephasing model requires $T_2(q) \leq 2T_1(q)$, which means that this condition needs to be enforced within the parameter optimisation, something that will, again, limit the space within which the GA can look for the optimal parameters.

Thus, the decision not to include the relaxation and dephasing parameters in the optimisation is justified by the above findings. Table 5.4 shows that the two-qubit quantum walk post-optimisation offers a similar increase in efficiency for half the generations (i.e., 50) as when also optimising the decoherence parameters for 100 generations (i.e., 84.85% vs 85.48% respectively). For the three-qubit quantum walk and the same number of generations the optimisation without the decoherence parameters approximates the quantum computer better than when the decoherence rates are optimised (i.e., 72.44% vs 64.57% respectively).

5.3 Discussion

This chapter enhances the quantum noise modelling approach implemented in Chapter 4. The unified noise model uses a number of hardware-calibrated noise parameters in order to simulate the noisy quantum evolution within the computer. In order to further improve the efficiency of the UNM, this chapter implements a classical optimisation routine relying on a genetic algorithm in order to produce a set of parameters that provide better approximation of the evolution within the quantum computer. A large number of experiments have shown that the noise parameters post-optimisation offer a better approximation of the quantum computer behaviour.

The novelty of the presented work is twofold: first, the idea and framework for such a technique, has not been carried out before and secondly, the unified noise model allows for such an optimisation (unlike other models (for example the IBMQ models compared with the UNM in Chapter 2). Further improvements in the accuracy of the optimised parameters could be obtained by adjusting the characteristics of the genetic algorithm, like implementing a larger number of iterations during the optimisation, or even experimenting with other optimisation techniques.

The one downside of the proposed methodology is the execution time of the GA routine when optimising the parameters of larger quantum systems. It is expected that a larger number of parameters would mean a lengthier optimisation process. Even so, the main source of the spikes observed in execution times is the fact that, during the fitness evaluation of the noise parameters, the GA needs to run a noisy simulation using the UNM. This is the only way to receive an indication of the performance of the noise parameters. For larger quantum walks, or indeed any scaling algorithm that could benefit from this technique, it is well-known that classical computers will struggle exponentially during simulations.

Finally, a very interesting extension to this work could be provided by employing a *quantum genetic algorithm* (QGA) in order to increase the efficiency of the optimisation routine. There have been a few studies regarding the efficiency of such a construct [146–148]. It is expected that, within the NISQ era, such optimisation routines would be very lengthy when run on a quantum computer. Thus, as it also becomes evident from the results presented in this thesis, the employment of a QGA to optimise the calibrated error rates lies far from the capabilities of near-term quantum computers due to the intense levels of noise. Nevertheless, when larger and more fault-tolerant machines or better error-correction techniques are available, the idea of QGA for noise parameter optimisation represents a real possibility.

5.4 Future Work

Further improvements in the accuracy of the optimised parameters could be obtained in the future by potential adjustments of the characteristics of the genetic algorithm. Such adjustments could include, for example, more iterations of the optimisation routine (i.e., a larger number of generations), allowing the genetic algorithm to explore and evaluate a larger space of potential error rates.

Programmatically, the GA methods could benefit from research around ways to lower the runtime of the optimisation, especially on large systems where the number of parameters increases dramatically. This thesis does not attempt to optimise the execution time of the routine, but it only attempts to present a novel methodology for such an achievement.

Chapter 6

Program Benchmarking Near-term Quantum Computers

This far, it is evident that quantum noise is a challenging obstacle. This thesis has worked extensively with noise modelling, engineering a model to better approximate a real quantum evolution and extensively discussing quantum walks, both in the context of circuit characteristics and as a tool for experimenting with the effects of noise. Based upon the produced knowledge, the last of the target topics set out in Chapter 1 is researched here: designing a framework for benchmarking quantum computers.

The outline of this chapter is as follows. Section 6.1 offers an introduction to the basic ideas and challenges of program benchmarks, as well as their novel use within quantum computing. Moving on, the necessary preliminary methods for the research are presented before the framework and benchmark metrics are introduced in Section 6.3. The proposed methodology is then used for a number of experiments, which showcase its utility by benchmarking three IBMQ computers. Finally, the chapter concludes with a discussion of the findings and an outline of potential future work.

The research presented in the chapter has been submitted for publication and the e-print can also be viewed on arXiv [149].

6.1 Program Benchmarking Quantum Machines

The process of benchmarking quantum computers aims to determine the performance of a quantum computing system under an appropriate, pre-determined set of metrics. In recent years, the increasing industrial and academic attention towards the field of quantum computing has generated great advances, both in terms of hardware (i.e., number of qubits, higher resilience and quantum volume, etc.) and software engineering. Quantum computers are constantly getting bigger and better. Within the Noisy Intermediate-Scale Quantum (NISQ) era [35], benchmarking the capabilities and performance of quantum computers when executing quantum programs is of even greater importance, especially for assessing key capabilities, like scalability or resilience to noise.

An intuitive approach to benchmarking quantum systems is establishing a set of quantum programs and measuring the performance of a quantum computer when executing each one. Such a procedure, also called *program benchmarking*, gains more merit in a scaling environment, thus satisfying a desirable characteristic of novel benchmarks as bigger quantum computers are built. There are various advantages to program benchmarking quantum computers, such as benchmarking the limits and behaviour of a quantum machine within a scaling, computationally intensive environment and particularly testing the system when performing a task (for example, a high-level quantum algorithm) that has potential applications in the "real-world".

Various companies appear to favor such an approach to examine the capabilities of their quantum systems. IonQ for example have tested their quantum computer using the Bernstein-Vazirani [150] and the Hidden Shift [151, 152] algorithms. The metric for performance was the likelihood of measuring the correct output [153]. On the other hand, Google has focused on the problem of quantum sampling for benchmarking their quantum systems. The relevant experiments have achieved results that, according to Google, demonstrate quantum supremacy [154]. In the latter case, while the chosen application was not particularly useful in a "real-world" scenario, its use in the implemented benchmarks excels at demonstrating the computing power of the system. Thus, a logical question arises, highlighting the difficulty of creating quantum program benchmarks:

Which benchmarks are more insightful for characterising the performance of quantum devices?

Within the current state of the field, there is a large number of competing quantum technologies, like superconducting [155], trapped-ions [156], silicon CMOS [157], photonics [158], and more. It is likely that a variety of quantum technologies will still exist post-NISQ era, either in a similar or alternative format, with each approach to quantum hardware being significantly different from the next. In benchmarking terms, the existence of different competing quantum technologies pose a major challenge. The technologies have different topologies and use different approaches for engineering the qubits, which in turn means that each qubit technology exhibits different characteristics and behaviour, and thus have unique strengths and weaknesses. As a simple example, the connectivity of an ion-trap computer provides a large advantage on some benchmarks over a superconducting quantum computer [159].

It is clear that quantum systems can behave differently on different benchmarks, which introduces another issue, that of *invested interests*. More specifically, invested interests characterises a situation when one uses benchmarks that are expected to perform well on a current system of interest [160]. While impressive considering the complexity of their technology, current quantum computers are very small compared to the computers expected to exist in the coming years, as the field expands and develops. Hence, current benchmarks are also very small and relatively simple compared to truly useful programs that future quantum machines are expected to execute. While running smaller versions of real-world applications introduces error, and is accepted in classical benchmarking, this is exacerbated for quantum computers. Entirely new issues may be introduced when scaling up and it is difficult to say whether performances measured today are good indicators of future performance. For example, consider the IonQ computer [153] which has all 11 qubits fully-connected. This configuration is possible at this scale, but this might not be true for a system with hundreds or thousands of qubits, i.e., due to limiting factors like massive cross-talk or excessive complexity of the hardware connectivity. Alternatively, such a system may require multiple fully-connected groups of qubits and communication among said groups will need to be realised, engineered and orchestrated differently [161]. This introduces additional complexity which is not found in today's small-scale quantum benchmarks.

Thus, considering the remarks made above, the question posed earlier can be refined as follows:

Which quantum processes would be useful for program benchmarking quantum computers in the future?

Famous algorithms such as quantum Markov chains [162], Shor's [163], Grover's [4] and quantum chemistry [159, 164, 165] are some obvious examples. Even if, for the most part, these algorithms will remain out of reach for near-term quantum computers (i.e., within the NISQ era) either due to excessive noise during the computation or limited number of qubits in the QPU, there is much to be gained from analysing their scalability and reaction to noise. Currently, classical-quantum hybrid algorithms [166–170] are popular due to their innate ability to make use of the limited resources of NISQ computers. Another example of an algorithm that holds great potential for benchmarking is quantum walks, due to their susceptibility to noise and clear quadratic advantage over classical random walks (see Chapter 3).

Inspired from the above conundrums, this chapter revolves around the use of high-level quantum algorithms to benchmark three of the newer IBM superconducting quantum computers, also introduced in Section 2.2.2: the IBMQx5 Bogota and Santiago and the IBMQx7 Casablanca machines, with architectures shown in Figures 2.3(a) and 2.3(b) respectively. The choice of quantum algorithms is crucial, as it is essential for them to be (i) scalable, in order to be able to face the challenge of the growing number of qubits in quantum computers, (ii) predictable in a way that allows one to clearly recognise its noisy behaviour, and (iii) demonstrate clear quantum advantage. With these criteria in mind, this thesis focuses on five quantum algorithms: (i) discrete-time quantum walks [171], (ii) continuous-time quantum walks [23], (iii) a circuit simulating the continuous-time quantum walk by decomposing its Hamiltonian to a sequence of Pauli gates, (iv) quantum phase estimation [172] and (v) Grover's algorithm for quantum search [4]. As mentioned above, using programs in order to benchmark computers is not an original idea. The contributions of this chapter are mainly identified around the novelty of the proposed framework, the experiments, results and the analysis that follows, as well as the algorithms used for benchmarking. Finally, it is noteworthy that using a continuous-time quantum

Parameter	Error Type	No. Parameters
p_r	Gate error rates	r
p_m	State preparation error rates	m
p_s	Measurement error rates	s
T_1	Thermal relaxation times	n
T_2	Dephasing times	n

Table 6.1 The noise parameters and number of noise parameters for each type of error within the UNM; n is the number of qubits in the system, m is the number of qubits that are measured, s is the number of state preparations that occur and r is the number of distinct types of gates implemented in the architecture, each considered once per qubit or pair of qubits.

algorithm and its Hamiltonian decomposition for benchmarking a (digital) quantum computer represents a novel attempt in the field.

6.2 Preliminary Methods

Before moving to identifying the benchmark framework and experiments, it is essential to present some preliminary methods that will play a central role in the research. To this end, this section offers a reference to the quantum algorithms and how they satisfy the criteria for program benchmarks, their respective circuits, the benchmark metrics, i.e., the indicators that encapsulate the performance of a quantum computer.

6.2.1 Unified Noise Model and Architecture Awareness

Aiming for a more comprehensive presentation of the present chapter, the noise model used to approximate the noisy behaviour of quantum computers is presented. This is the *unified noise model* (UNM), as it was engineered in Chapter 4. The UNM combines three sources of error: (i) hardware infidelities in the form of gate, state preparation and measurement errors, (ii) decoherence in the form of thermal relaxation and (iii) dephasing of the qubits. The experiments in Chapter 4 show that the UNM performs very well at approximating the behaviour of the IBMQ 15-qubit Melbourne computer and better than other state of the art noise models at the time.

The main characteristic of the UNM that makes it ideal for this work is its *architecture awareness*: the architectural graph that encompasses all the information regarding the connectivity of the qubits within the quantum processing unit (QPU) gets encoded within the noise model itself. Additionally, the UNM uses a number of noise parameters calibrated from the machine itself, as discussed in Section 4.2. These are parameters that express the error rates of the gates, state preparations and measurements as well as the time it takes for the qubits within the QPU to decohere and dephase (also presnted here in Table 6.1 for convenience). Each noise parameter is unique and corresponds to each qubit individually or pair of qubits.

6.2.2 The Quantum Algorithms

One of the major decisions related to the program benchmark framework is the programs/algorithms used to carry out the benchmarking process. As discussed above, it is essential for these algorithms to fit within the trend of the quantum computing field. Thus, there are three major criteria for rendering an algorithm "suitable" for a program benchmarking application, defined as follows:

Definition 6.1 (Criteria for algorithm selection)**.** *There are three characteristics identified as necessary for an algorithm to be used within the proposed program benchmarking framework.*

1. ***Scalability.*** *The algorithm should be able to scale up (or down) and run on increasingly larger quantum systems. This characteristic is essential for choosing an algorithm that can run on variable QPUs but also carry over the benchmarking framework on future quantum computers.*

2. ***Predictability.*** *The algorithm should produce a result that is easily predictable. An important addition to predictability is noise susceptibility: the algorithm should provide a result whose distortion under the effects of noise is easily distinguishable from the ideal evolution.*

3. ***Quantum advantage.*** *The algorithm should provide a computational speed-up over its classical counterpart or, in other words, represent a possibly relevant real-world application.*

As discussed earlier, in order to showcase the program benchmarking framework, this thesis identifies five algorithms of interest: (i) discrete-time quantum walks (DTQW), (ii) continuous-time quantum walks (CTQW), (iii) Pauli decomposition of the CTQW Hamiltonian (PD) (iv) quantum phase estimation (QPE) and (v) quantum search (QS). These algorithms, or the relevant theory, were introduced in Chapter 1, DTQW in Section 1.2.1, CTQW in Section 1.2.2, the Hamiltonian simulation theory in 1.1.2, quantum phase estimation in Section 1.2.4 and Grover's algorithm for quantum search in Section 1.2.3 respectively. The sections below explain in more detail the reasons the selected algorithms are suitable for program benchmarking.

Discrete-time Quantum Walks

Quantum walks (DTQW) are the quantum mechanical analogue of a classical random walk on a graph or a lattice [171, 61, 173]. They are considered for this research as they exhibit intrinsic properties that render their evolution easily predictable and highly susceptible to noise [61, 174], making them an ideal candidate for benchmarking a quantum device. A more comprehensive review of discrete-time quantum walks is given in Chapter 3 and Section 1.2.1.

First of all, a discrete-time quantum walk exhibits modular behaviour, a property defined in Definition 1.2. This characteristic describes the modular relationship between the parity of the number of coin-flips of the walk, the initial state and the current position of the walker. The interest in the context of benchmarking lies in the fact that this modular behaviour gets violated in a noisy environment, as clearly shown by the results of the experimental procedure in Section 3.2.3, satisfying Criterion 2 of predictability from Definition 6.1. Secondly, quantum walks propagate quadratically further than classical random walks [171, 8] (also proven in Section 3.1), thus showing clear quantum advantage over their classical counterparts and covering Criterion 3 in the above definition. Finally, quantum walks are a highly scalable process. The size of the state-space of a quantum walk (i.e., the number of states that the walk traverses, represented by the number of qubits in the relevant register) can easily increase to match the size of the quantum computer that is of benchmarking interest. This satisfies Criterion 1.

Continuous-time Quantum Walks

Continuous-time quantum walks (CTQW) were first introduced by Fahri and Gutman in [23]. This algorithm, much like the DTQW, has an easily predictable quantum evolution that is highly susceptible to quantum noise, satisfying Criterion 2, but exhibit very different characteristics to the discrete case, thus justifying its selection in addition to DTQW. First of all, the CTQW evolution is determined by a Hamiltonian, H, instead of a coin-flip and is driven by a unitary of the form $e^{-\mathrm{i}Ht}$. Unlike the DTQW, continuous-time quantum walks do not exhibit modular behaviour. Although, like DTQWs they feature a quadratic increase in the walker's propagation [27, 28], satisfying Criterion 3 for quantum advantage.

Finally, CTQWs are an easily scalable process as adding qubits to circuit can scale up the size of the state-space (Criterion 1). As far as the current thesis is aware, this is the first work that uses a continuous-time quantum algorithm to benchmark the performance of a digital quantum computer.

Pauli Decomposition of CTQW Hamiltonian

This process is not an algorithm itself, but it is essentially an alternative approach to the continuous-time quantum walk algorithm presented in the previous paragraph. In the field of quantum computing, the decomposition of quantum Hamiltonians to a set of universal gates is a well-studied area of research [175–179]. Within this thesis, the interest lies in decomposing the Hamiltonian of the CTQW using the well-known set of Pauli matrices as the universal gate set. This leads to an alternative way to implement the CTQW and use it for the benchmarking process. A more in-depth analysis of the essential elements of this process is presented in the following Section 6.2.3.

This procedure, often called *Hamiltonian simulation* as introduced in Section 1.1.2, adheres to the criteria for a good benchmarking program indirectly via the algorithm it decomposes. In other words, since the CTQW is suitable for benchmarking, so is the circuit that implements the decomposition of the CTQW Hamiltonian. Furthermore, it provides added value to this research since one can evaluate the performance of the quantum computer when executing the Hamiltonian simulation of a quantum process (i.e., CTQW), as well as provide the tools for a meaningful comparison of the decomposition with the original algorithm.

Quantum Phase Estimation

The quantum phase estimation (QPE) algorithm is a well-known process used to estimate the phase (or eigenvalue) of an eigenvector of a unitary operator. More precisely, given an arbitrary quantum operator U and a quantum state $|\psi\rangle$ such that $U|\psi\rangle = e^{2\mathrm{i}\pi\theta}|\psi\rangle$, the algorithm estimates the value of θ, given an approximation error [180, 172, 181]. A more comprehensive review of this topic is given in Section 1.2.4.

This thesis aims to exploit three characteristics of the QPE that make it interesting for program benchmarking. First of all, it is a scalable algorithm as increasing the number of qubits in the system results in a better accuracy of the estimated phase (Criterion 1). Furthermore, the result of the QPE is easily predictable and highly susceptible to quantum noise (Criterion 2). Finally, QPE offers clear quantum advantage, achieving an exponential speed-up over known classical methods, rendering the algorithm one of the most important subroutines in quantum computing and serving as the building block of major quantum algorithms, like Shor's [163] or the HHL algorithm [182]. The latter characteristic strongly satisfies Criterion 3.

Quantum Search

The last program utilised in the program benchmarking is Grover's algorithm [4], which describes a process of searching for a specific item within a database. Within this thesis, quantum search (QS) is used to look for a specific number s within a set of numbers $\mathcal{S} = \{0, \ldots, 2^{n-1}\}$, where n is the number of qubits within the quantum system that participate in the computation.

Quantum search represents an ideal algorithm for benchmarking quantum computers. The algorithm can scale up to search for an item within a larger database simply by adding qubits to the relevant quantum register, satisfying Criterion 1. The result is easily predictable, as it is simply the item (or number, in this case) sought, as well as susceptible to noise (Criterion 2). Additionally, QS can speed up an unstructured search problem quadratically, thus making it a very appealing application for quantum computers. Finally, Grover's algorithm can serve as a general trick or subroutine to obtain quadratic runtime improvements for a variety of other algorithms through what is called amplitude amplification [183] (Criterion 3).

