UNIVERSITY OF COPENHAGEN
FACULTY OF SCIENCE

NIELS BOHR INSTITUTE

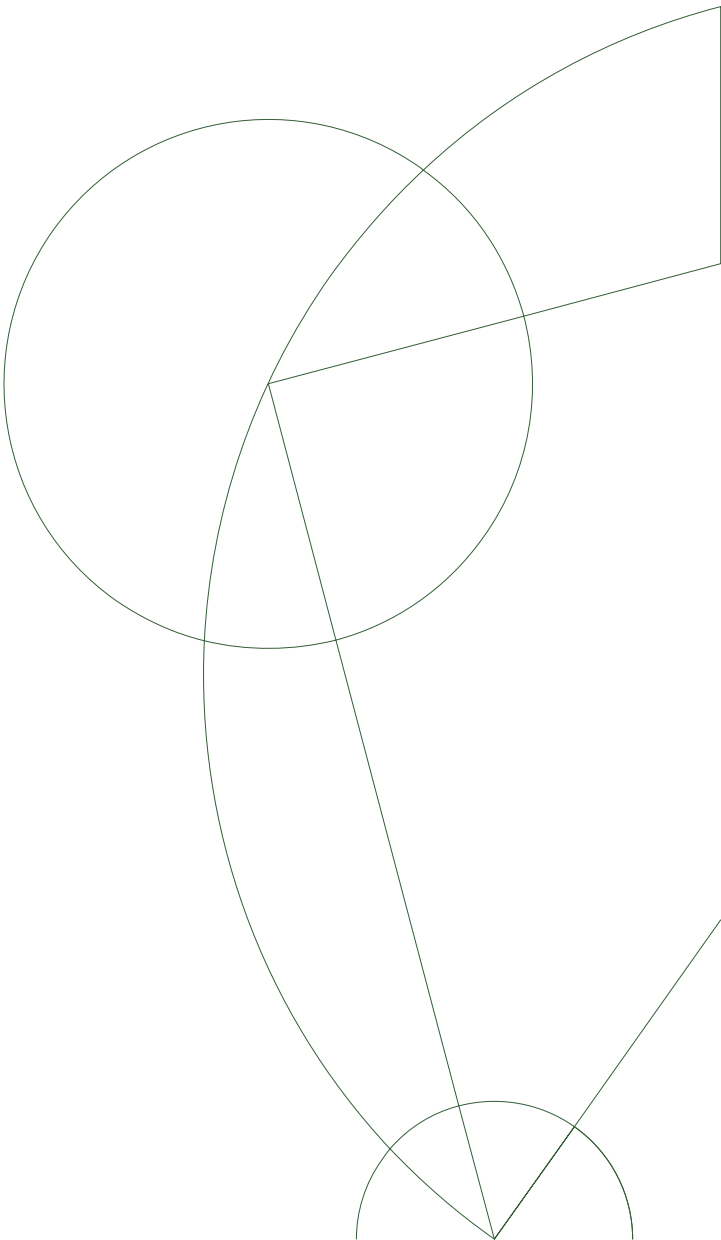
Master's Thesis

Efficient Simulation of Quantum Trajectories for Multiple Qubits in a Resonator Cavity Under Continuous Heterodyne Detection

Hy-Q - Center for Hybrid Quantum Networks

Anas Idiab <hjr420@alumni.ku.dk>
Sune Forchhammer <dkz419@alumni.ku.dk>

Supervisor
Klaus Mølmer <klaus.molmer@nbi.ku.dk>



Abstract

The area of quantum information has the potential to revolutionize many fields, such as computing, communication, material and chemical research, and fundamental science. Building a reliable quantum computer is faced with many challenges, due to the intrinsic quantum nature of the hardware. Due to this, it is useful to construct simulators where the influence of the environment is a completely controlled parameter. A model that is able to quickly predict the feasibility of a system configuration would accelerate technological development. It is therefore the aim of this thesis, to present a powerful analytical simulator tool, to gain insight into the workings of a superconducting quantum processor, supplementary to the information directly available in experimentation. The focus of this, will be on the readout process, and the physics of heterodyne measurements in the dispersive regime. It presents simulations, discussing findings related to dispersive shift, the Zeno effect, and trajectory inference from measurement records.

Acknowledgements

We would like to thank our supervisor, Klaus Mølmer, for the many discussions, and ideas for the directions we could take this project. Without Klaus, this thesis would not be possible, and discussions with him have given us new insights into quantum measurement theory, as well as the world of quantum mechanics in general.

We would also like to thank Morten Kjaergaard for guiding us to sources, and practical information. We also thank both Morten Kjaergaard and Jacob Hastrup for inviting us to their laboratory, to let us have a tour, and explaining to us how their system operates.

Contents

1 Introduction

Classical computers have proven immensely useful for expanding computational capabilities, thereby granting access to larger solution spaces and previously unobtainable answers due to the limited computational speed. However, the capabilities of classical computers are inherently limited by a fundamental constraint. Ultimately, each processing unit in classical computers, execute instructions in a serial fashion. Because of this, algorithms for which there are no known polynomial time scaling solutions, rapidly expand in computational complexity to a point where no classical processing can reach a solution within an acceptable time frame. Quantum computers demonstrate a potential to accelerate certain computationally hard tasks, and extend the range of solutions which is reachable within reasonable time.