6.2.3 Pauli Decomposition of Continuous-time Quantum Walk Hamiltonians

One of the most challenging subjects in the field of quantum computing is the so-called Hamiltonian simulation problem (HSP). The HSP, as introduced in Definition 1.1, encapsulates the attempts to implement a quantum Hamiltonian as quantum circuit. That circuit will use a sequence of (most likely) universal gates, like for example, a set of operations that can be executed on a target quantum computer, or that can be easily decomposed to a set of gates implemented within the said quantum computer. One such set can be formed using the well-known Pauli operators. A more extended review of quantum Hamiltonians and the HSP is provided in Section 1.1.2.

In general, consider an arbitrary quantum system with Hamiltonian, H, of size $N \times N$, where $N = 2^n$ and n is the number of qubits in the system, and the set of Pauli operators $S = \{\sigma_I, \sigma_x, \sigma_y, \sigma_z\}$ with well-known matrix representation. The Hamiltonian, H, can be decomposed into a sequence of Pauli operators of the set S as follows

$$H = \sum_{i_1,\dots,i_N \in \{I,x,y,z\}} \alpha_{i_1,\dots,i_N} \left(\sigma_{i_1} \otimes \cdots \otimes \sigma_{i_N}\right), \tag{6.1}$$

where the scalar α can be calculated as

$$\alpha_{i_1,\dots,i_N} = \frac{1}{N} \operatorname{tr}\left[\left(\sigma_{i_1} \otimes \cdots \otimes \sigma_{i_N}\right) \cdot H\right].$$

This is called the *Pauli decomposition* (PD) of the Hamiltonian driving the evolution of the quantum system. Intuitively, equation (6.1) results to taking the sum of all tensor products that arise from computing every possible combination of the Pauli operators (i.e., the operators of set S).

Within this thesis, the Hamiltonian decomposition is utilised to engineer a circuit implementing a CTQW on an N-cycle. The CTQW Hamiltonian can be defined as $H_{\text{qw}} = \gamma A = \frac{1}{d}A = \frac{1}{2}A$, where $\gamma = 1/d$ is the hopping rate between the two adjacent nodes in the cycle with node degree $d = 2$ and A is the adjacency (circulant) matrix, defined for a continuous-time quantum walk of arbitrary size as in equation (1.11).

The next step is to construct the unitary evolution operator that corresponds to the Hamiltonian of the quantum system. This can be done by exponentiating the Hamiltonian as $e^{-\mathrm{i}Ht}$, where H is a sum of terms that follows the Pauli decomposition of the Hamiltonian, i.e., a sum of the form of equation (6.1). It is important here to consider two things. First of all, during the matrix exponentiation, a decomposition of the form $e^{-\mathrm{i}(H_1+H_2)t} = e^{-\mathrm{i}H_1 t}e^{-\mathrm{i}H_2 t}$, where H_1 and H_2 are Hermitian operators, is possible iff H_1 and H_2 commute, i.e., $H_1H_2 - H_2H_1 = 0$. This rule is naturally expanded for more than two matrices on the exponent.

Secondly, in the case that not all matrices in the exponent commute, the unitary operator resulting from the Hamiltonian exponentiation needs to be decomposed using the Lie-product formula [175] as, introduced in equation (1.4), which is also written here for convenience:

$$e^{-\mathrm{i}(H_1+H_2+\dots)t} \approx \left(e^{-\mathrm{i}H_1 t/r} e^{-\mathrm{i}H_2 t/r} \dots\right)^r \tag{6.2}$$

where, for this case, $H = \sum_j H_j$ is the Pauli decomposition of the Hamiltonian to a sequence of Hermitian terms. Equation (6.2) suggests that one can approximate the left-hand-side exponent of the Pauli decomposition of the Hamiltonian, $e^{-\mathrm{i}(\sum_j H_j)t}$, with r repetitions of the right-hand-side product formula $\left(e^{-\mathrm{i}H_1 t/r} e^{-\mathrm{i}H_2 t/r} \dots\right)$. This approximation exhibits a bounded error depending on r [175, 177]. To ensure that the Pauli Hamiltonian decomposition exhibits error at most ϵ, the bound r can be taken as [175]

$$r = (||H||_2 t)^2/\epsilon,$$

where $||H||_2$ is the norm of the Hamiltonian H and t is the continuous-time duration of the quantum evolution.

6.2.4 The Quantum Computers and Circuits

Within this chapter the interest lies in benchmarking three of the IBMQ computers, the IBMQx5 Bogota, IBMQx5 Santiago and the IBMQx7 Casablanca machines. A more detailed review of the above quantum computers is provided in Section 2.2.2. All three of them exhibit a quantum volume $V_Q = 32$ [64, 41].

Quantum Computer Architectures

As introduced in Definition 2.5, the architecture of a quantum computer is conveniently described by its architectural graph, i.e., a schematic that shows the locality and connectivity of all the qubits within the QPU. It is essential for the benchmarking procedure carried out within this research to be architecturally aware, or in other words, to have knowledge and take into account the connectivity of the qubits within the QPU. This is also reflected on choosing the UNM, an architecture-aware noise model [85]. Figure 2.3 shows the qubit connectivity of the quantum computers benchmarked in this chapter.

Quantum Circuits and Characteristics

For the implementation of the discrete-time quantum walk, this chapter once again makes use of the gate efficient approach that is based on generalised-inverter gates, as shown in Chapter 3. The QPE circuit is based on the work done in [184] and heavily relies on quantum Fourier transform (and its inverse) [185] to estimate the relevant eigenvalue. For the QS circuit, this thesis makes use of two approaches to the implementation, one with ancilla qubits (QSa) and one without (QSn) [186].

For the continuous-time quantum walk, the Hamiltonian is automatically implemented by the Qiskit API when submitted for execution on the quantum computer, thus eliminating the need for a circuit design. This circuit will, once again, be an automatic decomposition of the Hamiltonian to the set of gates executable on IBMQ backends, but it will not be done using the Pauli decomposition. On the other hand, the PD of the Hamiltonian can be easily implemented on the quantum computers as it already maps the continuous-time Hamiltonian to a discrete basis gate set decomposition.

It is instructive here to identify four quantum circuit characteristics that are of interest for benchmarking:

- the number of quantum gates that participate in each circuit;

- the number of active qubits, or otherwise, the subset of qubits (could, of course, be all of them) of the quantum computer that are utilised by the quantum circuit (also called workspace);
- the depth of the circuit, i.e., the longest path between the start of the circuit and a measurement gate; for consistency, all the measurements are enforced to be at the end of the circuit, past the last quantum gate applied; and
- the runtime of the circuit on the quantum computer.

Table 6.2 shows those characteristics of the quantum circuits that implement each quantum algorithm.

6.2.5 Benchmark Indicators: Hellinger Distance

Finally, it is important to find an appropriate metric that will act as the indicator of the performance of a quantum computer. For this work, the end results of the benchmarking process will be in the form of a comparison of the quantum computer output distribution with the distributions resulting from the unified noise model simulations of each machine and the ideal evolution. Such a comparison allows a full picture of the behaviour of a quantum computer as opposed to the expected one, giving clear indications of the levels of noise through the difference between the two results.

It is therefore natural to choose a metric that reflects this difference, and one that has been used extensively within this thesis: the Hellinger distance. As defined in Definition 4.1, the Hellinger distance offers a concise, efficient and comprehensible way to quantify the distance between two distributions without the need for them to have the same support, while at the same time offering a good way to compare the results.

6.3 Framework for Program Benchmarks of Quantum Machines

This section introduces the novel methodology used to benchmark quantum computers using high-level quantum algorithms.

6.3.1 The Benchmark Metrics

Following the selection of the Hellinger distance as the metric for comparison between the experimental results, it is necessary to construct a framework within which this metric will give clear and instructive information. First of all, there is the need for a more sensible representation of the benchmark indicators that are based on the HD. To achieve a more memorable notation, in the following analysis the symbol q denotes the quantum computer, i the ideal evolution and n the noisy simulation. For example, the subscript $q|n$ denotes a value that corresponds to the difference between the quantum computer evolution (q)

Machine	No. Gates	Size of Workspace	Depth	QC Runtime ± s.d. (ms)
Bogota	47	3	35	2.41 ± 0.03
Santiago	47	3	35	2.34 ± 0.02
Casablanca	47	3	35	2.54 ± 0.05

(a) Discrete-time quantum walk circuit characteristics.

Machine	No. Gates	Size of Workspace	Depth	QC Runtime ± s.d. (ms)
Bogota	19	2	13	2.34 ± 0.04
Santiago	19	2	13	3.01 ± 0.07
Casablanca	19	2	13	2.62 ± 0.01

(b) Continuous-time quantum walk circuit characteristics.

Machine	No. Gates	Size of Workspace	Depth	QC Runtime ± s.d. (ms)
Bogota	243	2	183	2.42 ± 0.04
Santiago	243	2	183	2.28 ± 0.06
Casablanca	243	2	183	3.76 ± 0.04

(c) Pauli decomposition of CTQW circuit characteristics.

Machine	No. Gates	Size of Workspace	Depth	QC Runtime ± s.d. (ms)
Bogota	93	4	66	2.54 ± 0.01
Santiago	97	4	72	2.26 ± 0.03
Casablanca	100	4	75	2.68 ± 0.06

(d) Quantum phase estimation circuit characteristics.

Machine	No. Gates	Size of Workspace	Depth	QC Runtime ± s.d. (ms)
Bogota	497	4	358	2.70 ± 0.08
Santiago	479	4	343	2.56 ± 0.04
Casablanca (a)	788	6	503	2.81 ± 0.04
Casablanca (na)	465	4	336	2.74 ± 0.03

(e) Quantum search circuit characteristics.

Table 6.2 Quantum circuit characteristics for the five quantum circuits. No. gates and size of workspace are the number of gates and active qubits in the circuit respectively; depth of the circuit is the longest path between the start of the circuit and a measurement gate; QC runtime is the approximate average execution time of the circuit on the quantum computer along with standard deviation (s.d.). For the QS circuit, Casablanca (a) is the circuit with ancilla, and (na) the circuit without ancilla qubits.

and the noisy evolution (n). Three distances of interest are defined, or otherwise, three benchmark metrics, as follows.

Definition 6.2 (alpha benchmark)**.** *The Hellinger distance between the probability distribution of the quantum computer evolution, Q, and the ideal distribution, D, namely $h_{id}(Q, D)$, is the alpha benchmark, with notation $\alpha_{q|i}$:*

$$\alpha_{q|i} \equiv h(Q, D). \tag{6.3}$$

Definition 6.3 (beta benchmark)**.** *The Hellinger distance between the probability distribution of the quantum computer evolution, Q, and the distribution resulting from the noisy simulations, N, namely $h_{nm}(Q, N)$, is the beta benchmark, with notation $\beta_{q|n}$:*

$$\beta_{q|n} \equiv h(Q, N). \tag{6.4}$$

Definition 6.4 (gamma benchmark)**.** *The Hellinger distance between the distribution resulting from the noisy simulations, N, and the ideal distribution, D, namely $h_{sm}(N, D)$, is the gamma benchmark, with notation $\gamma_{n|i}$:*

$$\gamma_{n|i} \equiv h(N, D). \tag{6.5}$$

As mentioned in Section 6.2.5, the Hellinger distance satisfies the triangle inequality. Hence, since the benchmarks established in the above definitions describe the pairwise Hellinger distances between three probability distributions (the quantum computer, Q, the simulated evolution, N, and the ideal evolution, D), it is possible to derive a relationship between $\alpha_{q|i}$, $\beta_{q|n}$ and $\gamma_{n|i}$ through the Lemma below (whose proof follows simply from the triangle inequality for a metric).

Lemma 6.1 (benchmarks triangle inequality)**.** *Given probability distributions Q, N and D, as established in the benchmark definitions, the pairwise Hellinger distances between those distributions, i.e., $h(Q, D)$, $h(Q, N)$ and $h(N, D)$ follow the triangle inequality:*

$$h(Q, D) \leq h(Q, N) + h(N, D).$$

Thus, the relevant benchmarks will also follow the triangle inequality as:

$$\alpha_{q|i} \leq \beta_{q|n} + \gamma_{n|i} \tag{6.6}$$

The above Lemma effectively means that, according to the benchmark definitions, the deviation of the quantum computer evolution from the ideal ($\alpha_{q|i}$) will never be greater than the sum of the expected (i.e., simulated) evolution derived by the levels of noise within the machine and the distance between the simulated and ideal distributions ($\beta_{q|n} + \gamma_{n|i}$). In other words, defining the benchmark metrics using the Hellinger distance makes it possible to quantify the confidence on the estimated level of noise during the execution of the quantum circuit.

6.3.2 The Framework

Each of the definitions in Section 6.3.1 play an essential role in understanding the benchmarked behaviour of a quantum computer, as will be explained later in this section. For a better understanding, it is instructive to first describe the process which yields those benchmarks. Thus, the following sequence of six steps defines the *framework for program* *benchmarking* quantum computers. Figure 6.1 offers a simple visual representation of the framework.

Definition 6.5 (Framework for program benchmarking)**.** *Program benchmark the efficiency of an arbitrary quantum computer as follows.*

Step 1 ***Select benchmark method(s).*** *The selection of quantum algorithms that will be used for benchmarking should adhere to the three criteria described in Section 6.2.2: (i) scalability, (ii) predictability and noise susceptibility and (iii) quantum advantage.*

Step 2 ***Quantum noise model and simulator.*** *Select or implement a noise model that approximates the noisy evolution within the quantum machine and a simulator that can execute the noise model.*

Step 3 ***Run experiments.*** *Design and execute a suitable number of experiments of the benchmark method(s) (selected on* ***Step 1****) on the quantum computer. This step also includes calibrating the noise parameters that encapsulate the level of noise within the quantum computer at the time of the experiments.*

Step 4 ***Simulate the noisy evolution.*** *Use the noise model in order to simulate the noisy evolution of the quantum computer. Use the calibrated noise parameters to indicate the levels of noise on the time of the experiment, as described in* ***Step 3****. Due to constant fluctuations of the noise parameters, this step is necessary in order to provide a more accurate representation of these parameters, and in extent, the intensity of noise at the time of the execution of the experiment/algorithm.*

Step 5 ***Simulate the ideal evolution.*** *This can be done either through simple noise-free simulations or by calculating the probabilities through the quantum statevector.*

Step 6 ***Calculate the benchmarks.*** *The final benchmark metrics $\alpha_{q|i}$, $\beta_{q|n}$ and $\gamma_{n|i}$ are the Hellinger distances between the quantum computer, the UNM and the ideal evolution in the setting described in Definitions 6.2, 6.3 and 6.4.*

Following the benchmarking framework one can extract meaningful results from a series of comparisons between the benchmark metrics, $\alpha_{q|i}$, $\beta_{q|n}$ and $\gamma_{n|i}$. The $\beta_{q|n}$ benchmark essentially showcases how closely the noise model simulates the behaviour of the quantum computer. The $\alpha_{q|i}$ benchmark shows how far the behaviour of the quantum computer falls from the noise-free case thus giving an estimate of the overall computer performance under the effects of noise. The comparison between the $\beta_{q|n}$ and $\alpha_{q|i}$ benchmarks can highlight valuable information: if $\beta_{q|n} < \alpha_{q|i}$ the noise levels in the quantum computer are closer to the estimated ones from the noise simulations; on the opposite case the computer operates closer to the ideal evolution. In the latter case the quantum computer behaves more efficiently with lower level of noise than expected, thus giving us more confidence regarding the computational result.

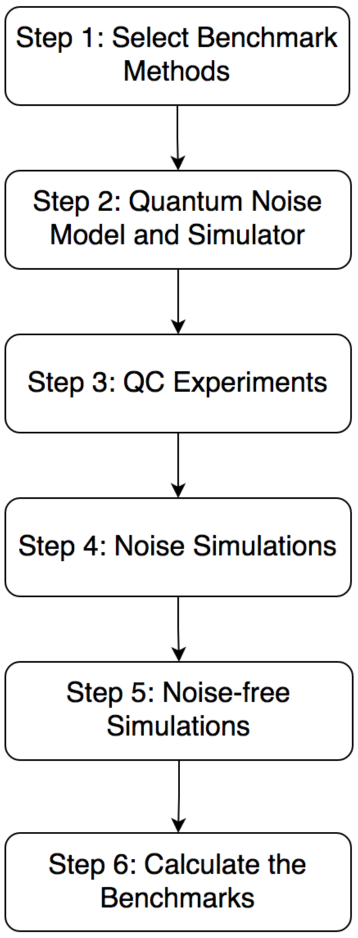

Step 1: Select Benchmark Methods
Step 2: Quantum Noise Model and Simulator
Step 3: QC Experiments
Step 4: Noise Simulations
Step 5: Noise-free Simulations
Step 6: Calculate the Benchmarks

Fig. 6.1 Framework for program benchmarking quantum machines in six steps.

The $\gamma_{n|i}$ benchmark, even though it can be used as an estimate of the noise levels in the quantum computer, does not give any relevant information on its own. The value comes when considering the $\gamma_{n|i}$ benchmark together with the $\alpha_{q|i}$ benchmark. First of all, the closer the values of $\alpha_{q|i}$ and $\gamma_{n|i}$ are, the smaller the value of $\beta_{q|n}$ (if $\beta_{q|n} = 0$ then $\alpha_{q|i} = \gamma_{n|i}$ and vice versa). Additionally, if $\alpha_{q|i} > \gamma_{n|i}$, then the noise model, and hence the noise parameters, underestimate the levels of noise during the quantum computer evolution, with the opposite being true if $\alpha_{q|i} < \gamma_{n|i}$. Moreover, the absolute difference $|\alpha_{q|i} - \gamma_{n|i}|$ can quantify this noise over- or underestimation. Precisely, the smaller the absolute difference the smaller the error in estimation. This information is useful to the benchmarking process as it further highlights whether the machine is more or less noisy than estimated while also showing the efficiency of the machine calibration techniques.