Quantum information has therefore become an active research area, with the aim of engineering better quantum processors, and one day, hopefully, exhibit quantum supremacy. In this pursuit, the field has advanced from exploring the behavior of isolated single qubit systems, to the development of multi-qubit processors[1]. As these systems expand, the simulation of the entangled behavior of their states on a classical processor will eventually become unfeasible. However, as this thesis demonstrates, it is still possible to extract useful information about a reasonably sized multi-qubit system, within a sensible computational time-frame, if proper considerations have been made to the performance and optimization of such a simulation.

Research is made into different hardware platforms for the implementation of the qubits. A promising candidate for the construction of a scalable quantum processor, is one where the qubit information is stored on anharmonic oscillators within a superconducting circuit. Hence, this thesis investigates the architecture of such a circuit and leverages it to construct a scalable simulator. This simulator is proficient in simulating qubits as well as the effects of measuring them.

The measurement of the system's state is accomplished through continuous measurement, which doesn't immediately project the state onto the computational basis, but gradually feeds the information to the observer. This scheme is described by a stochastic master equation (SME)[2], where the measurement back action, allows for multiple possible outcomes of the state evolution, described by a quantum trajectory. Due to quantum measurements' inherent randomness, these trajectories signify the probabilistic, rather than deterministic, evolution of the system's state. The evolution of each quantum trajectory, is embarrassingly parallel, which allows for complete utilization of the cores, and boosts the performance of simulating additional trajectories linearly with the number of cores available. In reality, it is not completely linear, because of data reduction nuances, which are described in the implementation section of this thesis.

2 Description of the system

The realization of quantum computation hinges on the fulfillment of several fundamental prerequisites that a quantum computer needs in order to effectively assist in solving complex problems that exceed the computational capacity of classical computers. The physicist David P. DiVincenzo has identified five essential criteria that a quantum computer must satisfy in order to achieve quantum supremacy[3]. DiVincenzo's ideas can be concisely encapsulated as follows:

1. **Scalable system, with well-characterized qubits**
   A fundamental requirement that arises from the encoding of information on a quantum computer is that the system must manifest the properties of a two-level quantum system, such as the spin of an electron. The realization of this objective is not contingent on the exclusive use of spinors to create qubits, but rather on the existence of a vector subspace in which the system exhibits spinor-like characteristics that are stable and accessible at a reasonable cost. Additionally, scalability is a crucial consideration in the development of quantum computing technologies, as classical computers excel at performing computations on a large number of bits. Furthermore, achieving quantum supremacy necessitates the consideration of scalability, as classical computers are capable of performing fast and dependable bit operations on a vast number of bits. While certain quantum algorithms have the potential to scale exponentially in terms of computation speed, surpassing the computational performance of classical computers demands the deployment of a sufficiently large number of qubits. However, the cost associated with scaling the number of qubits represents a formidable challenge due to the inherent quantum nature of the problem.

2. **Ability to initialize its qubits**
   The ability to exercise control over input values constitutes a fundamental tenet of computing, whether in the context of classical or quantum computing. To achieve this objective, it is necessary to initialize the registers of the computing system to a known state, which can subsequently be transformed into the desired state via the application of unitary gate operations. This requirement also plays a crucial role in quantum error-correction, which will be expanded upon in subsequent sections of this thesis.

3. **Long coherence time**
   A fundamental requirement for quantum computation is the preservation of long coherence times for the qubits. Specifically, the quantum state must retain its information for a duration exceeding the time required to execute calculations and perform readouts. This requires the

coherence time to exceed the duration between initialization and measurement. Decoherence of qubits typically results from dephasing or decay and is primarily attributable to the environment. To mitigate the impact of environmental noise, it is necessary to effectively isolate the qubit from its surroundings. However, this requirement poses a significant challenge to scalability, as quantum computation relies on the entanglement of qubits, necessitating the engineering of interactions between them. Moreover, the greater the number of entangled qubits, the more challenging it becomes to protect them from the environment, thereby exacerbating the issue of decoherence.

4. **Universal quantum gates**
   Turing completeness can be achieved by a classical computer through the use of a small set of logic gates, i.e. AND, OR, and NOT gates, which form a universal set of gates for classical computation. According to the Church-Turing thesis, this allows classical computers to compute any classical algorithm. While quantum computers expand the ways in which algorithms can be solved, they do not alter the concept of Turing undecidability, which posits that some problems lack algorithms that can provide answers within a finite amount of time. Quantum computing does change some aspects of complexity and how certain problems scale with input values. Similar to how a Turing-complete computer requires a universal set of gates, any universal quantum computer can be represented by a combination of a proper two-qubit gate and a set of single-qubit gates. The choice of specific gates is dependent on the selected qubit as well as the hardware employed. Furthermore, the set of gates utilized may be subject to change as further research into their characteristics may reveal additional benefits of alternative gate schemes. A more detailed analysis of this topic will be presented in section 4.1.

5. **Efficient measurement**
   Upon completion of the computation, it is necessary to read out the result to the registers. The ability to measure the state of each qubit in a reliable manner within a short time frame is essential to avoid decoherence and achieve high-fidelity results, i.e., to reduce the error in the output. The act of measurement destroys superpositions of state, which is a topic of interest in measurement theory. This thesis aims to examine the characteristics of continuous measurement on a system, and delve deeper into the topic of readout and its effects.

In the pursuit of constructing a qubit, numerous candidates have been identified, encompassing ion traps[4], neutral atoms[5], quantum dots[6], topological qubits[7], photonic qubits[8], and electron spin in crystal lattice[9]. The focus of this thesis, however, is the examination of a superconducting circuit-based model of the transmon qubit.