Finally, further interesting remarks can be made using the triangle inequality from Lemma 6.1. From equation (6.6) follows that $\beta_{q|n} \geq \alpha_{q|i} - \gamma_{n|i}$. Furthermore, as can be realised from the above analysis, a comparison between $\alpha_{q|i}$ and $\gamma_{n|i}$ can show whether the calibrated noise parameters over- or underestimate the level of noise during the evolution on the quantum computer, and the absolute difference $|\alpha_{q|i} - \gamma_{n|i}|$ can give an indication of the scale of the error in estimation. Importantly, the above inequality does not hold for absolute values when subtraction takes place (i.e., for an inequality of the form $|\beta_{q|n}| \geq |\alpha_{q|i} - \gamma_{n|i}|$ when $\gamma_{n|i} > \alpha_{q|i}$). Considering this, one can interpret the triangle inequality as a measure of the confidence on the calibrated parameters encapsulating a picture of the noise that is accurate enough to provide a good estimation of the quantum evolution, expressed as follows:

- If $\beta_{q|n} \geq |\alpha_{q|i} - \gamma_{n|i}|$, then the over- or underestimation of the noise is small enough to provide estimates of the quantum evolution with *high confidence.*
- If $\beta_{q|n} < |\alpha_{q|i} - \gamma_{n|i}|$, then the calibrated parameters generate *low confidence* on the levels of noise.

6.4 Experiments and Results

Following the presentation of a comprehensive framework for generating the benchmarks for a quantum computer, this section showcases the process of benchmarking the three IBMQ machines mentioned above (Bogota, Santiago, Casablanca).

6.4.1 Experimental Setup

Regarding the experimental setup for the benchmarking experiments, the quantum circuits that implement the chosen quantum algorithms are executed. In the case of the DTQW, just one step of the algorithm (i.e., one coin-flip) is run on a workspace (i.e., number of active qubits used by the circuit) of three qubits, or in other words, a state space of $N = 4$ (needing $n = 2$ qubits for its representation) and one qubit for the coin. The walk is initialised on state $|0\rangle$, as the previous work done in Chapter 3 shows that this configuration is satisfactory for errors to take place and the behaviour of the quantum walk to evolve in a predictable manner.

For the CTQW and its PD, the experiments are carried out on a small two-qubit state-space, i.e., four different states, similar to the discrete case. The main reason for this choice is that the Pauli decomposition gets excessively large for a quantum walk on a three-qubit space or larger, something that hinders the runtime of the experiments, while the results are not in any way more instructive. The algorithms are implemented for a continuous (arbitrary unit) QPU time of $t = 3$.

Next, the QPE routine is tailored to estimate a phase of $\theta = 2\pi/3$ using a workspace of four qubits. The theoretical probability of success can be estimated at 0.688. Finally, the QS implementation performs an unstructured search for the decimal element $|s\rangle = |10\rangle$ within a four qubits state-space, with easily deductible binary representation: $|10\rangle_{\text{dec}} = |1010\rangle_{\text{bin}}$. The chosen number s will be searched within a four-bit dataset containing numbers $|0\rangle$ to $|15\rangle$. The QS algorithm is run for three iterations and shows a theoretical probability of success estimated at 0.96. Table 6.3 shows more comprehensively the initial configuration for each benchmarking algorithm.

Each algorithm is run independently 100,000 times on the three chosen quantum computers. Noisy simulations are then executed using the UNM and the noise-free simulations via the ideal simulator on [63], thus reproducing the noisy and the ideal

Algorithm	No. Qubits	So. Workspace	Duration	Probability of Success
DTQW	2	3	1 (coin-flip)	$p_{\text{succ}} = 0.5$ in states $\|1\rangle$ and $\|3\rangle$
CTQW	2	2	$t = 3$	$p_{\|2\rangle} = 0.99$, $p_{\|1\rangle} = p_{\|3\rangle} = 0.005$
PD	2	2	$t = 3$	$p_{\|2\rangle} = 0.99$, $p_{\|1\rangle} = p_{\|3\rangle} = 0.005$
QPE	3	4	1 (iteration)	$p_{\text{succ}} = 0.688$ in state $\|3\rangle$
QSa	4	6	3 (iterations)	$p_{\text{succ}} = 0.96$ in state $\|10\rangle$
QSn	4	4	3 (iterations)	$p_{\text{succ}} = 0.96$ in state $\|10\rangle$

Table 6.3 The initial configuration for the quantum walk (DTQW), continuous-time quantum walk (CTQW) and its Pauli decomposition (PD), quantum phase estimation (QPE) and quantum search algorithms with ancilla (QSa) and without ancilla (QSn) for the benchmarking experiments. No. qubits is the number of qubits in the state space, So. workspace is the number of active qubits utilised by the circuit, iterations is the number of repetitions of the quantum circuit and the probability of success is the theoretical probability of the algorithm to give us the correct (or expected) result.

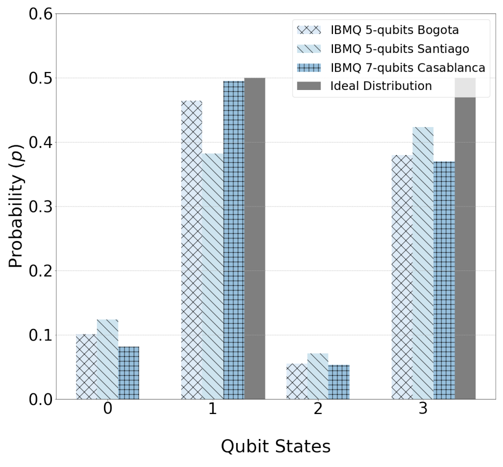
quantum evolutions respectively. After the experiments and simulations are concluded, the benchmarks $\alpha_{q|i}$, $\beta_{q|n}$ and $\gamma_{n|i}$ are computed, as defined in Section 6.3.

A visual representation of the comparison between the distributions resulting from the quantum computer executions and the ideal simulations for each quantum algorithm are shown in Figure 6.2. Each individual graph in the figure portrays the probability distributions for the execution of one of the quantum algorithms used for benchmarking. It also includes the probability distribution of the ideal evolution of the respective quantum algorithm. A further comparison between the quantum computer and the individual UNM distributions for each machine is given in Appendix E.

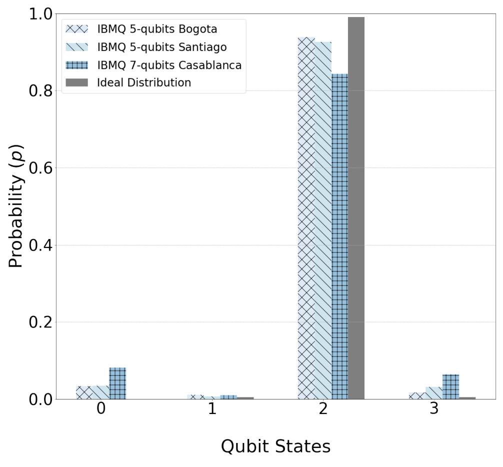
6.4.2 Results

After the algorithms have been setup as described above and the experiments and simulators have been executed on the quantum computer and relevant simulators, the results of the experimental procedure are shown on Table 6.4, categorised for each machine of interest. A further visualisation of the results is given in Figure 6.3.

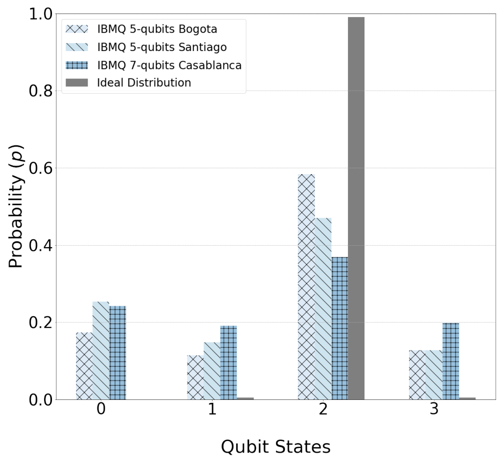
It is important at this point to emphasise an advantage of the structure presented for program benchmarking quantum computers. The use of the Hellinger distance to define the three benchmarks ($\alpha_{q|i}$, $\beta_{q|n}$ and $\gamma_{n|i}$) allows the comparison of the performance of different quantum algorithms on the same basis, i.e., a dimensionless quantity. In other words, through the proposed framework, apart from measuring the efficiency of a quantum processor, additional information can be gathered while also comparing the efficiency of different circuit implementations for the selected algorithms. This is done here for the two approaches on implementing the CTQW on a gate-based computer, one that is done through the IBMQ API and one that uses the Pauli decomposition of the CTQW Hamiltonian (see Section 6.2.3).



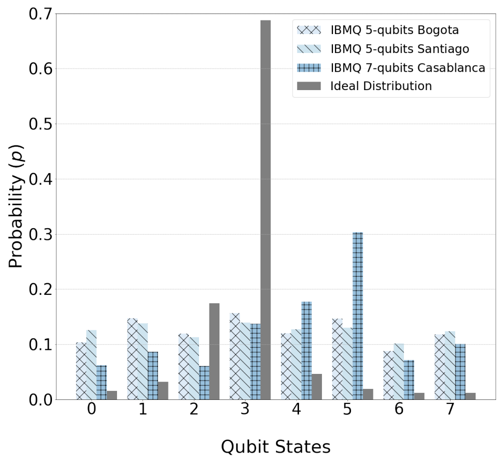
(a) DTQW distributions.



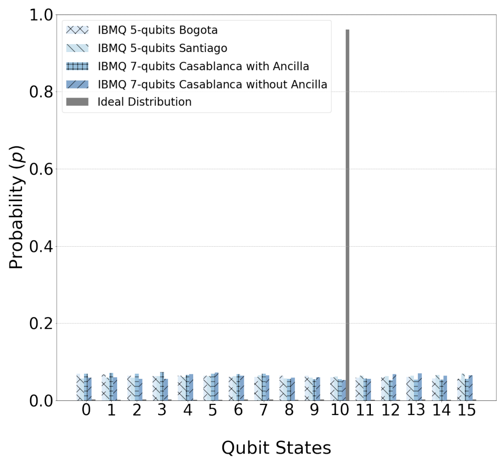
(b) CTQW distributions.



(c) PD distributions.



(d) QPE distributions.



(e) QS distributions.

Fig. 6.2 Comparison of the probability distributions for each of the algorithms when executed on the quantum machines and when simulated in a noise-free environment.

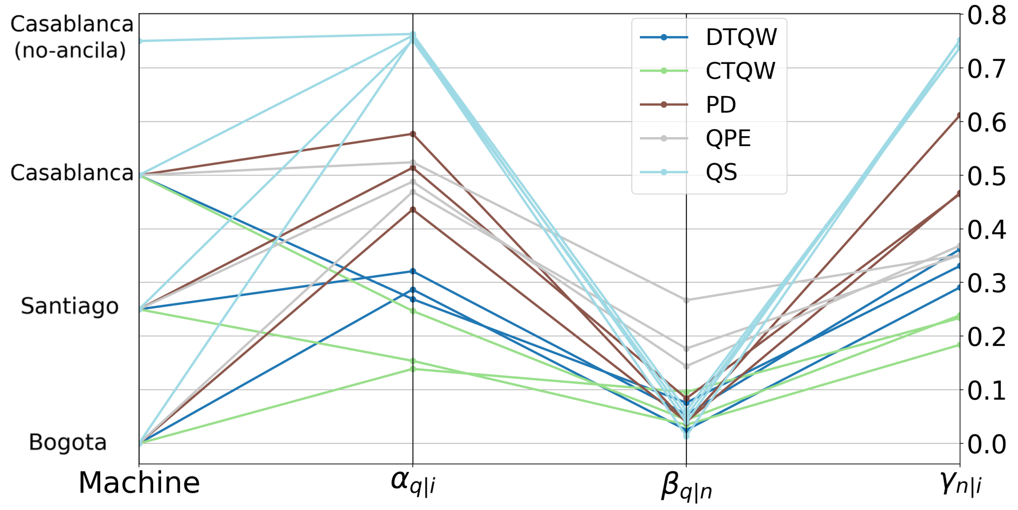


Fig. 6.3 Visualisation of the three benchmarks, $\alpha_{q|i}$, $\beta_{q|n}$ and $\gamma_{n|i}$ for each of the three quantum computers (mapped to the left-hand side y-axis) when executing the five quantum algorithms. Note: the subscripts q represents the quantum computer distribution, n the UNM simulation and i the ideal simulation.

Discrete-time Quantum Walk Benchmarking

The DTQW implementation produces a circuit with a relatively small depth and number of gates. The $\beta_{q|n}$ benchmark measures how close each quantum computer operates to the expected noise levels (indicated by the noise parameters and simulated by the UNM). Thus, in the small circuit case of the DTQW, the Bogota machine operates much closer to the expected evolution than the other computers with the smallest $\beta_{q|n}$.

The $\alpha_{q|i}$ and $\gamma_{n|i}$ benchmarks lead to a coherent picture of the overall performance of the quantum computers. In the DTQW case, the $\alpha_{q|i}$ benchmark is relatively small compared to the larger algorithms. This means that the quantum computers are not as erroneous as in the larger circuit cases, an expected result. A comparison between the $\alpha_{q|i}$ and $\gamma_{n|i}$ benchmarks shows that $\alpha_{q|i} < \gamma_{n|i}$ for all the machines, indicating that the the calibrated parameters overestimate the levels of noise during the evolution of the circuit. Finally, $\beta_{q|n} \geq |\alpha_{q|i} - \gamma_{q|i}|$ for all quantum computers, which implies high confidence for the noise level estimates.

Continuous-time Quantum Walk Benchmarking

The CTQW circuit is the smallest circuit used for benchmarking, which implies that the machine evolution will be reasonably close to the ideal case for this circuit. Evidently, the $\alpha_{q|i}$ benchmark is by far the smallest for the CTQW algorithm on all three machines. As $\alpha_{q|i} < \gamma_{n|i}$ for the Bogota and Santiago machines, the noise parameters overestimate the noise for those two computers, whereas they slightly underestimate the noise for the Casablanca computer.

Machine	$\alpha_{q\|i}$	$\beta_{q\|n}$	$\gamma_{n\|i}$	$\|\alpha_{q\|i} - \gamma_{n\|i}\|$	QC Runtime (sec)	Sim. Runtime (sec)
Bogota	0.287	0.025	0.291	0.004	240.7	3,331.5
Santiago	0.321	0.054	0.362	0.041	234.2	3,119.3
Casablanca	0.269	0.076	0.331	0.062	254.5	4,626.6

(a) Quantum walk algorithm on two qubits.

Machine	$\alpha_{q\|i}$	$\beta_{q\|n}$	$\gamma_{n\|i}$	$\|\alpha_{q\|i} - \gamma_{n\|i}\|$	QC Runtime (sec)	Sim. Runtime (sec)
Bogota	0.139	0.096	0.234	0.095	234.4	3,026.9
Santiago	0.154	0.035	0.184	0.030	301.7	2,905.7
Casablanca	0.247	0.044	0.239	0.008	262.1	3,035.2

(b) Continuous-time quantum walk algorithm on two qubits.

Machine	$\alpha_{q\|i}$	$\beta_{q\|n}$	$\gamma_{n\|i}$	$\|\alpha_{q\|i} - \gamma_{n\|i}\|$	QC Runtime (sec)	Sim. Runtime (sec)
Bogota	0.436	0.041	0.467	0.031	242.7	11,764.9
Santiago	0.514	0.084	0.465	0.049	228.5	11,618.6
Casablanca	0.577	0.039	0.612	0.035	376.4	12,861.3

(c) Pauli decomposition of the CTQW algorithm on two qubits.

Machine	$\alpha_{q\|i}$	$\beta_{q\|n}$	$\gamma_{n\|i}$	$\|\alpha_{q\|i} - \gamma_{n\|i}\|$	QC Runtime (sec)	Sim. Runtime (sec)
Bogota	0.469	0.177	0.351	0.118	254.2	6,287.1
Santiago	0.488	0.144	0.369	0.119	226.8	6,304.2
Casablanca	0.524	0.267	0.351	0.173	268.3	7,741.1

(d) Quantum phase estimation algorithm on three qubits.

Machine	$\alpha_{q\|i}$	$\beta_{q\|n}$	$\gamma_{n\|i}$	$\|\alpha_{q\|i} - \gamma_{n\|i}\|$	QC Runtime (sec)	Sim. Runtime (sec)
Bogota	0.754	0.014	0.752	0.002	270.4	22,834.3
Santiago	0.751	0.051	0.738	0.013	256.7	21,923.2
Casablanca (a)	0.760	0.040	0.752	0.008	280.1	36,425.7
Casablanca (na)	0.763	0.061	0.738	0.025	274.1	22,889.9

(e) Quantum search algorithm on four qubits.

Table 6.4 Results from benchmarking the three machines for each of the algorithms. The benchmark indicators $\alpha_{q|i}$, $\beta_{q|n}$ and $\gamma_{n|i}$ are the HD between the quantum computer and the UNM, the quantum computer and the ideal distribution and the UNM and the ideal distribution respectively. QC runtime is the cumulative execution time in seconds for 100,000 iterations of each algorithm on the respective machine; Sim. runtime is the time it takes to simulate for 100,000 iterations the behaviour of each machine when executing the algorithms.

For the CTQW, the Santiago machine operates closer to the expected levels of noise according to its error rates, with the smallest $\beta_{q|n}$. Similarly to the DTQW, the benchmarks point to confidence in the calibrated noise as $\beta_{q|n} \geq |\alpha_{q|i} - \gamma_{q|i}|$.

Pauli Decomposition Benchmarking

Unlike the CTQW circuit, its Pauli decomposition is a large and quite deep circuit, which means that one should expect a very noisy evolution. The small $\beta_{q|n}$ values show that the quantum computers operate relatively close to expectations.

As shown by the large values of $\alpha_{q|i}$ for all quantum computers, the PD circuit is quite error-prone compared to the two previous cases of the quantum walk. Thus, an important conclusion that can be safely drawn is that, for a two-qubit continuous-time quantum walk, the Pauli decomposition of its Hamiltonian leads to a more complex and noisier circuit than the decomposition to base gates done automatically through the IBMQ API.

The $\gamma_{n|i}$ benchmarks show that the parameters overestimate the noise of the Bogota and Casablanca machines and underestimate the noise within the Santiago computer. Furthermore, $\beta_{q|n} \geq |\alpha_{q|i} - \gamma_{q|i}|$ points to high-confidence on the noise estimates. This result is important as, through the benchmarks, there is evidence produced on the performance of two different techniques for implementing the same algorithm.

Quantum Phase Estimation Benchmarking

Moving on, the QPE circuit is slightly more complex than the DTQW circuit. An overall bigger $\beta_{q|n}$ benchmark on every machine indicates that the QPE circuit execution deviates slightly more from the expected evolution compared to the other algorithms. This could be the result of random fluctuations of the calibrated parameters.

Following the $\alpha_{q|i}$ benchmark, one can extract similar results for the quantum computers as the PD circuit benchmarks, finding that the larger values indicate that the quantum computer performance is hindered by the size of the circuit. Interestingly, it is evident that $\alpha_{q|i} > \gamma_{n|i}$ for all the quantum computers, showcasing that the calibrated parameters underestimate the noise and one can expect noisier results from all the quantum computers for computations of similar size. There is high-confidence of the calibrated parameters as $\beta_{q|n} \geq |\alpha_{q|i} - \gamma_{q|i}|$ for all experiments.

Quantum Search Benchmarking

Finally, the quantum search circuit represents the largest implemented algorithm and thus, it is expected that it will also be the noisiest. Following the above methodology, it is found that the $\beta_{q|n}$ benchmark shows the smallest values for each machine, indicating that the quantum computers operated close to the predicted evolutions. The $\alpha_{q|i}$ and $\gamma_{n|i}$ benchmarks are quite close, increasing the confidence on the estimated noise levels. Nevertheless, it is safe to conclude that the machines in the case of the QS exhibit intense levels of noise, as the $\alpha_{q|i}$ benchmarks are very large.

A quick comparison between the $\alpha_{q|i}$ and $\gamma_{n|i}$ shows that the calibrated parameters offer a very good picture of the noise within the quantum computer in this experiment.

Analysis

A comparison of the $\alpha_{q|i}$ and $\gamma_{n|i}$ benchmarks shows that when executing the DTQW and CTQW circuits, the machines are less error-prone than implied by the UNM simulations (and in extent, the associated calibrated noise parameters) with the opposite being true for the PD, QS and QPE. Thus, it is concluded that the machines are better at executing small circuits, an expected result. Additionally, a larger numerical difference between $\alpha_{q|i}$ and $\gamma_{n|i}$ indicates a more efficient computer behaviour. For example, Casablanca is the most efficient in terms of noise for the DTQW case as $\alpha_{q|i} < \gamma_{n|i}$ with the biggest difference. Similarly, in the QS case the Casablanca machine is the least efficient with $\alpha_{q|i} > \gamma_{n|i}$. Here, as there are two different implementations of the QS algorithm, it also is possible to compare them with each other. The benchmarks show that the QS circuit with ancilla is closer to the expected evolution by the UNM, shown by $\beta_{q|n}$, and to the ideal evolution, shown by $\alpha_{q|i}$, although not by much, shown by $\alpha_{q|i}$ versus $\gamma_{n|i}$.

The results from the above analysis are crucial as they can lead towards the selection of a machine appropriate to specific circuit needs. Additionally, a comparison between the continuous-time quantum walk circuit and its Pauli decomposition is provided. This fact is clearly reflected in the benchmarks, as the values of $\alpha_{q|i}$ are much lower for the CTQW circuit, while both circuits operate within expectations (small $\beta_{q|n}$ values) and are not massively over- or underestimated by their noise parameters (small $|\alpha_{q|i} - \gamma_{n|i}|$). Thus, it is concluded that, for the two-qubit case, a Pauli decomposition of the CTQW Hamiltonian leads to a less efficient circuit.

Overall, the general benchmarking results can be concentrated as follows. In the smallest circuits (i.e., CTQW and DTQW case) the $\beta_{q|n}$ benchmarks show how close the computers operate to the expected levels of noise. The Casablanca machine shows the best $\alpha_{q|i}$ benchmark, i.e., it is the closest to the ideal evolution, closely followed by Bogota and with Santiago being the furthest away. The UNM and the calibrated parameters always overestimate the noise in this case as $\alpha_{q|i} < \gamma_{n|i}$, thus showing that all the quantum computers exhibit relatively low levels of noise when executing small circuits. In the slightly deeper circuit of the QPE, the Bogota machine outperforms the others, followed by Santiago and Casablanca. In this case though the $\beta_{q|n}$ benchmark indicates slightly bigger deviation from the expected level of noise which is biggest on the Casablanca machine. The UNM and calibrated parameters always underestimate the noise in the quantum computer as $\alpha_{q|i} > \gamma_{n|i}$, meaning that the quantum computers are more error-prone. Lastly, in the largest QS and PD circuits, the benchmarks indicate that all the machines operate close to the noise model with low $\beta_{q|n}$ benchmarks and exhibit very low performance as the $\alpha_{q|i}$ benchmarks are large, getting close to 1 for the QS. The noise model slightly underestimates the noise, but in this case of very deep circuits, the benchmarks show that the machines will not produce any meaningful results, an expected outcome.

6.5 Discussion

This thesis has presented an approach to benchmarking quantum computers using scaling, high-level quantum algorithms considered as attractive "real-world" problems. To this end, three benchmark metrics are defined, each highlighting different aspects of the machine's efficiency either as a standalone or through comparisons between them. Each benchmark metric describes the difference between two quantum evolutions and together they follow the triangle inequality.

In order to better present the main characteristics of the benchmarks, the discussion can be streamlined as a comparison with the widely used metric of *quantum volume* [41], which quantifies the expected size of a circuit that can be reliably run on a quantum computer. In contrast, the program benchmarks showcase the performance of the quantum computer when running a specific circuit itself. This approach has advantages and disadvantages over architecture-neutral benchmarks. First of all, the benchmark metrics showcase the exact performance of a QPU when running the quantum circuit. Additionally, they highlight the difference with an expected evolution and an ideal evolution, a result that better identifies the weaknesses of the machine in a more structural and comprehensive manner. More specifically, the metrics allow one to realise the manner and intensity by which the computer deviates from the expected evolutions. Finally, as mentioned in Section 2.2.2, all three machines utilised for the experimental procedure possess the same QPU technology and exhibit the same quantum volume of 32. On the other hand, this benchmarking methodology exhibits different metrics and results for each quantum computer. Thus, the proposed metrics capture the performance of each QPU more thoroughly and provide a more detailed representation of their performance.

In terms of disadvantages, the proposed program benchmarking process is slower compared to the calculation of the quantum volume. This is an expected outcome as it is necessary to run a number of experiments on the quantum computer as well as the noisy and ideal simulations. Moreover, the flip-side of the architecture-specific nature of the benchmark metrics dictates that each computer will exhibit different benchmarks when executing different algorithms.

In conclusion, the proposed architecture-specific program benchmarks showcase the performance of a quantum computer in a "real-world" environment, highlighting their efficiency when running a specific algorithm, as well as carry out meaningful comparisons between related circuits (e.g., the CTQW vs the Pauli decomposition of its Hamiltonian). On the other hand, architecture-neutral benchmarks like the quantum volume are more generic and excel at showcasing the limitations of QPUs when running arbitrary quantum circuits.

This work has shown that using quantum algorithms to benchmark quantum computers in a well-structured environment can stress different aspects and very informatively highlight the performance of a quantum computer. Additionally, scaling quantum algorithms, when applied through the proposed benchmark framework, excel as methods for benchmarking near-term quantum computers. Finally, the resulting benchmark metrics represent an excellent indicator of the efficiency of the benchmarked machines when executing the quantum circuits.

Finally, this is the first work that uses a continuous-time quantum algorithm to benchmark the performance of a digital quantum machine. The results show that, for small state spaces of the continuous-time quantum walk, the Pauli decomposition does not produce an efficient circuit. This result is expected as the complexity of a Hamiltonian operating on two qubits is very small.

6.6 Future Work

A first and obvious extension to the work outlined in this chapter is the implementation of more algorithms as benchmark methods in the proposed framework. There is a large number of applications with appealing "real-world" traits that are perfectly suitable for such an endeavour, i.e., quantum machine learning, linear algebra, chemistry, and more. For example, interesting programs that could be used for future benchmarks include, but are not limited to, the variational quantum eigensolver (VQE), quantum approximate optimisation algorithms (QAOA), quantum annealing, and more. Furthermore, as showcased within the above discussion, the framework can be used to further compare approaches to quantum simulation, like two different decompositions of the CTQW (the Pauli and the automatic decomposition from the IBMQ backend). Thus, another topic for future work is to use the framework, or perhaps even further tailor it, to benchmark decompositions, simulations, simulators or circuit approaches to various quantum algorithms.

As it has been mentioned multiple times this far, the quantum computing field has in store bigger and better quantum computers. Therefore, an exciting prospect in the future of program benchmarking is experimenting with larger, newer quantum machines. Unfortunately, this work massively depends on simulations of the quantum evolution, which excels at showcasing the performance of the current computers but will, inevitably, struggle with larger QPUs. In other words, the proposed framework currently has an upper bound on what machines it is able to benchmark. A very interesting topic for future research would be to find ways to overcome this bound, for example, benchmark localised areas of the QPU or by running parallel quantum circuits on two different groups of qubits of the QPU, which then would lead to two separate but manageable quantum simulations.

Nevertheless, keeping in mind that the proposed methodology for program benchmarking quantum computers adheres to the current and near-term quantum computers that are more likely to be used for early commercial use and, evidently, excels at what it tries to achieve. It is the belief of this thesis that future benchmarks of potentially massive quantum computers will need alternative techniques altogether, that do not depend on classical simulations.

Finally, a prominent area of research in the field of quantum computing is the verification of quantum programs. In contrast with the framework developed in this chapter, formal verification aims to validate implementation of quantum algorithms. Considering the difficulty of implementing such complex routines in NISQ machines, research towards creating an arsenal of verification frameworks for quantum programs could lead to very beneficial results. An introduction to formal verification of quantum programs can be found in [187].

Chapter 7

Conclusions

7.1 Summary

Quantum computing is one of the fastest developing fields of research in the past couple of decades, concentrating increasing attention from both academia and industry. The large amounts of funding and research pouring into designing, building and improving quantum computers, as well as their applications, has brought forward advancements in the field that were once considered decades away [35, 188–190]. Nevertheless, some serious obstacles remain on the road towards universal quantum computation, chief amongst which is quantum noise. Hence, in order for quantum computers to fulfill their promises in creating a new technological era, it is essential to engineer ways that deal with obstacles like the quantum noise.

Considering the above, the present thesis identifies its contributions towards this end: obtaining further knowledge on the noise by modelling and simulating the noisy evolution within a quantum computer, additionally, creating a framework that allows the benchmarking of quantum computers in a scaling, algorithmic environment that caters for "real-world" applications. The research also concentrates around implementation characteristics of quantum walks, which are proven to be the perfect tool to assist in the study of noise and benchmarking.

7.2 Evaluation of Research Aims and Results

This thesis presents the work on modelling and simulating noise within quantum computers, as well as methods for program benchmarking near-term devices. The contributions have been established and proven theoretically, and also applied in numerous experiments both in a simulated environment and on real quantum computers.

The first research objective was focused on quantum walks. Within the field of quantum computing and beyond, this process is of great value and research interest, as it offers the means to design high-level quantum algorithms that exhibit strong quantum advantage. Considering the importance of quantum walks, Chapter 3 delves into an in-depth study of *two well-established implementations and the corresponding circuit* *characteristics*. Following a comprehensive proof of the quadratic quantum advantage shown by a quantum walk, the chapter presents a comprehensive theoretical analysis of the implementations, proving the complexity, benefits and detriments of each approach when executed on a real quantum computer. The experiments also showcase the effects of noise during the execution of each circuit on an IBMQ computer, further supporting the theoretical results. The above analysis also constitutes a proposed methodology for comparing the efficiency and performance of alternative implementations of arbitrary quantum circuits, an invaluable contribution within the NISQ era and beyond.

Moving on, the thesis turns its attention to one of the major contribution in quantum noise modelling. The objective in this area has been identified as *closely approximating the behaviour of real quantum computers*. To this end, the work in Chapter 4 concentrates on three separate and major sources of noise during a computation: (i) hardware infidelities, (ii) decoherence in the form of thermal energy exchange between the QPU and the environment and (iii) dephasing of the physical qubits. The thesis proposes an approach that combines the aforementioned three sources in a single model, named the *unified noise model* (UNM). The main characteristic of this model is that it is architecture-aware, i.e., it encodes knowledge of the qubit connectivity within the QPU and the levels of noise during the computation via utilising a set of quantum noise parameters. Chapter 4 offers the theoretical description of the UNM and a comprehensive review of its characteristics. Following, the simulation and experimental results show that the UNM excels at approximating the noisy evolution of an IBMQ computer, doing so considerably better than other state-of-the-art noise models (also implemented in IBMQ Qiskit [63]).

The experiments done in Chapter 4 show that even though the UNM creates better approximations of the noisy evolution during a quantum circuit execution, deviations still persist from the quantum computer experiments. This can be attributed to multiple factors, for example, the fact that the UNM does not take into account electromagnetic noise or cross-talk between the qubits, or due to estimation errors of the calibrated noise parameters. This thesis tries to tackle these deviations by concentrating on optimising the noise parameters used during the simulation of the UNM. Chapter 5 showcases the optimisation process, which is essentially a classical genetic algorithm optimisation routine tailored to estimate a set of noise parameters that perform better at approximating the evolution of the quantum computer than the hardware-calibrated ones, when used by the UNM. The results are very encouraging, with the post-optimisation simulations of the UNM showing up to 84% increase in the efficiency of the model for small quantum systems.

Finally, Chapter 6 concentrates on the last major contribution of the thesis, benchmarking quantum computers in a high-level and scaling environment. The approach followed by the present work is directed at using scaling quantum algorithms in order to *engineer a methodology for program benchmarking quantum machines*. After outlining the determining factors and challenges of program benchmarks, the chapter introduces the basic ideas for the proposed methodology. Three criteria are identified for selecting a "program" for the program benchmarks: (i) scalability, (ii) predictable evolution that is susceptible to noise and (iii) appealing real-world applications, i.e., quantum advantage. Following this, the proposed *framework for program benchmarking quantum computers* is outlined, using three novel benchmarks, $\alpha_{q|i}$, $\beta_{q|n}$ and $\gamma_{n|i}$. The experiments use the unified noise model for simulations and five quantum algorithms to benchmark three of the IBMQ publicly available computers. The results show that the proposed framework and benchmarks paint a finer picture of the performance of the quantum machines with respect to other approaches (eg., the quantum volume). Even though this framework is valuable within the NISQ era, its use beyond the near-term is limited by the inability of classical computers to simulate larger QPUs.

7.3 Thesis Conclusion

It is evident that the present thesis tries to work with some of the issues that generate a significant amount of discussion in the field of quantum computing. For the field of quantum machine learning, quantum walks and their implementations are crucial for designing and executing larger and more complex machine learning algorithms. Furthermore, one can argue that it is evident that quantum walks (both discrete- and continuous-time) can also prove very useful processes in the study of quantum noise and also as programs for benchmarking quantum computers.

As stated multiple times in this manuscript, quantum noise is one of the main challenges preventing universal and scalable quantum computation. Studying the theoretical and experimental results on unifying noise sources on a single model, it is safe to conclude that such an endeavour has been proven to produce a good approximation of the noisy circuit evolution. Additionally, hardware-calibrated noise parameters often produce simulations that deviate from the actual noise within the quantum computer. The proposed model is able to showcase this weakness, to present an insight on what the noise parameters look like within the simulated world and to assist on limiting the gap between calibrated and simulated noise parameters. The proposed approach to noise modelling can assist with the understanding of noise within quantum computers and consequently be utilised during the design or testing of error correcting methods or calibration techniques and attempts to minimise the noise in near-term quantum computers.

Finally, the work on program benchmarking, the proposed framework and the subsequent experiments can lead to the conclusion that indeed benchmarks produced using scaling, high-level algorithms have the ability to provide instructive and concrete indications on the performance of quantum computers. The arguments are mainly drawn by experimenting on three IBMQ computers, but nothing prevents the proposed methodology from being applied on any other quantum computer designed using any quantum technology, making the framework *universal.*

7.4 Future Work

Every chapter within this thesis asserts a number of propositions for future work. Current trends in the field indicate that quantum machines will get bigger and better, but researchers still have to deal with the excessive amounts of noise, especially in long computations. Thus, future work in the subject of characterising quantum circuit performance and efficiency in a similar way to that presented in Chapter 3 is of great value. This is not only limited to alternative implementations of quantum walks, but for the comparison of circuit implementations of any algorithm.

It is evident that, with more qubits being added to quantum computers, simulating the noisy evolution of quantum circuits will get progressively harder. This is a limitation not only of the unified noise model proposed in this thesis, but of any attempt to model. Nevertheless, with noise-free universal quantum computing still being an elusive target, noise analysis is still crucial. It is theorised in Chapter 4 that the gaps between the approximated and real quantum evolution can be partially filled by taking into account additional sources of noise, like electromagnetic decoherence, cross-talk between the qubits or Clifford errors. Thus, incorporating these sources in the UNM would constitute valuable future work.

Finally, in the context of program benchmarks, obtaining a comprehensive characterisation of the performance of a quantum computer gets increasingly important as quantum computers get increasingly bigger. Using the proposed framework of Chapter 6 in the future, one could gather information regarding the efficiency of a quantum computer or even evaluate how well a quantum computer is suited to executing an arbitrary computation. On the other hand, the proposed methodology depends on simulating the evolution of the quantum circuit, a process that gets increasingly taxing for larger QPUs. Further research could focus on clever ways to evaluate the program benchmarks, with some indicated examples outlined in Section 6.6. Furthermore, this thesis uses superconducting quantum computers made by IBM, primarily due to the fact that they are publicly available. It would be very interesting to compare the results from the experiments carried out on IBMQ computers with the same experiments implemented on superconducting machines manufactured by other companies, such as Rigetti or Google, or other technologies, such as trapped ions or photonic platforms.

References

[1] P. Benioff, “The computer as a physical system: A microscopic quantum mechanical hamiltonian model of computers as represented by turing machines,” *Journal of Statistical Physics*, vol. 22, no. 5, pp. 563–591, 1980.

[2] R. P. Feynman, “Simulating physics with computers,” *International Journal of Theoretical Physics*, vol. 21, no. 6, pp. 467–488, 1982.

[3] P. W. Shor, “Scheme for reducing decoherence in quantum computer memory,” *Phys. Rev. A*, vol. 52, pp. 2493–2496, Oct 1995.

[4] L. K. Grover, “A fast quantum mechanical algorithm for database search,” in *Proceedings of the Twenty-Eighth Annual ACM Symposium on Theory of Computing*, STOC ’96, (New York, NY, USA), p. 212–219, Association for Computing Machinery, 1996.

[5] H. F. Trotter, “On the product of semi-groups of operators,” *Proceedings of the American Mathematical Society*, vol. 10, no. 4, pp. 545–551, 1959.

[6] J. E. Cohen, S. Friedland, T. Kato, and F. P. Kelly, “Eigenvalue inequalities for products of matrix exponentials,” *Linear Algebra and its Applications*, vol. 45, pp. 55–95, 1982.

[7] S. Lloyd, “Universal quantum simulators,” *Science*, vol. 273, no. 5278, pp. 1073–1078, 1996.