As emphasized by DiVincenzo's third criterion, the indispensability of isolating the qubit from the surrounding environmental noise cannot be overstated. Additionally, the challenge posed by thermal energy is not to be neglected, as unwanted transitions driven by quanta of energy may contribute to qubit decoherence. For the manifestation of the system's quantum behavior, it is imperative that dissipation be minimized. Consequently, superconducting materials are employed in the fabrication of the circuit, ensuring zero resistance at the designated operating temperature.

2.1 Quantum harmonic oscillator

An approach to understand the properties of the transmon qubit, is to start by examining the properties of the classical LC-circuit, then subsequently employ canonically conjugate quantum operators to transform the Hamiltonian for its quantum counterpart.

In a series-connected circuit with linear capacitance, C, and linear inductance L connected in series, as shown in Fig. 1a, the first step to derive the equations motion for the circuit entails the application of Ohm's law for inductors and capacitors:

$$V_L = L\frac{\mathrm{d}I}{\mathrm{d}t} = \frac{\mathrm{d}^2Q}{\mathrm{d}t^2} \tag{2.1}$$

$$I = C\frac{\mathrm{d}V_C}{\mathrm{d}t} = \frac{\mathrm{d}}{\mathrm{d}t}\left(CV_C\right) = \frac{\mathrm{d}Q}{\mathrm{d}t} \Rightarrow V_C = \frac{Q}{C}. \tag{2.2}$$

Here, V_L and V_C denote the voltage on the inductor and capacitor respectively, I represents the electric current traversing the circuit, and Q signifies the electric charge accumulation on the capacitor. Owing to the linearity of inductance and capacitance, L and C remain constant, and the definitions of their linearity yield $Q = CV$ as well as a magnetic flux, derived from Faraday's law, on the inductor: $\Phi = LI$.

Kirchhoff's voltage and current rules dictate that $V_L + V_C = 0$ and $I_L = I_C = I$. Consequently, through direct substitution of equations (2.1) and (2.2), the following is derived:

$$\ddot{Q}(t) + \frac{1}{LC}Q(t) = 0, . \tag{2.3}$$

This expression is recognized as the equation of motion for a classical harmonic oscillator. The physical interpretation of this equation is understood from the conservation of energy, with the energy stored in the two fields. The energy is either stored in the magnetic field around the inductor as charges move through it and modify the magnetic flux, or it is stored in the magnetic field as charges accumulate on the capacitor. When charging the capacitor, additional work is performed, further contributing to the energy storage.

The equation of motion contains a charge variable, denoted by $Q(t)$, which can be linked to a generalized position coordinate, leading to the interpretation of the inductor as an inertia component of the system. The selection of a position coordinate is entirely arbitrary, and hence for reasons that shall later become apparent, a different choice of generalized position coordinates is introduced by defining the branch flux of a circuit element as:

$$\Phi = \int_{\infty}^{t} V(\tau)\mathrm{d}\tau. \tag{2.4}$$

Equation (2.3) may be reformulated in terms of the branch flux, resulting in $\ddot{\Phi}+\Phi/LC = 0$. Note that, although branch flux represents a more general flux compared to magnetic flux, in the context of a linear inductor, the branch flux and magnetic flux are entirely equivalent. With this choice of coordinates, the circuit's energy oscillates between the electrical energy of the capacitor and the magnetic energy of the inductance. Consequently, the energy of a capacitor and the energy of an inductor can be expressed as kinetic and potential energy, respectively:

$$K_C = \frac{1}{2}CV^2, \tag{2.5}$$

$$U_L = \frac{1}{2L}\Phi^2. \tag{2.6}$$

Owing to equation (2.4), $V = \dot{\Phi}$, which leads to the Lagrangian for the circuit:

$$\mathcal{L} = K_C - U_L = \frac{1}{2}C\dot{\Phi}^2 + \frac{1}{2L}\Phi^2. \tag{2.7}$$

It is important to note that the calculation of the Euler-Lagrange equation returns the appropriate equations of motion. This Lagrangian can be employed to map $(\phi, \dot{\phi}) \rightarrow (p, \phi)$ and derive the Hamiltonian. The conjugate momentum to the flux is computed as follows:

$$p = \frac{\partial \mathcal{L}}{\partial \dot{\phi}} = C\dot{\phi}. \tag{2.8}$$

In this coordinate system, the capacitor's charge, $Q = C\dot{\phi}$, is not unexpectedly associated with the momentum coordinate. The Hamiltonian is then defined by the Legendre transformation:

$$H = p\dot{\phi} - \mathcal{L} = \frac{1}{2}C\dot{\Phi}^2 + \frac{1}{2L}\Phi^2. \tag{2.9}$$

This Hamiltonian is analogous to a mechanical harmonic oscillator, characterized by a mass of $m = C$ and a resonant frequency $\omega_r = 1/\sqrt{LC}$. To quantize the Hamiltonian, momentum and generalized position must be promoted into

operators, $(Q, \Phi) \rightarrow (\hat{Q}, \hat{\Phi})$, ensuring that they adhere to the canonical commutation relations $\left[\hat{\Phi}, \hat{Q}\right] = i\hbar$. Subsequently, the new Hamiltonian can be expressed in terms of the Dirac step operators of a quantum harmonic oscillator (QHO):

$$\hat{H} = \hbar\omega_r \left(\hat{a}^\dagger \hat{a} + \frac{1}{2}\right), \tag{2.10}$$