[8] M. Szegedy, “Quantum speed-up of Markov chain based algorithms,” in *Proceedings of the 45th Annual IEEE Symposium on Foundations of Computer Science*, FOCS ’04, (Washington, DC, USA), pp. 32–41, IEEE Computer Society, 2004.

[9] A. M. Childs, R. Cleve, E. Deotto, S. Farhi, E.and Gutmann, and D. A. Spielman, “Exponential algorithmic speedup by a quantum walk,” *Proceedings of the thirty-fifth ACM symposium on Theory of computing - STOC ’03*, 2003.

[10] P. C. Richter, “The quantum complexity of markov chain monte carlo,” in *Logic and Theory of Algorithms*, (Berlin, Heidelberg), pp. 511–522, Springer Berlin Heidelberg, 2008.

[11] T. G. Wong, “Unstructured search by random and quantum walk,” arXiv:2011.14533 [quant-ph], 2020.

[12] F. Magniez, A. Nayak, J. Roland, and M. Santha, “Search via quantum walk,” in *Proceedings of the Thirty-ninth Annual ACM Symposium on Theory of Computing*, STOC ’07, (New York, NY, USA), pp. 575–584, ACM, 2007.

[13] W. Zhou, “Review on quantum walk algorithm,” *Journal of Physics: Conference Series*, vol. 1748, p. 032022, jan 2021.

[14] N. Konno, *Quantum Walks*, pp. 309–452. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008.

[15] K. Temme, T. J. Osborne, K. G. Vollbrecht, D. Poulin, and F. Verstraete, "Quantum Metropolis sampling," *Nature*, vol. 471, no. 7336, pp. 87–90, 2011.

[16] M.-H. Yung and A. Aspuru-Guzik, "A quantum–quantum metropolis algorithm," *Proceedings of the National Academy of Sciences*, vol. 109, no. 3, pp. 754–759, 2012.

[17] A. W. Harrow, A. Hassidim, and S. Lloyd, "Quantum algorithm for linear systems of equations," *Phys. Rev. Lett.*, vol. 103, p. 150502, Oct 2009.

[18] A. Ambainis, E. Bach, A. Nayak, A. Vishwanath, and J. Watrous, "One-dimensional quantum walks," in *Proceedings of the Thirty-third Annual ACM Symposium on Theory of Computing*, STOC '01, (New York, NY, USA), pp. 37–49, ACM, 2001.

[19] B. L. Douglas and J. B. Wang, "Efficient quantum circuit implementation of quantum walks," *Phys. Rev. A*, vol. 79, p. 052335, May 2009.

[20] H. Krovi, "Symmetry in quantum walks," arXiv:0711.1694 [quant-ph], 2007.

[21] D. Reitzner, D. Nagaj, and V. Bužek, "Quantum walks," *Acta Physica Slovaca Reviews and Tutorials*, vol. 61, no. 6, pp. 603–725, 2011.

[22] A. C. Orthey, Jr. and E. P. M. Amorim, "On the spreading of quantum walks starting from local and delocalized states," 2017.

[23] E. Farhi and S. Gutmann, "Quantum computation and decision trees," *Phys. Rev. A*, vol. 58, pp. 915–928, Aug 1998.

[24] V. Kendon, "Decoherence in quantum walks – a review," *Mathematical Structures in Computer Science*, vol. 17, Nov 2007.

[25] D. Aharonov, A. Ambainis, J. Kempe, and U. Vazirani, "Quantum walks on graphs," in *Proceedings of the Thirty-third Annual ACM Symposium on Theory of Computing*, STOC '01, (New York, NY, USA), pp. 50–59, ACM, 2001.

[26] A. Ahmadi, R. Belk, C. Tamon, and C. Wendler, "Mixing in continuous quantum walks on graphs," arXiv:0209106 [quant-ph], 2002.

[27] S. Chakraborty, L. Novo, and J. Roland, "Finding a marked node on any graph via continuous-time quantum walks," *Phys. Rev. A*, vol. 102, p. 022227, Aug 2020.

[28] A. Ambainis, A. Gilyén, S. Jeffery, and M. Kokainis, "Quadratic speedup for finding marked vertices by quantum walks," in *Proceedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing*, STOC 2020, (New York, NY, USA), p. 412–424, Association for Computing Machinery, 2020.

[29] A. M. Childs, "On the relationship between continuous- and discrete-time quantum walk," *Communications in Mathematical Physics*, vol. 294, p. 581–603, Oct 2009.

[30] M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information: 10th Anniversary Edition.* Cambridge University Press, 2010.

[31] H. Ball, W. D. Oliver, and M. J. Biercuk, "The role of master clock stability in quantum information processing," *npj Quantum Information*, vol. 2, no. 1, p. 16033, 2016.

[32] 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*, vol. 6, p. 021318, Jun 2019.

[33] D. M. Berns, W. D. Oliver, S. O. Valenzuela, A. V. Shytov, K. K. Berggren, L. S. Levitov, and T. P. Orlando, "Coherent quasiclassical dynamics of a persistent current qubit," *Phys. Rev. Lett.*, vol. 97, p. 150502, Oct 2006.

[34] J. Preskill, "Lecture notes in quantum computation, chapter 3," California Institute of Technology, October 2018.

[35] J. Preskill, "Quantum computing in the nisq era and beyond," *Quantum*, vol. 2, p. 79, Aug 2018.

[36] S. Boixo, S. V. Isakov, V. N. Smelyanskiy, R. Babbush, N. Ding, Z. Jiang, M. J. Bremner, J. M. Martinis, and H. Neven, "Characterizing quantum supremacy in near-term devices," *Nature Physics*, vol. 14, p. 595–600, Apr 2018.

[37] S. Aaronson and L. Chen, "Complexity-theoretic foundations of quantum supremacy experiments," in *Proceedings of the 32nd Computational Complexity Conference*, CCC '17, (Dagstuhl, DEU), Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2017.

[38] E. Pednault, J. A. Gunnels, G. Nannicini, L. Horesh, T. Magerlein, E. Solomonik, E. W. Draeger, E. T. Holland, and R. Wisnieff, "Pareto-efficient quantum circuit simulation using tensor contraction deferral," 2020.

[39] C. J. Ballance, T. P. Harty, N. M. Linke, M. A. Sepiol, and D. M. Lucas, "High-fidelity quantum logic gates using trapped-ion hyperfine qubits," *Phys. Rev. Lett.*, vol. 117, p. 060504, Aug 2016.

[40] F. Lecocq, F. Quinlan, K. Cicak, J. Aumentado, S. A. Diddams, and J. D. Teufel, "Control and readout of a superconducting qubit using a photonic link," *Nature*, vol. 591, no. 7851, pp. 575–579, 2021.

[41] N. Moll, P. Barkoutsos, L. S. Bishop, J. M. Chow, A. Cross, D. J. Egger, S. Filipp, A. Fuhrer, J. M. Gambetta, M. Ganzhorn, and et al., "Quantum optimization using variational algorithms on near-term quantum devices," *Quantum Science and Technology*, vol. 3, p. 030503, Jun 2018.

[42] J. Kempe, "Quantum random walks: An introductory overview," *Contemporary Physics*, vol. 44, pp. 307–327, Jul 2003.

[43] P. C. Richter, "Quantum speedup of classical mixing processes," *Physical Review A*, vol. 76, Oct 2007.

[44] N. Shenvi, J. Kempe, and K. Birgitta Whaley, "Quantum random-walk search algorithm," *Physical Review A*, vol. 67, May 2003.

[45] A. N. Chowdhury and R. D. Somma, "Quantum algorithms for gibbs sampling and hitting-time estimation," *Quantum Info. Comput.*, vol. 17, pp. 41–64, Feb. 2017.

[46] A. Montanaro, "Quantum speedup of Monte Carlo methods," *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 471, Sep 2015.

[47] A. Dhahri, C. K. Ko, and H. J. Yoo, "Quantum Markov chains associated with open quantum random walks," *Journal of Statistical Physics*, vol. 176, pp. 1272–1295, Jun 2019.

[48] A. Ambainis, "Quantum walk algorithm for element distinctness," *SIAM Journal on Computing*, vol. 37, no. 1, pp. 210–239, 2007.

[49] A. M. Childs and J. Goldstone, "Spatial search by quantum walk," *Physical Review A*, vol. 70, Aug 2004.

[50] A. Montanaro, "Quantum-walk speedup of backtracking algorithms," *Theory of Computing*, vol. 14, no. 15, pp. 1–24, 2018.

[51] J. Kempe, "Discrete quantum walks hit exponentially faster," *Probability Theory and Related Fields*, vol. 133, pp. 215–235, Oct 2005.

[52] L. Sansoni, F. Sciarrino, G. Vallone, P. Mataloni, A. Crespi, R. Ramponi, and R. Osellame, "Two-particle bosonic-fermionic quantum walk via integrated photonics," *Phys. Rev. Lett.*, vol. 108, p. 010502, Jan 2012.

[53] T. Rakovszky and J. K. Asboth, "Localization, delocalization, and topological phase transitions in the one-dimensional split-step quantum walk," *Physical Review A*, vol. 92, Nov 2015.

[54] S. Mugel, A. Celi, P. Massignan, J. K. Asboth, M. Lewenstein, and C. Lobo, "Topological bound states of a quantum walk with cold atoms," *Physical Review A*, vol. 94, Aug 2016.

[55] M. Grafe and A. Szameit, "Integrated photonic quantum walks," *Journal of Physics B: Atomic, Molecular and Optical Physics*, vol. 53, p. 073001, mar 2020.

[56] M. A. Broome, A. Fedrizzi, B. P. Lanyon, I. Kassal, A. Aspuru-Guzik, and A. G. White, "Discrete single-photon quantum walks with tunable decoherence," *Phys. Rev. Lett.*, vol. 104, p. 153602, Apr 2010.

[57] A. Schreiber, K. N. Cassemiro, V. Potoček, A. Gábris, P. J. Mosley, E. Andersson, I. Jex, and C. Silberhorn, "Photons walking the line: A quantum walk with adjustable coin operations," *Phys. Rev. Lett.*, vol. 104, p. 050502, Feb 2010.

[58] M. Karski, L. Förster, J. Choi, A. Steffen, W. Alt, D. Meschede, and A. Widera, "Quantum walk in position space with single optically trapped atoms," *Science*, vol. 325, no. 5937, pp. 174–177, 2009.

[59] C. Robens, W. Alt, D. Meschede, C. Emary, and A. Alberti, "Ideal negative measurements in quantum walks disprove theories based on classical trajectories," *Physical Review X*, vol. 5, Jan 2015.

[60] H. Schmitz, R. Matjeschk, C. Schneider, J. Glueckert, M. Enderlein, T. Huber, and T. Schaetz, "Quantum walk of a trapped ion in phase space," *Phys. Rev. Lett.*, vol. 103, p. 090504, Aug 2009.

[61] K. Georgopoulos, C. Emary, and P. Zuliani, "Comparison of quantum-walk implementations on noisy intermediate-scale quantum computers," *Physical Review A*, vol. 103, Feb 2021.

[62] A. Nayak and A. Vishwanath, "Quantum walk on the line," arXiv:0010117 [quant-ph], 2000.

[63] "IBM Qiskit." https://qiskit.org/, Last accessed May 2021.

[64] "IBM Quantum Experience." https://www.ibm.com/quantum-computing/technology/experience, Last accessed May 2021.

[65] Y. Cao, G. C. Wang, H. D. Liu, and C. F. Sun, "Implementation of a toffoli gate using an array of coupled cavities in a single step," *Scientific Reports*, vol. 8, no. 1, p. 5813, 2018.

[66] V. V. Shende and I. L. Markov, "On the cnot-cost of toffoli gates," *Quantum Info. Comput.*, vol. 9, p. 461–486, May 2009.

[67] A. Barenco, C. H. Bennett, R. Cleve, D. P. DiVincenzo, N. Margolus, P. Shor, T. Sleator, J. A. Smolin, and H. Weinfurter, "Elementary gates for quantum computation," *Phys. Rev. A*, vol. 52, pp. 3457–3467, Nov 1995.

[68] Y. Liu, G. L. Long, and Y. Sun, "Analytic one-bit and CNOT gate constructions of general n-qubit controlled gates," *International Journal of Quantum Information*, vol. 06, no. 03, pp. 447–462, 2008.

[69] D. Gottesman, "An introduction to quantum error correction and fault-tolerant quantum computation," *Quantum Information Science and Its Contributions to Mathematics, Proceedings of Symposia in Applied Mathematics*, pp. 13–18, 2010.

[70] E. Knill, R. Laflamme, and W. H. Zurek, "Resilient quantum computation: error models and thresholds," *Proceedings of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences*, vol. 454, pp. 365–384, Jan 1998.

[71] A. R. Calderbank and P. W. Shor, "Good quantum error-correcting codes exist," *Phys. Rev. A*, vol. 54, pp. 1098–1105, Aug 1996.

[72] D. P. DiVincenzo and P. W. Shor, "Fault-tolerant error correction with efficient quantum codes," *Phys. Rev. Lett.*, vol. 77, pp. 3260–3263, Oct 1996.

[73] E. Knill, "Quantum computing with realistically noisy devices," *Nature*, vol. 434, no. 7029, pp. 39–44, 2005.

[74] D. Bacon, "Operator quantum error-correcting subsystems for self-correcting quantum memories," *Phys. Rev. A*, vol. 73, p. 012340, Jan 2006.

[75] A. Y. Kitaev, "Fault-tolerant quantum computation by anyons," *Annals of Physics*, vol. 303, no. 1, pp. 2 – 30, 2003.

[76] E. Dennis, A. Kitaev, A. Landahl, and J. Preskill, "Topological quantum memory," *Journal of Mathematical Physics*, vol. 43, pp. 4452–4505, Sep 2002.

[77] H. Bombin and M. A. Martin-Delgado, "Topological quantum distillation," *Phys. Rev. Lett.*, vol. 97, p. 180501, Oct 2006.

[78] R. Duan, M. Grassl, Z. Ji, and B. Zeng, "Multi-error-correcting amplitude damping codes," *2010 IEEE International Symposium on Information Theory*, Jun 2010.

[79] H. Bombin, "Topological subsystem codes," *Phys. Rev. A*, vol. 81, p. 032301, Mar 2010.

[80] P. Aliferis and J. Preskill, "Fault-tolerant quantum computation against biased noise," *Physical Review A*, vol. 78, Nov 2008.

[81] D. K. Tuckett, S. D. Bartlett, and S. T. Flammia, "Ultrahigh error threshold for surface codes with biased noise," *Physical Review Letters*, vol. 120, Jan 2018.

[82] J. M. Martinis, "Qubit metrology for building a fault-tolerant quantum computer," *npj Quantum Information*, vol. 1, no. 1, p. 15005, 2015.

[83] A. W. Cross, D. P. Divincenzo, and B. M. Terhal, "A comparative code study for quantum fault tolerance," *Quantum Info. Comput.*, vol. 9, pp. 541–572, July 2009.

[84] M. Gutièrrez, L. Svec, A. Vargo, and K. R. Brown, "Approximation of realistic errors by clifford channels and pauli measurements," *Physical Review A*, vol. 87, Mar 2013.

[85] K. Georgopoulos, C. Emary, and P. Zuliani, "Modeling and simulating the noisy behavior of near-term quantum computers," *Phys. Rev. A*, vol. 104, p. 062432, Dec 2021.

[86] K. R. Brown, A. W. Harrow, and I. L. Chuang, "Arbitrarily accurate composite pulse sequences," *Phys. Rev. A*, vol. 70, p. 052318, Nov 2004.

[87] L. Viola, E. Knill, and S. Lloyd, "Dynamical decoupling of open quantum systems," *Phys. Rev. Lett.*, vol. 82, pp. 2417–2421, Mar 1999.

[88] A. Bassi, A. Grobardt, and H. Ulbricht, "Gravitational decoherence," *Classical and Quantum Gravity*, vol. 34, p. 193002, Sep 2017.

[89] C. Pfister, J. Kaniewski, M. Tomamichel, A. Mantri, R. Schmucker, N. McMahon, G. Milburn, and S. Wehner, "A universal test for gravitational decoherence," *Nature Communications*, vol. 7, no. 1, p. 13022, 2016.

[90] I. Pikovski, M. Zych, F. Costa, and Č. Brukner, "Universal decoherence due to gravitational time dilation," *Nature Physics*, vol. 11, no. 8, pp. 668–672, 2015.

[91] S. Boixo, S. V. Isakov, V. N. Smelyanskiy, R. Babbush, N. Ding, Z. Jiang, M. J. Bremner, J. M. Martinis, and H. Neven, "Characterizing quantum supremacy in near-term devices," *Nature Physics*, vol. 14, pp. 595–600, Apr 2018.

[92] S. Aaronson and S. Gunn, "On the classical hardness of spoofing linear cross-entropy benchmarking," arXiv:1910.12085 [quant-ph], 2019.

[93] J. Yuen-Zhou, J. J. Krich, I. Kassal, A. S. Johnson, and A. Aspuru-Guzik, "The process matrix and how to determine it: quantum process tomography," in *Ultrafast Spectroscopy*, 2053-2563, pp. 1–9, IOP Publishing, 2014.

[94] J. L. O'Brien, G. J. Pryde, A. Gilchrist, D. F. V. James, N. K. Langford, T. C. Ralph, and A. G. White, "Quantum process tomography of a controlled-not gate," *Physical Review Letters*, vol. 93, Aug 2004.

[95] M. Mohseni, A. T. Rezakhani, and D. A. Lidar, "Quantum-process tomography: Resource analysis of different strategies," *Physical Review A*, vol. 77, Mar 2008.

[96] E. Onorati, A. H. Werner, and J. Eisert, "Randomized benchmarking for individual quantum gates," *Phys. Rev. Lett.*, vol. 123, p. 060501, Aug 2019.

[97] E. Magesan, J. M. Gambetta, and J. Emerson, "Scalable and robust randomized benchmarking of quantum processes," *Phys. Rev. Lett.*, vol. 106, p. 180504, May 2011.

[98] C. C. López, B. Lévi, and D. G. Cory, "Error characterization in quantum information processing: A protocol for analyzing spatial correlations and its experimental implementation," *Phys. Rev. A*, vol. 79, p. 042328, Apr 2009.

[99] E. Knill, D. Leibfried, R. Reichle, J. Britton, R. B. Blakestad, J. D. Jost, C. Langer, R. Ozeri, S. Seidelin, and D. J. Wineland, "Randomized benchmarking of quantum gates," *Physical Review A*, vol. 77, January 2008.

[100] J. Helsen, X. Xue, L. M. K. Vandersypen, and S. Wehner, "A new class of efficient randomized benchmarking protocols," *npj Quantum Information*, vol. 5, p. 71, Aug. 2019.

[101] J. Emerson, R. Alicki, and K. Życzkowski, "Scalable noise estimation with random unitary operators," *Journal of Optics B: Quantum and Semiclassical Optics*, vol. 7, pp. 347–352, sep 2005.

[102] C. Dankert, R. Cleve, J. Emerson, and E. Livine, "Exact and approximate unitary 2-designs and their application to fidelity estimation," *Phys. Rev. A*, vol. 80, p. 012304, Jul 2009.

[103] A. W. Cross, E. Magesan, L. S. Bishop, J. A. Smolin, and J. M. Gambetta, "Scalable randomised benchmarking of non-Clifford gates," *npj Quantum Information*, vol. 2, Apr 2016.

[104] A. Erhard, J. J. Wallman, L. Postler, M. Meth, R. Stricker, E. A. Martinez, P. Schindler, T. Monz, J. Emerson, and R. Blatt, "Characterizing large-scale quantum computers via cycle benchmarking," *Nature Communications*, vol. 10, Nov 2019.

[105] A. Carignan-Dugas, J. J. Wallman, and J. Emerson, "Characterizing universal gate sets via dihedral benchmarking," *Phys. Rev. A*, vol. 92, p. 060302, Dec 2015.

[106] M. M. Wilde, *Quantum Information Theory.* Cambridge University Press, 2017.

[107] C. King, "The capacity of the quantum depolarizing channel," *IEEE Transactions on Information Theory*, vol. 49, no. 1, pp. 221–229, 2003.

[108] Z. Ji, G. Wang, R. Duan, Y. Feng, and M. Ying, "Parameter estimation of quantum channels," *IEEE Transactions on Information Theory*, vol. 54, pp. 5172–5185, Nov 2008.

[109] C. Blank, D. K. Park, J.-K. K. Rhee, and F. Petruccione, "Quantum classifier with tailored quantum kernel," *npj Quantum Information*, vol. 6, no. 1, p. 41, 2020.

[110] X. Y. Jin, A. Kamal, A. P. Sears, T. Gudmundsen, D. Hover, J. Miloshi, R. Slattery, F. Yan, J. Yoder, T. P. Orlando, S. Gustavsson, and W. D. Oliver, "Thermal and residual excited-state population in a 3d transmon qubit," *Phys. Rev. Lett.*, vol. 114, p. 240501, Jun 2015.

[111] M.-D. Choi, "Positive linear maps on C*-algebras," *Canadian Journal of Mathematics*, vol. 24, no. 3, pp. 520–529, 1972.

[112] M.-D. Choi, "Completely positive linear maps on complex matrices," *Linear Algebra and its Applications*, vol. 10, no. 3, pp. 285–290, 1975.

[113] K. Georgopoulos, "Combined quantum noise model for simulating the noisy behaviour of nisq machines." GitHub, https://github.com/kgeorgopoulos2/qnoisecomb.

[114] Z.-X. Jin and S.-M. Fei, "Quantifying quantum coherence and nonclassical correlation based on Hellinger distance," *Physical Review A*, vol. 97, Jun 2018.

[115] S. T. Merkel, J. M. Gambetta, J. A. Smolin, S. Poletto, A. D. Córcoles, B. R. Johnson, C. A. Ryan, and M. Steffen, "Self-consistent quantum process tomography," *Phys. Rev. A*, vol. 87, p. 062119, Jun 2013.

[116] D. Kim, D. R. Ward, C. B. Simmons, D. E. Savage, M. G. Lagally, M. Friesen, S. N. Coppersmith, and M. A. Eriksson, "High-fidelity resonant gating of a silicon-based quantum dot hybrid qubit," *npj Quantum Information*, vol. 1, no. 1, p. 15004, 2015.

[117] J. P. Dehollain, J. T. Muhonen, R. Blume-Kohout, K. M. Rudinger, J. K. Gamble, E. Nielsen, A. Laucht, S. Simmons, R. Kalra, A. S. Dzurak, and A. Morello, "Optimization of a solid-state electron spin qubit using gate set tomography," *New Journal of Physics*, vol. 18, 10 2016.

[118] R. Blume-Kohout, J. K. Gamble, E. Nielsen, K. Rudinger, J. Mizrahi, K. Fortier, and P. Maunz, "Demonstration of qubit operations below a rigorous fault tolerance threshold with gate set tomography," *Nature Communications*, vol. 8, no. 1, p. 14485, 2017.

[119] M. Ware, G. Ribeill, D. Riste, C. A. Ryan, B. Johnson, and M. P. da Silva, "Experimental pauli-frame randomization on a superconducting qubit," *Physical Review A*, vol. 103, Apr 2021.

[120] T. Proctor, M. Revelle, E. Nielsen, K. Rudinger, D. Lobser, P. Maunz, R. Blume-Kohout, and K. Young, "Detecting and tracking drift in quantum information processors," *Nature Communications*, vol. 11, Oct 2020.

[121] S. S. Hong, A. T. Papageorge, P. Sivarajah, G. Crossman, N. Didier, A. M. Polloreno, E. A. Sete, S. W. Turkowski, M. P. da Silva, and B. R. Johnson, "Demonstration of a parametrically activated entangling gate protected from flux noise," *Phys. Rev. A*, vol. 101, p. 012302, Jan 2020.

[122] M. K. Joshi, A. Elben, B. Vermersch, T. Brydges, C. Maier, P. Zoller, R. Blatt, and C. F. Roos, "Quantum information scrambling in a trapped-ion quantum simulator with tunable range interactions," *Phys. Rev. Lett.*, vol. 124, p. 240505, Jun 2020.

[123] E. Nielsen, K. Rudinger, J. K. Gamble, and R. Blume-Kohout, "A python implementation of gate set tomography." http://www.pygsti.info/, Last accessed November 2021.

[124] E. Nielsen, K. Rudinger, T. Proctor, A. Russo, K. Young, and R. Blume-Kohout, "Probing quantum processor performance with pygsti," *Quantum Science and Technology*, vol. 5, p. 044002, Jul 2020.

[125] E. Nielsen, J. K. Gamble, K. Rudinger, T. Scholten, K. Young, and R. Blume-Kohout, "Gate set tomography," 2020.

[126] S. E. Smart, Z. Hu, S. Kais, and D. A. Mazziotti, "Relaxation of a stationary state on a quantum computer yields unique spectroscopic fingerprint of the computer's noise," arXiv:2104.14552 [quantu-ph], 2021.

[127] X. Cai, "Quantum dephasing induced by non-markovian random telegraph noise," *Scientific Reports*, vol. 10, pp. 2045–2322, Oct 2020.

[128] J. Roffe, "Quantum error correction: an introductory guide," *Contemporary Physics*, vol. 60, pp. 226–245, Jul 2019.

[129] S. J. Devitt, W. J. Munro, and K. Nemoto, "Quantum error correction for beginners," *Reports on Progress in Physics*, vol. 76, p. 076001, Jun 2013.

[130] S. M. Barnett and B. J. Dalton, "Conceptions of quantum optical phase," *Physica Scripta*, vol. T48, pp. 13–21, jan 1993.

[131] S. Daffer, K. Wódkiewicz, and J. K. McIver, "Quantum markov channels for qubits," *Phys. Rev. A*, vol. 67, p. 062312, Jun 2003.

[132] J. J. Hope, G. M. Moy, M. J. Collett, and C. M. Savage, "Steady-state quantum statistics of a non-markovian atom laser," *Phys. Rev. A*, vol. 61, p. 023603, Jan 2000.

[133] J. M. Martinis, K. B. Cooper, R. McDermott, M. Steffen, M. Ansmann, K. D. Osborn, K. Cicak, S. Oh, D. P. Pappas, R. W. Simmonds, and C. C. Yu, "Decoherence in josephson qubits from dielectric loss," *Phys. Rev. Lett.*, vol. 95, p. 210503, Nov 2005.

[134] O. Astafiev, Y. A. Pashkin, Y. Nakamura, T. Yamamoto, and J. S. Tsai, "Quantum noise in the josephson charge qubit," *Phys. Rev. Lett.*, vol. 93, p. 267007, Dec 2004.

[135] D. Ahn, J. Lee, M. S. Kim, and S. W. Hwang, "Self-consistent non-markovian theory of a quantum-state evolution for quantum-information processing," *Phys. Rev. A*, vol. 66, p. 012302, Jul 2002.

[136] T. Weber, M. Riebisch, K. Borras, K. Jansen, and D. Krucker, "Modelling for quantum error mitigation," *2021 IEEE 18th International Conference on Software Architecture Companion (ICSA-C)*, Mar 2021.

[137] M. Gutiérrez and K. R. Brown, "Comparison of a quantum error-correction threshold for exact and approximate errors," *Phys. Rev. A*, vol. 91, p. 022335, Feb 2015.

[138] R. Harper, S. T. Flammia, and J. J. Wallman, "Efficient learning of quantum noise," *Nature Physics*, vol. 16, no. 12, pp. 1184–1188, 2020.

[139] Y. I. Bogdanov, A. Y. Chernyavskiy, A. Holevo, V. F. Lukichev, and A. A. Orlikovsky, "Modeling of quantum noise and the quality of hardware components of quantum computers," *International Conference Micro- and Nano-Electronics 2012*, Jan 2013.

[140] B. Nachman, M. Urbanek, W. A. de Jong, and C. W. Bauer, "Unfolding quantum computer readout noise," *npj Quantum Information*, vol. 6, no. 1, p. 84, 2020.

[141] M. L. Dahlhauser and T. S. Humble, "Modeling noisy quantum circuits using experimental characterization," *Phys. Rev. A*, vol. 103, p. 042603, Apr 2021.

[142] M. Mitchell, *An Introduction to Genetic Algorithms.* Cambridge, MA, USA: MIT Press, 1998.

[143] M. M. Navarro and B. B. Navarro, "Evaluations of crossover and mutation probability of genetic algorithm in an optimal facility layout problem," in *Parallel Problem Solving from Nature PPSN VI*, (Kuala Lumpur, Malaysia), pp. 89–98, Proceedings of the 2016 International Conference on Industrial Engineering and Operations Management, 2016.

[144] T. Jansen and I. Wegener, "On the choice of the mutation probability for the (1+1) ea," in *Parallel Problem Solving from Nature PPSN VI*, (Berlin, Heidelberg), pp. 89–98, Springer Berlin Heidelberg, 2000.

[145] R. Greenwell, J. Angus, and M. Finck, "Optimal mutation probability for genetic algorithms," *Mathematical and Computer Modelling*, vol. 21, no. 8, pp. 1–11, 1995.

[146] R. Lahoz-Beltra, "Quantum genetic algorithms for computer scientists," *Computers*, vol. 5, no. 4, 2016.

[147] A. Malossini, E. Blanzieri, and T. Calarco, "Quantum genetic optimization," *IEEE Transactions on Evolutionary Computation*, vol. 12, no. 2, pp. 231–241, 2008.

[148] J. King, M. Mohseni, W. Bernoudy, A. Fréchette, H. Sadeghi, S. V. Isakov, H. Neven, and M. H. Amin, "Quantum-assisted genetic algorithm," arxiv.1907.00707 [quant.ph, 2019.

[149] K. Georgopoulos, C. Emary, and P. Zuliani, "Quantum computer benchmarking via quantum algorithms," arXiv:2112.09457 [quant-ph], 2021.

[150] E. Bernstein and U. Vazirani, "Quantum complexity theory," *SIAM Journal on Computing*, vol. 26, no. 5, pp. 1411–1473, 1997.

[151] W. van Dam, S. Hallgren, and L. Ip, “Quantum algorithms for some hidden shift problems,” *SIAM Journal on Computing*, vol. 36, no. 3, pp. 763–778, 2006.

[152] M. Rötteler, *Quantum algorithms for highly non-linear Boolean functions*, pp. 448–457. 2010.

[153] K. Wright, K. M. Beck, S. Debnath, and et. al., “Benchmarking an 11-qubit quantum computer,” *Nature Communications*, vol. 10, no. 1, p. 5464, 2019.

[154] F. Arute, K. Arya, R. Babbush, and et. al., “Quantum supremacy using a programmable superconducting processor,” *Nature*, vol. 574, no. 7779, pp. 505–510, 2019.

[155] S. Kwon, A. Tomonaga, G. Lakshmi Bhai, S. J. Devitt, and J.-S. Tsai, “Gate-based superconducting quantum computing,” *Journal of Applied Physics*, vol. 129, no. 4, p. 041102, 2021.

[156] C. D. Bruzewicz, J. Chiaverini, R. McConnell, and J. M. Sage, “Trapped-ion quantum computing: Progress and challenges,” *Applied Physics Reviews*, vol. 6, p. 021314, Jun 2019.

[157] M. Veldhorst, H. G. J. Eenink, C. H. Yang, and A. S. Dzurak, “Silicon cmos architecture for a spin-based quantum computer,” *Nature Communications*, vol. 8, no. 1, p. 1766, 2017.

[158] S. Slussarenko and G. J. Pryde, “Photonic quantum information processing: A concise review,” *Applied Physics Reviews*, vol. 6, p. 041303, Dec 2019.

[159] N. M. Linke, D. Maslov, M. Roetteler, S. Debnath, C. Figgatt, K. A. Landsman, K. Wright, and C. Monroe, “Experimental comparison of two quantum computing architectures,” *Proceedings of the National Academy of Sciences of the United States of America*, vol. 114, pp. 3305–3310, March 2017.

[160] D. J. Lilja, *Measuring Computer Performance: A Practitioner’s Guide.* Cambridge University Press, 2000.

[161] M. Martonosi and M. Roetteler, “Next steps in quantum computing: Computer science’s role,” arXiv:1903.10541 [cs.ET], 2019.

[162] S. Gudder, “Quantum Markov chains,” *Journal of Mathematical Physics*, vol. 49, p. 072105, July 2008.

[163] P. W. Shor, “Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer,” *SIAM Journal on Computing*, vol. 26, pp. 1484–1509, Oct 1997.

[164] J. Olson, Y. Cao, J. Romero, P. Johnson, P.-L. Dallaire-Demers, N. Sawaya, P. Narang, I. Kivlichan, M. Wasielewski, and A. Aspuru-Guzik, “Quantum information and computation for chemistry,” 2017.

[165] S. McArdle, S. Endo, A. Aspuru-Guzik, S. C. Benjamin, and X. Yuan, “Quantum computational chemistry,” *Rev. Mod. Phys.*, vol. 92, p. 015003, Mar 2020.

[166] Y. Cao, J. Romero, J. P. Olson, M. Degroote, P. D. Johnson, M. Kieferová, I. D. Kivlichan, T. Menke, B. Peropadre, N. P. D. Sawaya, S. Sim, L. Veis, and A. Aspuru-Guzik, “Quantum chemistry in the age of quantum computing,” *Chemical Reviews*, vol. 119, pp. 10856–10915, 10 2019.

[167] M. Schuld, A. Bocharov, K. M. Svore, and N. Wiebe, “Circuit-centric quantum classifiers,” *Phys. Rev. A*, vol. 101, p. 032308, Mar 2020.

[168] D. Wecker, M. B. Hastings, and M. Troyer, “Progress towards practical quantum variational algorithms,” *Phys. Rev. A*, vol. 92, p. 042303, Oct 2015.

[169] E. Farhi, J. Goldstone, and S. Gutmann, “A quantum approximate optimization algorithm,” arXiv:1411.4028 [quant-ph], 2014.

[170] A. Peruzzo, J. McClean, P. Shadbolt, M.-H. Yung, X.-Q. Zhou, P. J. Love, A. Aspuru-Guzik, and J. L. O’Brien, “A variational eigenvalue solver on a photonic quantum processor,” *Nature Communications*, vol. 5, no. 1, p. 4213, 2014.

[171] D. Aharonov, A. Ambainis, J. Kempe, and U. Vazirani, “Quantum walks on graphs,” in *Proceedings of the Thirty-Third Annual ACM Symposium on Theory of Computing*, STOC ’01, (New York, NY, USA), pp. 50–59, Association for Computing Machinery, 2001.

[172] T. Tansuwannont, S. Limkumnerd, S. Suwanna, and P. Kalasuwan, “Quantum phase estimation algorithm for finding polynomial roots,” *Open Physics*, vol. 17, no. 1, pp. 839–849, 2019.

[173] J. Kempe, “Quantum random walks: An introductory overview,” *Contemporary Physics*, vol. 44, pp. 307–327, Jul 2003.

[174] D. Reitzner, D. Nagaj, and V. Bužek, “Quantum walks,” *Acta Physica Slovaca Reviews and Tutorials*, vol. 61, no. 6, pp. 603–725, 2011.

[175] S. Lloyd, “Universal quantum simulators,” *Science*, vol. 273, no. 5278, pp. 1073–1078, 1996.

[176] A. M. Childs and R. Kothari, “Simulating sparse hamiltonians with star decompositions,” in *Theory of Quantum Computation, Communication, and Cryptography* (W. van Dam, V. M. Kendon, and S. Severini, eds.), (Berlin, Heidelberg), pp. 94–103, Springer Berlin Heidelberg, 2011.

[177] M. Suzuki, “General nonsymmetric higher-order decomposition of exponential operators and symplectic integrators,” *Journal of the Physical Society of Japan*, vol. 61, no. 9, pp. 3015–3019, 1992.

[178] D. Aharonov and A. Ta-Shma, “Adiabatic quantum state generation and statistical zero knowledge,” in *Proceedings of the Thirty-Fifth Annual ACM Symposium on Theory of Computing*, STOC ’03, (New York, NY, USA), pp. 20–29, Association for Computing Machinery, 2003.

[179] S. S. Hegde, K. R. K. Rao, and T. S. Mahesh, “Pauli decomposition over commuting subsets: Applications in gate synthesis, state preparation, and quantum simulations,” 2016.

[180] A. Y. Kitaev, “Quantum measurements and the abelian stabilizer problem,” arXiv:9511026 [quant-ph], 1995.

[181] H. Mohammadbagherpoor, Y. Oh, P. Dreher, A. Singh, X. Yu, and A. J. Rindos, “An improved implementation approach for quantum phase estimation on quantum computers,” in *2019 IEEE International Conference on Rebooting Computing (ICRC)*, pp. 1–9, 2019.

[182] A. W. Harrow, A. Hassidim, and S. Lloyd, “Quantum algorithm for linear systems of equations,” *Phys. Rev. Lett.*, vol. 103, p. 150502, Oct 2009.

[183] G. Brassard, P. Hoyer, M. Mosca, and A. Tapp, “Quantum amplitude amplification and estimation,” *Quantum Computation and Information*, pp. 53–74, 2002.

[184] H. Kang, “Quantum phase estimation: Circuit implementation in qiskit.” https://qiskit.org/textbook/ch-algorithms/quantum-phase-estimation.html, Last accessed April 2021.