where $\hat{a}^\dagger$ denotes the creation operator and $\hat{a}$ represents the annihilation operator. These two operators act upon the eigenstates of the QHO, altering their state: $\hat{a}\,|n\rangle = \sqrt{n}\,|n-1\rangle$ and $\hat{a}^\dagger\,|n\rangle = \sqrt{n+1}\,|n+1\rangle$. A very general definition of the annihilation operator is:

$$\hat{a} = \frac{1}{\sqrt{2}}\left(\tilde{q} + i\tilde{p}\right). \tag{2.11}$$

$\tilde{q}$ and $\tilde{p}$, denotes some reduced generalized position and momentum operator respectively. Translated into the case of the LC-circuit, that is the Hamiltonian of Eq. (2.9), these operators take the form:

$$\tilde{q} = \sqrt{\frac{\omega_r C}{\hbar}}\hat{\Phi} = \frac{\phi_{zpf}}{\sqrt{2}}\hat{\phi}, \tag{2.12}$$

$$\tilde{p} = \frac{1}{\sqrt{\hbar C \omega_r}}\hat{Q} = \frac{n_{zpf}}{\sqrt{2}}\hat{n}, \tag{2.13}$$

Here, ϕ_{zpf} and n_{zpf} represent the zero-point fluctuations of the two coordinates, expressed in terms of the reduced charge: $\hat{n} = \hat{Q}/(2e)$, and the reduced flux: $\hat{\phi} = 2\pi\hat{\Phi}/\Phi_0$, where $2e$ is the combined charge of a Cooper pair in the superconducting circuit, and Φ_0 denotes the flux quanta. Defining the photon number operator as $\hat{N} = \hat{a}^\dagger\hat{a}$, the following commutation relations will prove valuable later:

$$\left[\hat{a}, \hat{a}^\dagger\right] = 1, \qquad \left[\hat{N}, \hat{a}^\dagger\right] = \hat{a}^\dagger \qquad \left[\hat{N}, \hat{a}\right] = -\hat{a} \tag{2.14}$$

Written in terms of the reduced charge and flux, the Hamiltonian for the QHO can be rewritten as:

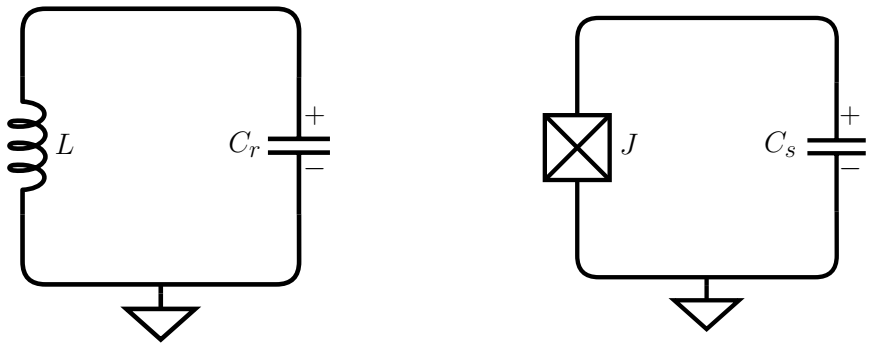
$$\hat{H} = 4E_c\hat{n}^2 + \frac{1}{2}E_L\hat{\phi}^2, \tag{2.15}$$

where $E_C = e^2/(2C) = E_L/(32n_{zpf}^4)$ and $E_L = (\Phi_0/2\pi)^2 = 2E_C/\phi_{zpf}^4$. The LC circuit boasts a notable advantage in its ability to offer protection against dispersion, a crucial factor for achieving long coherence times. To ensure this protection, the energy separation between adjacent states, $\hbar\omega_r$, must be considerably greater than the energy of the mean thermal fluctuations at the operational temperatures, $k_B T$. Generally, qubits are operated at a

temperature of approximately $T \approx 20\text{mK}$, with conducting qubits possessing frequencies surpassing 4GHz. Consequently, the noise levels encountered are several orders of magnitude lower than the frequencies employed to drive transitions [10, 11].

Despite this useful property of the QHO, its linear characteristics hinder its utility for facilitating quantum information processing. In accordance with DiVincenzo's first criterion, a qubit must operate in a superposed state consisting of binary values. Due to the linear nature of the QHO, the transition energies between adjacent energy eigenstates are degenerate, presenting a challenge for effective qubit implementation.

To address this issue, it is necessary to explore energy regimes that expose a subspace within the larger Hilbert space, in which single-level excitations are exclusively permitted. A methodology must be developed to confine the qubit's logical states such that they operate strictly within this two-state system. In essence, the goal is to drive excitations between the logical states of the quantum system without simultaneously inducing excitations between all other contiguous energy eigenstates.



(a) The LC circuit.

(b) Josephson Junction circuit.

Figure 1: Two distinct electrical circuit diagrams are pictured above. In (a), a series arrangement of an inductor, L, and a capacitance, C, is presented, also commonly referred to as a resonator. The coiled wire of the inductor is responsible for inducing a magnetic field, while the plates of the capacitance generate an electrical field. In (b), a simple transmon qubit is illustrated. The effects of the junction, J, are comprehended exclusively through quantum mechanics, originating from the tunneling of Cooper pairs, which alters the branch flux and functions as a non-linear inductance when contrasted with the LC-circuit.

2.2 Josephson junction and the transmon qubit

It is imperative to consider only quantum systems in which the energy separation between higher states is sufficiently distinct from the energy separation of other states. An effective Hamiltonian for the circuit must be devised that

retains the benefits of the LC-circuit, while allowing the dynamics of higher-level states to be disregarded. The incorporation of a non-linear, dissipationless component into the circuit serves as a solution. By substituting the linear inductor with a Josephson junction, a non-linear term can be introduced to the Hamiltonian.