[185] M. A. Nielsen and I. L. Chuang, *The quantum Fourier transform and its applications*, pp. 216–247. Cambridge University Press, 2010.

[186] P. Strömberg and V. Blomkvist-Karlsson, *4-qubit Grover’s algorithm implemented for the IBMQx5 architecture.* PhD thesis, 2018.

[187] M. Lewis, S. Soudjani, and P. Zuliani, “Formal verification of quantum programs: Theory, tools and challenges,” 2021.

[188] D. Gil, “The Future is Quantum,” in *Extreme Ultraviolet (EUV) Lithography X* (K. A. Goldberg, ed.), vol. 10957, International Society for Optics and Photonics, SPIE, 2019.

[189] E. P. DeBenedictis, “A future with quantum machine learning,” *Computer*, vol. 51, no. 2, pp. 68–71, 2018.

[190] A. Steane and E. Rieffel, “Beyond bits: the future of quantum information processing,” *Computer*, vol. 33, no. 1, pp. 38–45, 2000.

[191] N. G. van Kampen, *Stochastic Processes in Physics and Chemistry.* North-Holland Personal Library, Amsterdam: Elsevier, third edition ed., 2007.

[192] O. Mülken and A. Blumen, “Continuous-time quantum walks: Models for coherent transport on complex networks,” *Physics Reports*, vol. 502, p. 37–87, May 2011.

Appendix A

Laplacians and Bessel Functions

A.1 The Bessel Functions

The Bessel functions are canonical solutions $y(x)$ of Bessel's differential equation

$$x^2\frac{\mathrm{d}^2y}{\mathrm{d}x^2}+x\frac{\mathrm{d}y}{\mathrm{d}x}+\left(x^2-\alpha^2\right)y=0$$

for an arbitrary complex number α, which is the order of the Bessel function. Here, we will only present the Bessel functions that are relevant to us, i.e the Bessel functions of the first kind, J_α, and the modified Bessel functions of the first kind, I_α.

Bessel functions of the first kind. Bessel functions of the first kind, denoted as $J_\alpha(x)$, are solutions of Bessel's differential equation that are finite at the origin ($x=0$) for integer or positive α and diverge as x approaches zero for negative non-integer α. It is possible to define the function by its series expansion around $x=0$, which can be found by applying the Frobenius method to Bessel's equation as

$$J_\alpha(x)=\sum_{m=0}^{\infty}\frac{(-1)^m}{m!\Gamma(m+\alpha+1)}\left(\frac{x}{2}\right)^{2m+\alpha},$$

where $\Gamma(z)$ is the gamma function, a shifted generalisation of the factorial function to non-integer values.

The Bessel function of the first kind is an entire function if α is an integer, otherwise it is a multivalued function with singularity at zero. For non-integer α, the functions $J_\alpha(x)$ and $J_{-\alpha}(x)$ are linearly independent, and are therefore the two solutions of the differential equation. On the other hand, for integer order α, the following relationship is valid (the gamma function has simple poles at each of the non-positive integers)

$$J_{-n}(x)=(-1)^nJ_n(x).$$

This means that the two solutions are no longer linearly independent. In this case, the second linearly independent solution is then found to be the Bessel function of the second kind, which we will not analyse here.

Another definition of the Bessel function, for integer values of n, is possible using an integral representation as

$$J_n(x) = \frac{1}{\pi}\int_0^{\pi} \cos(n\tau - x\sin\tau)d\tau$$

or alternatively

$$J_n(x) = \frac{1}{2\pi}\int_{-\pi}^{\pi} e^{\mathrm{i}(x\sin\tau - n\tau)}d\tau.$$

Modified Bessel functions of the first kind. The Bessel functions are valid even for complex arguments x, and an important special case is that of a purely imaginary argument. In this case, the solutions to the Bessel equation are called the modified Bessel functions (or occasionally the hyperbolic Bessel functions) of the first (and second kind, which we will not delve into here) and are defined as

$$I_\alpha(x) = \mathrm{i}^{-\alpha}J_\alpha(ix) = \sum_{m=0}^{\infty}\frac{1}{m!\Gamma(m+\alpha+1)}\left(\frac{x}{2}\right)^{2m+\alpha}.$$

This is used when α is not an integer; when it is an integer, then the limit is used. These are chosen to be real-valued for real and positive arguments x. The series expansion for $I_\alpha(x)$ is thus similar to that for $J_\alpha(x)$, but without the alternating $(-1)^m$ factor.

Additionally, we can express the first kind Bessel functions in terms of the modified Bessel functions of the first kind as

$$J_\alpha(\mathrm{i}z) = e^{\frac{\alpha \mathrm{i}\pi}{2}}I_\alpha(z).$$

A.2 The Laplace Transform

In mathematics and signal processing, the Laplace transform is an integral transform that converts a function of a real variable t (often time) to a function of a complex variable s (complex frequency). The Laplace transform of a time-dependent function $P(t)$, denoted as $\hat{P}(s) = \mathcal{L}\{P(t)\}$, is defined as

$$\mathcal{L}\{P(t)\} = \int_0^{\infty} e^{-st}P(t)\mathrm{d}t.$$

The basic properties of the Laplace transform that are worth mentioning here can be revised as follows.

- Linearity: $\mathcal{L}\{aP(t) + bQ(t)\} = a\hat{P}(s) + b\hat{Q}(s)$
- Derivative: $\mathcal{L}\{P'(t)\} = s\hat{P}(s) - P(0)$
- Shifting: $\mathcal{L}\{e^{at}P(t)\} = \hat{P}(s-a)$

The relevant inverse Laplace transform involving the Bessel functions are (for $\nu > -1$)

$$\hat{P}(s) = \frac{(s - \sqrt{s^2 - a^2})^\nu}{\sqrt{s^2 - a^2}} \iff P(t) = a^\nu I_\nu(at)$$

$$\hat{P}(s) = \frac{(\sqrt{s^2 + a^2} - s)^\nu}{\sqrt{s^2 + a^2}} \iff P(t) = a^\nu J_\nu(at)$$

Appendix B

Analysis of the Continuous-time Quantum Walk Hamiltonian

In order to better understand and analyse the Hamiltonian of the continuous-time quantum walk, we need to start by the classical case. In the continuous-time random walk (CTRW), the transition probability from node y to node x in time t can be denoted by $p_{x,y}(t)$. Therefore, the initial condition is $\langle x|y\rangle = p_{x,y}(0) = \delta_{x,y}$, where the Kronecker δ can be defined as $\delta_{x,y} = \begin{cases} 1 & x = y \\ 0 & \text{otherwise} \end{cases}$.

The transition rates per unit time are the elements of a transfer matrix T, where $T_{x,y} = \langle x|T|y\rangle$. Assuming a Markovian process, the following equation can be shown to hold [191]

$$\frac{\mathrm{d}}{\mathrm{d}t} p_{x,y}(t) = \sum_l T_{x,l} p_{l,y}(t). \tag{B.1}$$

This equation essentially defines continuous-time random walks in the classical setting.

If we let the transition rates γ between all nodes to be equal, then we can define the transfer matrix through the adjacency matrix A as $T = -\gamma A$. The formal solution of equation (B.1) is $p_{x,y}(t) = \langle x|e^{Tt}|y\rangle = \langle x|e^{-\gamma A t}|y\rangle$. Denoting the eigenstates of the adjacency matrix A by e_n, we can get the solution

$$p_{x,y}(t) = \sum_n e^{-\lambda_n \gamma t} \langle x|e_n\rangle \langle e_n|y\rangle , \tag{B.2}$$

where λ_n are the eigenvalues of the adjacency matrix, corresponding to eigenvectors e_n.

Since the eigenvalues are positive ($\lambda_n > 0$ for $n > 1$ and $\lambda_1 = 0$), the long-time limit follows directly. For $t >> 1$ in the sum of equation (B.2) all exponential terms but one decay rapidly to zero, with the one for $\lambda_1 = 0$ being the only one surviving with corresponding eigenstate $|e_1\rangle = \frac{1}{N}\sum_l |l\rangle$. Therefore, the long-time limit of all transition probabilities is $\lim_{t\to\infty} p_{x,y}(t) = 1/N$. This means that every CTRW whose transfer matrix follows directly from the adjacency matrix will eventually decay at long times to the equipartition value $1/N$ [192].

We can now move on to the continuous-time quantum walk and the quantum mechanical dynamics. Assume that the states $|y\rangle$ span the Hilbert space $\mathcal{H}^S$ and they are orthonormal and complete, i.e $\langle x|y\rangle = \delta_{x,y}$, with $\sum |y\rangle\langle y| = \mathbb{1}$. The dynamics then are governed by a specific Hamiltonian H, analogous to the theory mentioned in the previous section, such that Scrhödinger's equation for the amplitudes reads

$$\mathrm{i}\frac{\mathrm{d}}{\mathrm{d}t}\psi_{x,y}(t) = \sum_y H_{x,y}\psi_{x,y}(t) \iff \mathrm{i}\frac{\mathrm{d}}{\mathrm{d}t}\langle x|\psi(t)\rangle = \sum_y \langle x|H|y\rangle \langle y|\psi(t)\rangle\,, \tag{B.3}$$

where $\psi_{x,y}(t)$ being the amplitude of the quantum state $|\psi_{x,y}(t)\rangle$ at time t and the equivalency standing for obvious reasons. The transition probabilities are now $\pi_{x,y}(t) = |\psi_{x,y}(t)|^2$.

We identify that $H = -T = \gamma A$. Let the eigenvalues of H be ℓ_n and the corresponding eigenvectors f_n. The quantum mechanical transition probabilities can now be defined as

$$\pi_{x,y}(t) = |\sum_n e^{-\mathrm{i}\ell_n t}\langle x|f_n\rangle\langle f_n|y\rangle|^2$$

Unlike the classical case, in the quantum case there is no unique long-time limit of $\pi_{x,y}(t)$ due to the unitary nature of the quantum evolution. In order to compare the long-time evolution of the CTRW we can use the long-time average [25]

$$\begin{aligned}\chi_{x,y} &\equiv \lim_{T\to\infty}\frac{1}{T}\int_0^T \pi_{x,y}(t)\mathrm{d}t\\ &= \sum_{m,n}\delta_{\ell_n,\ell_m}\langle x|f_n\rangle\langle f_n|y\rangle\langle y|f_m\rangle\langle f_m|x\rangle\,,\end{aligned}$$

where δ_{ℓ_n,ℓ_m} follows the rules for Kronecker δ. The long-time average $\chi_{x,y}$ still depends on the initial and final nodes x and y.

Appendix C

Generalised Inverter Gate for Arbitrary Control Qubits

In many circumstances, we need to control an inversion with an arbitrary number of $n_c > 2$ control qubits. A solution can be given by introducing intermediate computations, with their results stored in an ancilla register of size $n_c - 1$ [30]. A visualisation of this solution for a generalised `CNOT` gate with n_c control qubits is shown in Figure C.1. This decomposition of the generalised `CNOT` gate can be further simplified to use just regular `CNOT` operations.

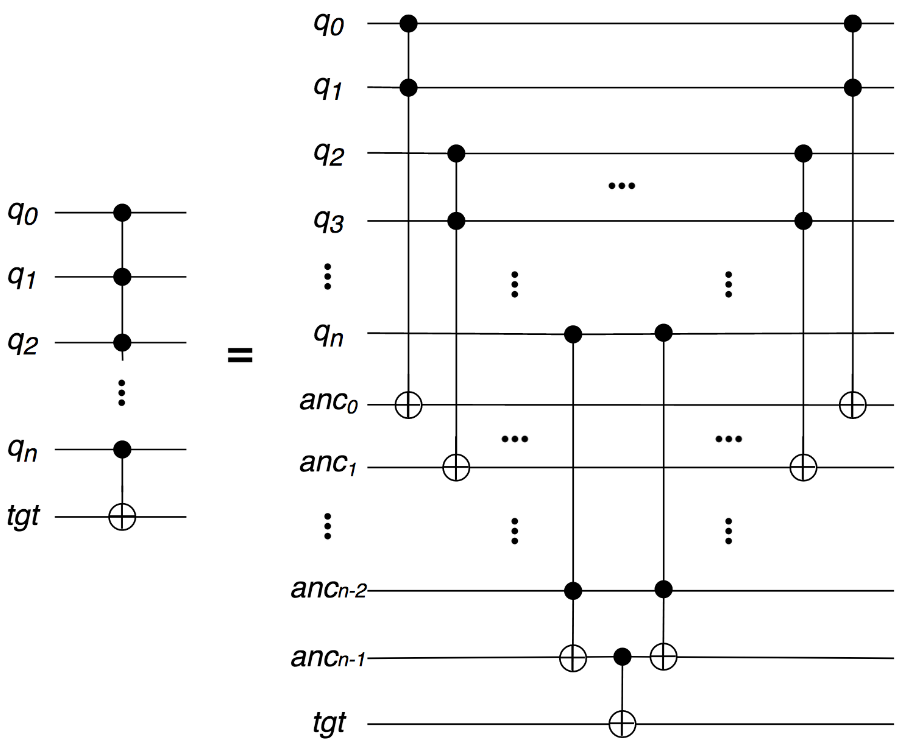


Fig. C.1 Generalised Toffoli gate with n control qubits (q_0 to q_n), $n - 1$ ancilla qubits (anc_0 to anc_{n-1}) and one target qubit (tgt).

Appendix D

Method of Stationary Phase

D.1 Method

The method of mathematical analysis called *method of stationary phase* can be used in mathematical physics to estimate the integrals derived in equation (3.12). Consider integrals of type

$$I(t) = \int_a^b g(k) \mathrm{e}^{\mathrm{i}t\phi(k)}\, \mathrm{d}k$$

where $t > 0$ is an arbitrary parameter, $g(k)$ is a function (otherwise called *amplitude* in mathematical physics) and $\phi(k)$ is a real-valued function (i.e., *phase* in mathematical physics). The endpoints of integration, i.e., places where the derivatives of $g(k)$ fail to be continuous and places where the derivatives of $\phi(k)$ vanish, are called critical points. An intuitive critical point could be, for example, $k = a$, where $\phi'(a) = 0$ but $\phi''(a) \neq 0$. Such a critical point is also called a stationary point as it represents a point where the phase function has a minimum or a maximum and is, thus, stationary. The interest here lies on the behaviour of the integral at points where the oscillations of the phase are small, i.e., near the stationary points of the function $\phi(k)$.

There are three possibilities that need to be considered, one where the stationary point is on the lower endpoint of integration, i.e., $k = a$, one at the higher, $k = b$, and one on an arbitrary interior point, $k = c$. The stationary phase formula can be written as

$$I_x(t) \sim g(x) e^{\mathrm{i}t\phi(x) + \mathrm{i}\,\mathrm{sgn}(\phi''(x))\pi/4} \sqrt{\frac{2\pi}{t|\phi''(x)|}}, \tag{D.1}$$

where $x \in \{a, b, c\}$ are the various stationary points, as described above.

D.2 Approximation of Hadamard Walk Evolution

The integrals from equation (3.12) can be transformed to suit equation (D.1), first by setting $n = \lambda t$, when these integrals obtain the form

$$I(t; \lambda) = \frac{1}{2\pi} \int_{-\pi}^{\pi} g(k) e^{\mathrm{i}\phi(k;\lambda)} t \mathrm{d}k \tag{D.2}$$

where $\phi(k;\lambda) = k\lambda - \omega_k$ and $g(k)$ is either an even or an odd function. In this slightly generalised case there are no stationary points for $\lambda > 1/\sqrt{2}$ or $\lambda < -1/\sqrt{2}$ and $I(t;\lambda)$ by getting with λ further from zero decreases exponentially fast. For $\lambda \in (-1/\sqrt{2}, 1/\sqrt{2})$ the stationary points $\pm k_\lambda \in [0;\pi]$ are found, where

$$\cos k_\lambda = \frac{\lambda}{\sqrt{1-\lambda^2}}$$

and

$$\frac{\partial^2\phi}{\partial k^2}\left(\pm k_\lambda;\lambda\right) = \pm\left(1-\lambda^2\right)\sqrt{1-2\lambda^2} \qquad \left(= -\omega''_{k_\lambda}\right)$$

Under these conditions and by dividing the integration range $[-\pi;\pi]$ in equation (D.2) into four subintervals by points 0 and $\pm k_\lambda$ follows

$$I(t;\lambda) = \frac{2g\left(k_\lambda\right)}{\sqrt{2\pi t\left|\omega''_{k_\lambda}\right|}}\begin{cases} \cos\left[t\phi\left(k_\lambda;\lambda\right) + \frac{\pi}{4}\right], & \text{for } g \text{ even} \\ \mathrm{i}\sin\left[t\phi\left(k_\lambda;\lambda\right) + \frac{\pi}{4}\right], & \text{for } g \text{ odd} \end{cases}$$

Finally, for equations (3.12) the following are obtained

$$\alpha^t(\lambda t) \sim \frac{2}{\sqrt{2\pi t\left|\omega''_{k_\lambda}\right|}}\cos\left[t\phi\left(k_\lambda;\lambda\right) + \frac{\pi}{4}\right],$$

$$\beta^t(\lambda t) \sim \frac{2\lambda}{\sqrt{2\pi t\left|\omega''_{k_\lambda}\right|}}\cos\left[t\phi\left(k_\lambda;\lambda\right) + \frac{\pi}{4}\right],$$

$$\gamma^t(\lambda t) \sim -\frac{2\sqrt{1-2\lambda^2}}{\sqrt{2\pi t\left|\omega''_{k_\lambda}\right|}}\sin\left[t\phi\left(k_\lambda;\lambda\right) + \frac{\pi}{4}\right].$$

Now, the probability of being in position $n = \lambda t$ after t steps is

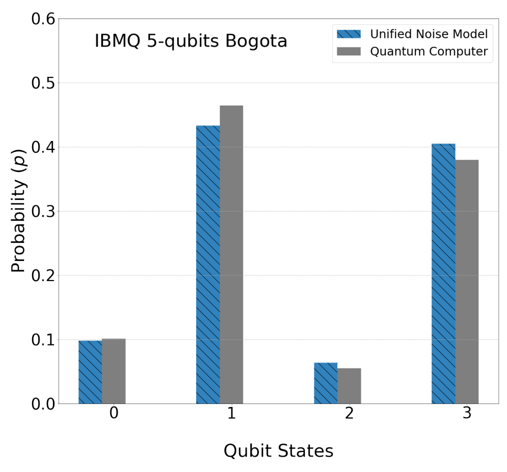
$$P^t(\lambda t) \sim \frac{2(1+\lambda)}{\pi t\left(1-\lambda^2\right)\sqrt{1-2\lambda^2}}[1 + \lambda\sqrt{2}\cos\theta]$$

where $\theta = 2t\phi\left(k_\lambda;\lambda\right) + \frac{\pi}{2} + \mu$ and $\tan\mu = \frac{\sqrt{1-2\lambda^2}}{1+2\lambda}$.

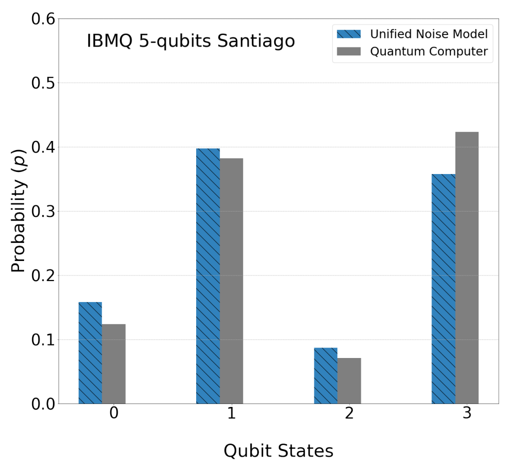
Appendix E

Comparison of Probability Distributions of the Benchmark Experiments

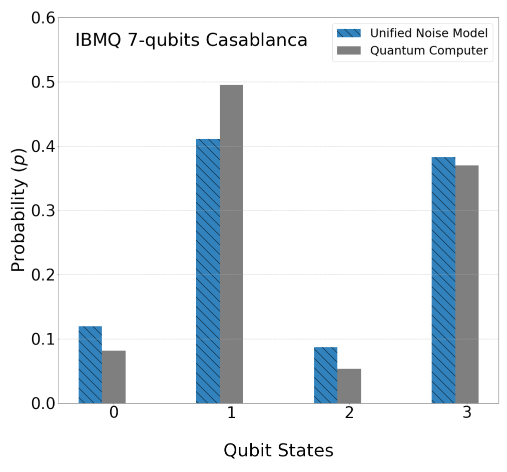
Furthermore, in the context of the additional experiments carried out for the benchmark methodology in Chapter 6, this section presents a series of figures that showcase the difference between the probability distributions of each quantum computer (i.e., the IBMQx5 Bogota, IBMQx5 Santiago and IBMQx7 Casablanca machines) and the unified noise model (UNM). Each Figure includes a set of plots that represent each algorithm used for benchmarking, i.e., for the discrete-time quantum walk (Figure E.1), the continuous-time quantum walk (Figure E.2), the Pauli decomposition of the continuous-time quantum walk Hamiltonian (Figure E.3), the quantum phase estimation algorithm (Figure E.4) and the quantum search algorithm (Figure E.5). Finally, each figure contains a plot of the probability distributions resulting from the UNM simulations and the ideal evolution for each quantum algorithm, thus including every meaningful comparison that derives from the proposed benchmarks in Chapter 6.



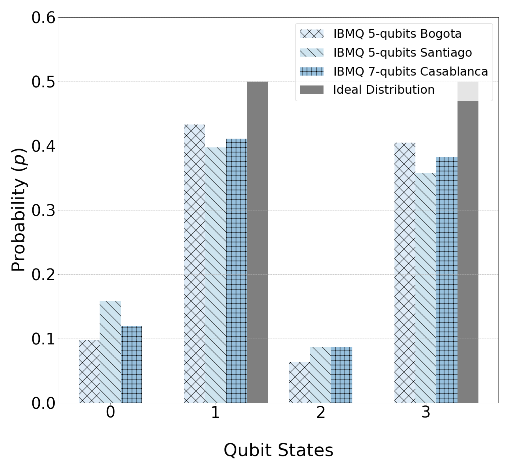
(a)



(b)

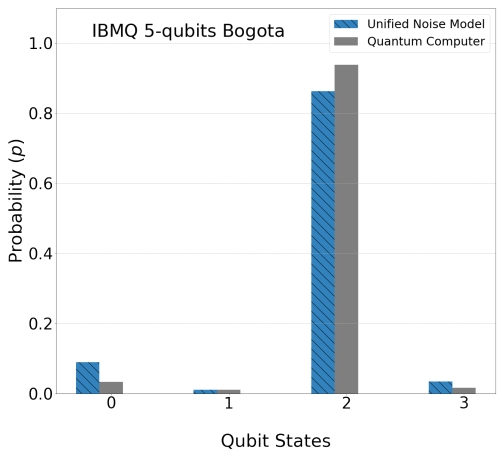


(c)

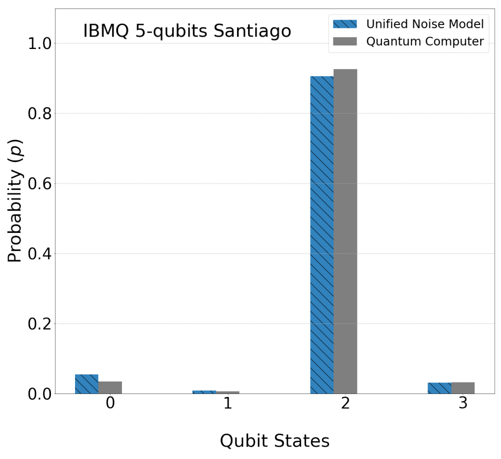


(d)

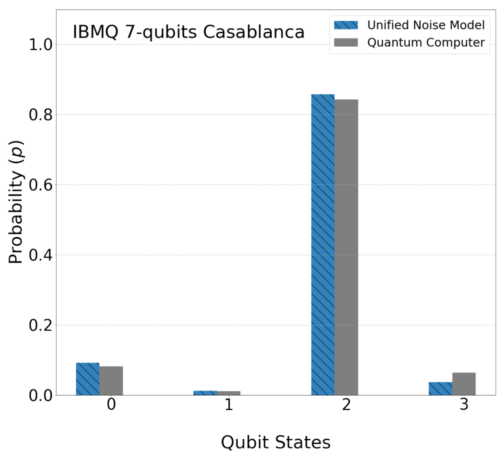
Fig. E.1 Comparison between the probability distributions of the discrete-time quantum walk execution on each quantum machine: (a) the IBMQ 5-qubit Bogota, (b) IBMQ 5-qubit Santiago and (c) IBMQ 7-qubit Casablanca machines; (d) comparison between the UNM simulations for each machine and the ideal distribution.



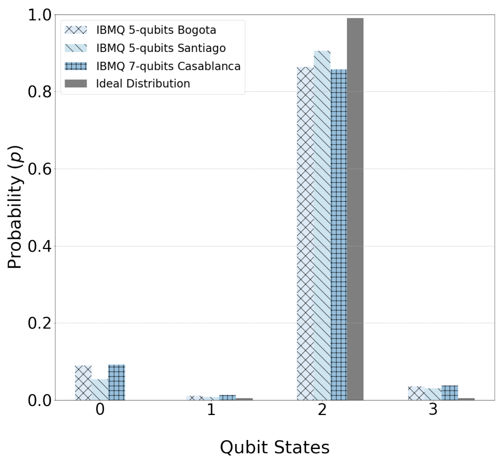
(a)



(b)

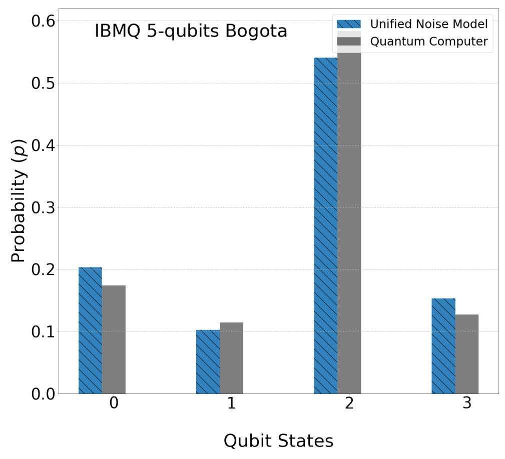


(c)

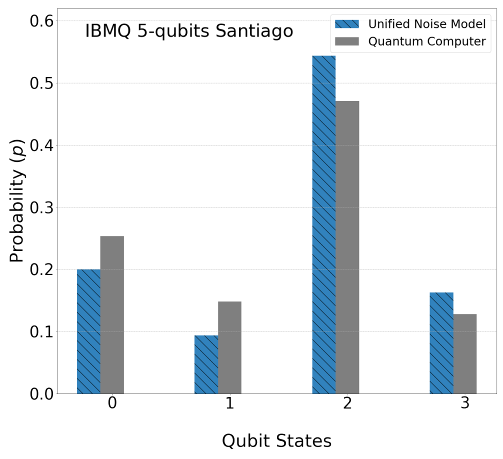


(d)

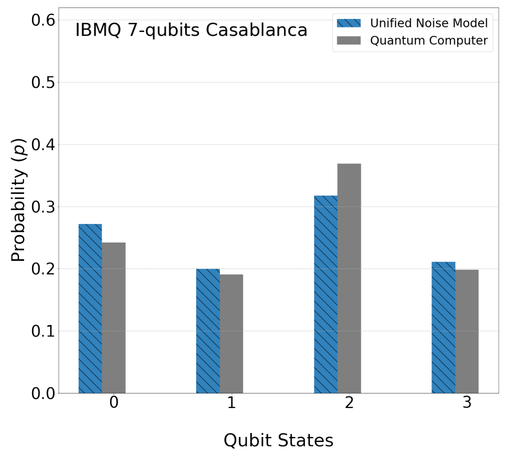
Fig. E.2 Comparison between the probability distributions of the continuous-time quantum walk execution on each quantum machine: (a) the IBMQ 5-qubit Bogota, (b) IBMQ 5-qubit Santiago and (c) IBMQ 7-qubit Casablanca machines; (d) comparison between the UNM simulations for each machine and the ideal distribution.



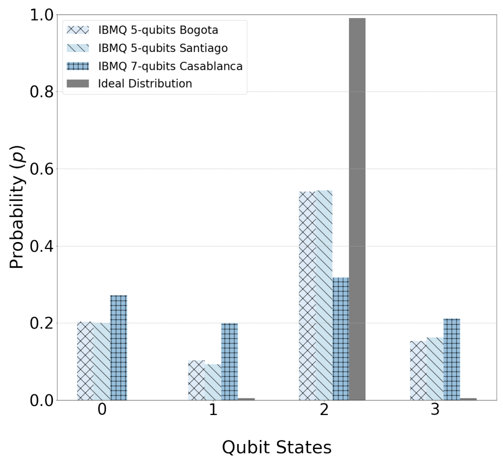
(a)



(b)

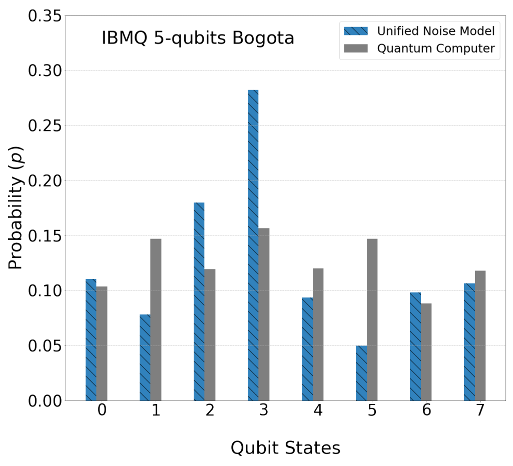


(c)

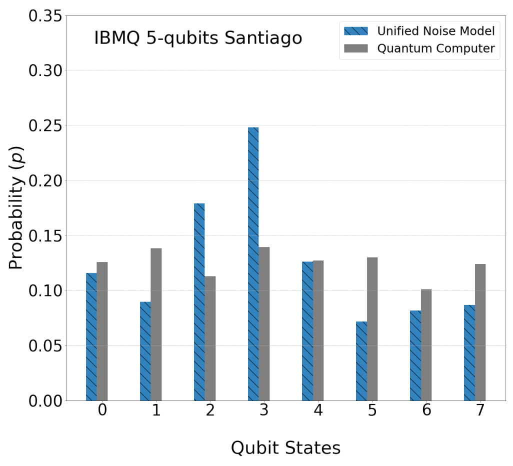


(d)

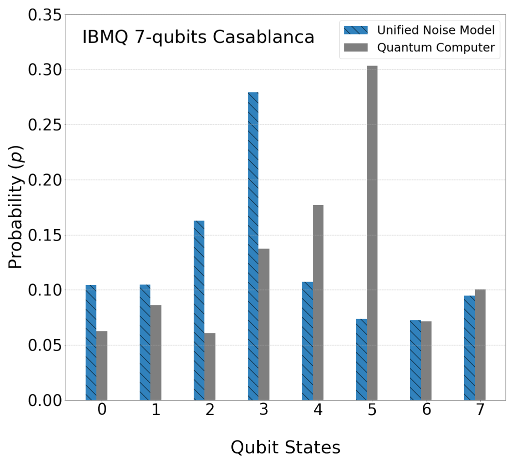
Fig. E.3 Comparison between the probability distributions of the Pauli decomposition of the continuous-time quantum walk execution on each quantum machine: (a) the IBMQ 5-qubit Bogota, (b) IBMQ 5-qubit Santiago and (c) IBMQ 7-qubit Casablanca machines; (d) comparison between the UNM simulations for each machine and the ideal distribution.



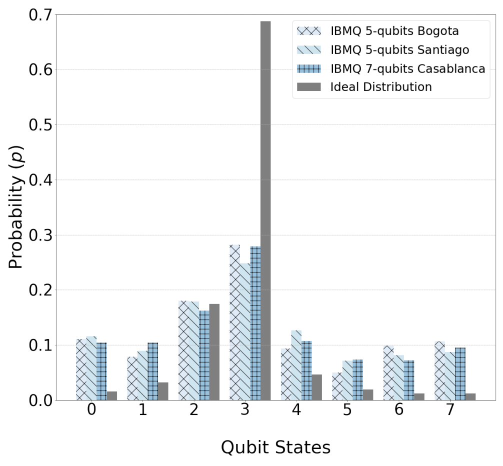
(a)



(b)

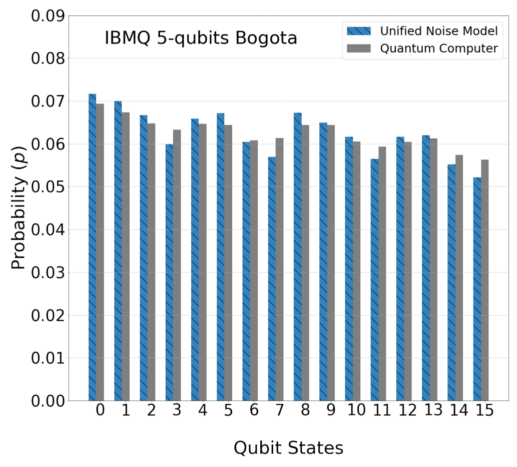


(c)

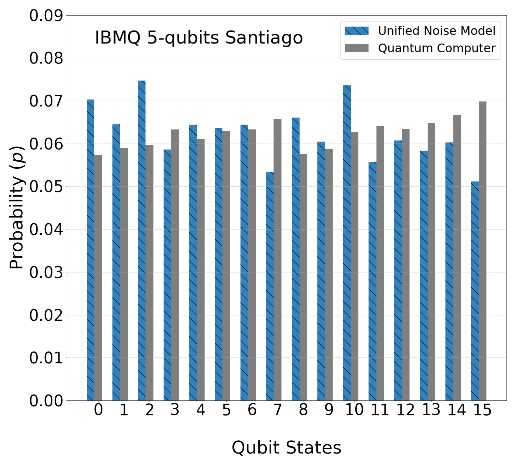


(d)

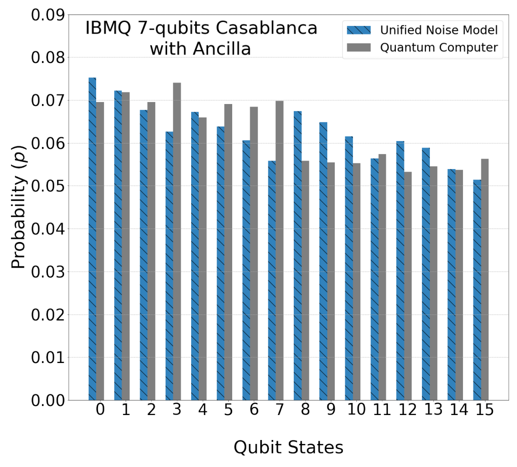
Fig. E.4 Comparison between the probability distributions of the quantum phase estimation execution on each quantum machine: (a) the IBMQ 5-qubit Bogota, (b) IBMQ 5-qubit Santiago and (c) IBMQ 7-qubit Casablanca machines; (d) comparison between the UNM simulations for each machine and the ideal distribution.



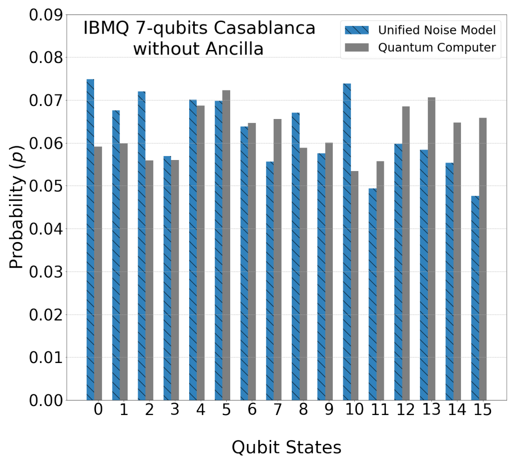
(a)



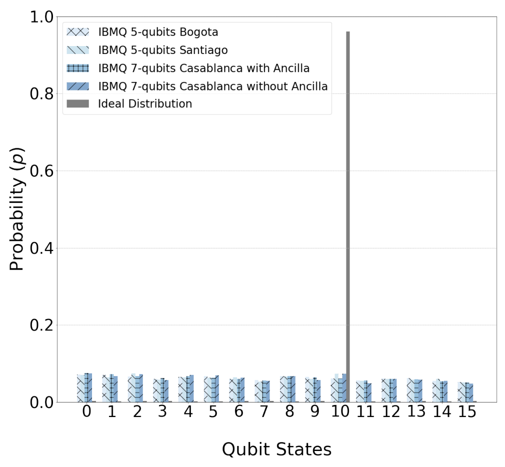
(b)



(c)



(d)



(e)

Fig. E.5 Comparison between the probability distributions of the quantum search execution on each quantum machine: (a) the IBMQ 5-qubit Bogota, (b) IBMQ 5-qubit Santiago and (c) IBMQ 7-qubit Casablanca machines; (d) comparison between the UNM simulations for each machine and the ideal distribution.