The Josephson element has been demonstrated to exhibit the behavior of a non-linear inductor[12]. With the Hamiltonian expressed in terms of the branch flux, the inductance term in the equations can be readily replaced with the energy of the Josephson junction. Originating from the Josephson equations, two new equations for the voltage and currents traversing the circuit are obtained:

$$I = I_C \sin(\hat{\phi}), \tag{2.16}$$

$$V = \frac{\hbar}{2e}\frac{\mathrm{d}\hat{\phi}}{\mathrm{d}t}, \tag{2.17}$$

In this context, I_C represents the critical current, defined as the maximum supercurrent that can flow through the junction without developing any voltage across it. When the current flowing through the junction surpasses the critical current, the superconducting properties of the junction are lost. By employing Eq.(2.16) and Eq.(2.17), the energy of the Josephson circuit element can be derived as follows:

$$U_J = \int_{-\infty}^{t} V(\tau)I(\tau)\mathrm{d}\tau = \frac{I_C\hbar}{2e}\int_{-\infty}^{t} \dot{\phi}\sin(\phi)\mathrm{d}\tau = -E_J\cos(\phi). \tag{2.18}$$

E_J is called the Josephson energy, and it replaces the inductive energy in the QHO. The Hamiltonian for the non-linear circuit is defined by incorporating the new potential energy term as

$$\hat{H} = 4E_C\hat{n}^2 - E_J\cos\left(\hat{\phi}\right), \tag{2.19}$$

where C is now regarded as the total capacitance of the circuit, deriving from both the contribution of the previous capacitance and the capacitance obtained from the junction itself. In order to include the lowest order non-linear terms, the $\cos(\hat{\phi})$ is Taylor expanded to 4th order terms, and the Hamiltonian is expressed in natural units, such that $\hbar$ can be disregarded:

$$\begin{aligned}\hat{H} &\approx 4E_C\hat{n}^2 + \frac{1}{2}E_J\hat{\phi}^2 - \frac{1}{24}E_J\hat{\phi}^4 - E_J \\ &= \omega_0\left(\hat{a}^\dagger\hat{a} + \frac{1}{2}\right) - \frac{E_C}{12}\left(\hat{a}^\dagger + \hat{a}\right)^4 - E_J\end{aligned} \tag{2.20}$$

where it is noted that $E_J\phi_{zpf}^4 = 2E_C$. The zero point energy does not influence the qubit dynamics and can thus be neglected if the focus is solely on the system evolution. Therefore, constant terms are dropped from the Hamiltonian.

Eq. (2.20) can be rewritten in terms of its eigenvalues $H = \sum_n E_n |n\rangle\langle n|$. Employing perturbation theory, the energy to first order, E_n, can be derived:

$$E_n \approx \omega_0 n - \frac{E_C}{12}\left(6n^2 + 6n + 3\right) = \omega_q n - \frac{E_C}{2} n(n-1) + constant, \quad (2.21)$$

where $\omega_q = \omega_0 - E_C$ represents the resonance frequency of the transmon qubit. The constant term is disregarded, and the Hamiltonian is expressed as

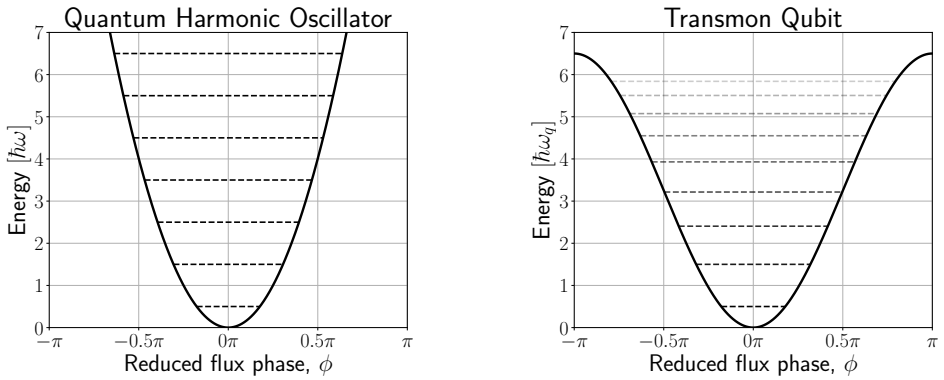
$$\hat{H} = \sum_n \left(\omega_q - \frac{E_C}{2}(n-1)\right) n\,|n\rangle\langle n|\,. \quad (2.22)$$

Utilizing the identity, $\hat{a}^\dagger\hat{a} = \sum_n n\,|n\rangle\langle n|$, Eq. (2.22) can be rewritten in terms of the creation and annihilation operators, yielding $\hat{H} = \omega_q\hat{a}^\dagger\hat{a} + \delta\hat{a}^\dagger\hat{a}^\dagger\hat{a}\hat{a}/2$, where $\delta = -E_C$, which corresponds to the Hamiltonian of the Duffing oscillator. By ensuring that E_C is sufficiently large, excitations of higher energy can be avoided, effectively transforming Eq (2.22) into a Hamiltonian describing the dynamics of only the ground and first excited state: $\hat{H} = \omega_q\,|1\rangle\langle 1|$. Since the bottom of the Hamiltonian is not of concern, a constant term, $\omega_q I/2 = \omega_q/2\,(|0\rangle\langle 0| + |1\rangle\langle 1|)$, can be subtracted from the Hamiltonian, allowing the lowest two energy states to be written using the spin operator:

$$\hat{H} = -\omega_q\frac{\hat{\sigma}_z}{2}, \quad (2.23)$$

where $\hat{\sigma}_z = |0\rangle\langle 0| - |1\rangle\langle 1|$ represents the Pauli spin operator. Long lifetimes are not the sole consideration for improving the fidelity of a quantum circuit, see Sec. 4; the system should also be capable of implementing fast and reliable gate operations. To achieve this, the ability to precisely tune the frequency of individual qubits, bringing them into or out of resonance, is often desirable for finely controlling interactions and entanglement among the qubits.

One method to engineer tunable qubits involves the utilization of superconducting quantum interference devices (SQUIDs) by replacing a single junction with two parallel Josephson Junctions. This approach effectively modifies the Hamiltonian, permitting the variation of the Josephson energy[13, 14], and consequently, the adjustment of the qubit frequency, ω_q. Such qubits are commonly known as split-transmon qubits. Despite the benefits of tunable qubits, they come with a downside: the introduction of additional error due to flux fluctuations. Various techniques have been developed to combat this noise, including the implementation of flux qubits[15]. For the purposes of this paper, it is assumed that these errors are sufficiently mitigated, allowing the selection and alteration of qubit frequencies during simulation time without significant decoherence.



(a) (b)

Figure 2: (a) Illustrates the energy potential of the quantum harmonic oscillator originating from the parallel LC-oscillator circuit. The units along the vertical direction are measured in transition energy of the resonator frequency, such that the energy levels of the resonator, given by $E_n = \hbar\omega_r(n + 1/2)$, representing the energy of the n photons populating the resonator, are each separated by a single tick. (b) Depicts the Josephson Junction potential, shifted so that the bottom of the sinusoidal well rests at zero. The units are measured in the transition frequency between the ground and the first excited state, which corresponds to the energy of the qubit's idling frequency. The non-linearity of the potential disrupts the degeneracy of the states, resulting in a non-constant energy separation, allowing for a restriction to a two-level dynamical subspace where quantum computation can occur.

2.3 Interactions

The transmon qubit enables the creation of a qubit, yet it does not provide control over its manipulation or facilitate entanglements between different qubits. To design such interactions, the circuit of the transmon qubit must be coupled to another circuit without compromising the qubit dynamics or its coherence time. A viable solution to this challenge involves coupling the qubit to another circuit through the use of a capacitor with capacitance C_g.

For two circuits with capacitance C_1 and C_2, and a potential $U(\Phi_1, \Phi_2)$, dependent only on the constituent branch fluxes, the circuits are capacitively coupled by C_g, as depicted in Fig. 3. By applying Kirchhoff's law, it is determined that the change in flux running through the capacitor is given by $\dot{\Phi}_1 - \dot{\Phi}_2$. Consequently, the complete circuit possesses a Lagrangian[16]:

$$\mathcal{L} = \frac{1}{2}C_1\dot{\Phi}_1^2 + \frac{1}{2}C_2\dot{\Phi}_2^2 + \frac{1}{2}C_g\left(\dot{\Phi}_1 - \dot{\Phi}_2\right)^2 - U(\Phi_1, \Phi_2). \tag{2.24}$$

The kinetic energy can be written using matrices, by defining the vectors $\mathbf{\Phi}$, and the matrix $\hat{C}$:

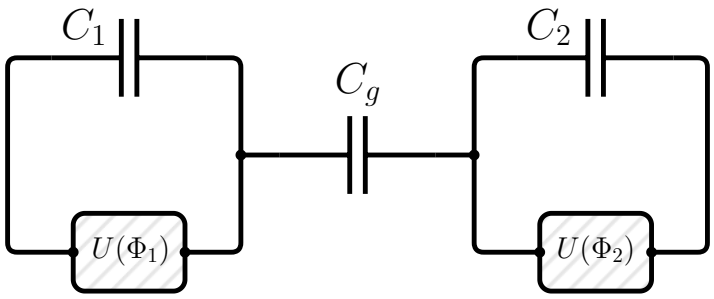


Figure 3: Schematic of two circuits that are capacitively coupled with a capacitance C_g. The two circuits have a self capacitance of C_1 and C_2 for the first and second circuit respectively. Each circuit has an inducting component, which is not necessarily linear, giving the Lagrangian a potential $U(\mathbf{\Phi})$.

$$\mathbf{\Phi} = \begin{pmatrix} \Phi_1 \\ \Phi_2 \end{pmatrix}, \qquad (2.25) \qquad \hat{C} = \begin{bmatrix} C_1 + C_g & -C_g \\ -C_g & C_2 + C_g \end{bmatrix}. \qquad (2.26)$$

With the defined matrices, the Lagrangian can be transformed into a more concise form:

$$\mathcal{L} = \frac{1}{2}\dot{\mathbf{\Phi}}^\intercal \hat{C} \dot{\mathbf{\Phi}} - U(\mathbf{\Phi}). \tag{2.27}$$

In order to determine the Hamiltonian, as was done in Eq. (2.7), the momentum conjugate must first be identified, which corresponds to the charges of the two capacitors with capacitance C_1 and C_2:

$$Q_1 = \frac{\mathrm{d}\mathcal{L}}{\mathrm{d}\dot{\Phi}_1} = C_1\dot{\Phi}_1 + C_1\left(\dot{\Phi}_1 - \dot{\Phi}_2\right), \tag{2.28}$$

$$Q_2 = \frac{\mathrm{d}\mathcal{L}}{\mathrm{d}\dot{\Phi}_2} = C_2\dot{\Phi}_2 - C_2\left(\dot{\Phi}_1 - \dot{\Phi}_2\right). \tag{2.29}$$

Expressed in vector form, $\mathbf{Q} = (Q_1, Q_2)^\intercal$, Eq. (2.28) and (2.29) provide the equations of motion. These equations can then be reformulated in terms of the change in branch flux:

$$\mathbf{Q} = \hat{C}\dot{\mathbf{\Phi}} \Rightarrow \dot{\mathbf{\Phi}} = \hat{C}^{-1}\mathbf{Q}. \tag{2.30}$$

Considering that $\hat{C}$ is symmetrical, its inverse is also symmetrical, leading to $(\hat{C}^{-1}\mathbf{Q})^\intercal = \mathbf{Q}^\intercal\hat{C}^{-1}$. The Hamiltonian is obtained using the Legendre transformation:

$$\begin{aligned} H &= \dot{\mathbf{\Phi}}^\intercal\mathbf{Q} - \mathcal{L} = \dot{\mathbf{\Phi}}^\intercal\mathbf{Q} - \frac{1}{2}\dot{\mathbf{\Phi}}^\intercal\hat{C}\dot{\mathbf{\Phi}} + U(\mathbf{\Phi}) \\ &= \frac{1}{2}\mathbf{Q}^\intercal\hat{C}^{-1}\mathbf{Q} + U(\mathbf{\Phi}). \end{aligned} \tag{2.31}$$

If the two circuits are sufficiently separated, such that the magnetic flux of one system does not affect the potential of the other system, the total potential can be represented as a sum over the two coupled circuits, $U(\mathbf{\Phi}) = U_1(\Phi_1) + U_2(\Phi_2)$. By calculating the inner product, the Hamiltonian becomes:

$$H = \frac{1}{2}\frac{Q_1^2}{\tilde{C}_1} + \frac{1}{2}\frac{Q_2^2}{\tilde{C}_2} + \frac{Q_1 Q_2}{\tilde{C}_g} + U_1(\Phi_1) + U_2(\Phi_2), \tag{2.32}$$

where $Q_i = (C_j + C_g)/((C_1 + C_g)(C_2 + C_g) - C_g^2)$, for $(i \neq j)$ and $i, j = 1, 2$, while $\tilde{C}g = C_g/((C_1 + C_g)(C_2 + C_g) - C_g^2)$. The interaction term, $H_{int} = Q_1 Q_2/\tilde{C}_g$, facilitates energy exchanges between the two systems. Moreover, the coupling induces a shift in the kinetic energy of each of the two circuits, altering it by an amount $1/\tilde{Q}_i - 1/Q_i$.

Consider a scenario in which a transmon qubit is capacitively coupled to an LC resonator circuit through a weak capacitance C_g, such that $C_q, C_r \gg C_g$. In this situation, the shift in kinetic energy becomes negligible, resulting in $H = H_q + H_r + H_{int}$. Upon appropriate quantization of the two circuits, with $Q_i \to 2e\hat{n}_1$, it is found that:

$$\begin{aligned} \hat{H}_{int} &= \frac{C_g}{E_{C_r} E_{C_q}} \hat{n}_r \hat{n}_q = \frac{C_g n_{zpf,r} n_{zpf,q}}{E_{C_r} E_{C_q}} i^2 (\hat{a}_r - \hat{a}_r^\dagger)(\hat{a}_q - \hat{a}_q^\dagger) \\ &= g(\hat{a}_r - \hat{a}_r^\dagger)(\hat{a}_q^\dagger - \hat{a}_q), \end{aligned} \tag{2.33}$$

In this case, $g = C_g n_{zf,r} n_{zf,q}/(E_{C_r} E_{C_q})$. For the two lowest levels of the transmon qubit, the annihilation operator is rewritten as a spin down operator, with $\hat{a}_q \to \hat{\sigma}_-$. When considering frequencies where higher-order terms, such as double excitation $(\hat{\sigma}_+ \hat{a}^\dagger)$ and de-excitation terms $(\hat{\sigma}_- \hat{a})$, average to zero, the circuit picture becomes analogous to actual fermions with $1/2$ spins placed in a microwave cavity, see Fig. 4. Consequently, the interaction Hamiltonian takes the form of the Jaynes-Cummings interaction Hamiltonian,

$$\hat{H}_{int} = g(\hat{\sigma}_+ \hat{a} + \hat{\sigma}_- \hat{a}^\dagger). \tag{2.34}$$

The qubits can be excited at the cost of a photon in the cavity, and vice-versa. For a two-level qubit, the complete Hamiltonian between the LC resonator (2.9), and the transmon qubit (2.23), described by the Jaynes-Cummings Hamiltonian, is for a single qubit

$$\hat{H} = \frac{1}{2}\omega_q \hat{\sigma}_z + \omega_r \hat{a}^\dagger \hat{a} + g(\hat{\sigma}_+ \hat{a} + \hat{\sigma}_- \hat{a}^\dagger). \tag{2.35}$$

Recent advancements in superconducting qubits have significantly extended coherence times, reaching into the millisecond regime[17]. The coherence time is anticipated to increase by about a factor of ten over the next three years, demonstrating a promising trajectory for the development of these qubits[18]. This trend towards exponentially longer coherence times in superconducting

circuits has been dubbed Schoelkopf's law, named in honor of the substantial contributions made by Robert J. Schoelkopf in the field of superconducting qubits. As a result of these advancements and the potential for further improvements, the transmon qubits present a highly viable avenue for research into universal quantum computers and the ongoing quest to develop practical quantum computing technologies.

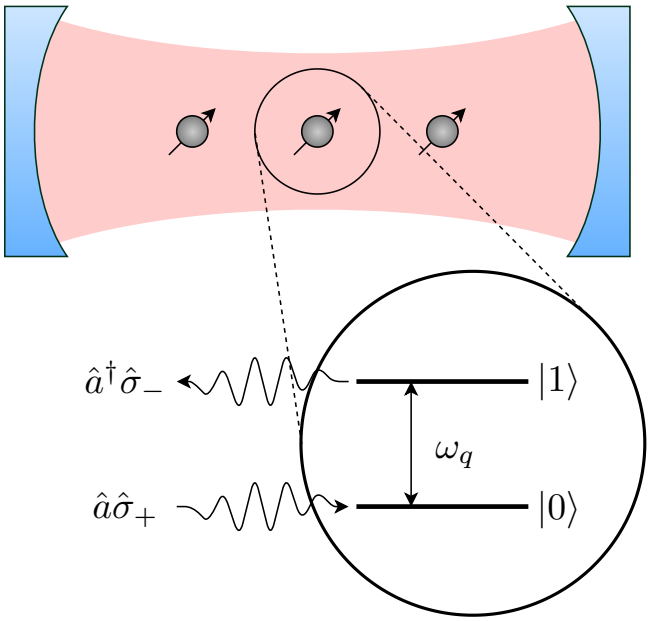


Figure 4: Depiction of the optical representation of qubits placed inside a resonator cavity. A qubit, such as a dopant atom with nuclear spin 1/2 in a solid-state material, may interact with the microwave field of the cavity through the emission or absorption of a photon in the cavity. It is important to emphasize that the analogy becomes imperfect when higher-level states of the transmon qubit are considered.

3 Extending the wave function formalism

Though the investigation of a single qubit in a cavity reveals a plethora of intriguing physics, it falls short of capturing many of the crucial phenomena that enables quantum supremacy. To delve into these concepts, it is imperative to broaden the scope of the system's physics into larger Hilbert spaces, encompassing the dynamics of multiple distinct qubits and enabling the construction of entangled states. This section aims to present the mathematical framework facilitating such an extension, specifically in cases where each qubit is coupled to the same resonator circuit.

3.1 Multiple qubits

The extension of a two-state Hilbert space into 2^n states for n different qubits is a straightforward process. A tensor operation that yields the tensor product of the various Hilbert spaces, $\mathcal{H}_1 \otimes \mathcal{H}_2 \otimes ... \otimes \mathcal{H}_n$, is required, ensuring that operations within any single subspace remain independent of the others. The mathematical operation exhibiting these properties on sets of matrices is the tensor operation, denoted by the same symbol, $\otimes$, and known as the Kronecker product. The Kronecker product generates new basic vectors from the single qubit state, such as $|0\rangle \otimes |0\rangle$, $|0\rangle \otimes |1\rangle$, $|1\rangle \otimes |0\rangle$, and $|1\rangle \otimes |1\rangle$ for a two-qubit system. For the sake of simplicity, these states are often represented as $|00\rangle$, $|01\rangle$, $|10\rangle$, and $|11\rangle$. A basic set of eigenvectors for the desired Hilbert space, specifically the one characterizing an n-qubit system without considering the resonator, can be readily constructed. These eigenvectors are derived from eigenvectors of the sub-Hilbert spaces comprising the system,

$$|m\rangle = \bigotimes_{q}^{n} |m_q\rangle \,. \tag{3.1}$$

In this context, each m_q represents the local constituent states of an individual qubit, assuming values of either $|0\rangle$ or $|1\rangle$ depending on the global eigenvector of $|m\rangle$. According to the principles of quantum mechanics, if each of these states constitutes a solution to the Schrödinger equation, then any linear combination of these states is also considered a valid solution,

$$|\psi\rangle = \sum_{m}^{2^n} c_m \, |m\rangle \,. \tag{3.2}$$

3.1.1 Density Matrix

It should be noted that if the state can be expressed as a product of states, such as $|\psi\rangle = |\psi_1\rangle \otimes |\psi_2\rangle \otimes \cdots \otimes |\psi_n\rangle$, then it does not represent an entangled state. By examining the pure state density matrix, defined as $\rho = |\psi\rangle\langle\psi|$, the density matrix for a non-entangled state, composed of two constituent states, can be similarly expressed as,

$$\begin{aligned} \rho &= |\psi\rangle\langle\psi| \\ &= (|\psi_1\rangle \otimes |\psi_2\rangle)(\langle\psi_1| \otimes \langle\psi_2|) \\ &= |\psi_1\rangle\langle\psi_1| \otimes |\psi_2\rangle\langle\psi_2| \\ &= \rho_1 \otimes \rho_2. \end{aligned} \tag{3.3}$$

The pure state density matrix provides an alternative approach for examining the system's evolution. However, for the density matrix to accurately describe

a physical process, it must satisfy several properties:

$$\rho^\dagger = \rho, \tag{3.4}$$

$$\rho^2 = |\psi\rangle\langle\psi|\psi\rangle\langle\psi| = |\psi\rangle\langle\psi| = \rho, \tag{3.5}$$

$$\mathrm{Tr}(\rho) = \sum_m \langle m|\psi\rangle\langle\psi|m\rangle = \sum_m |c_m|^2 = 1. \tag{3.6}$$

It is common to be in situations where complete knowledge about the state of a qubit is not available, for instance, even if complete knowledge is assumed upon initialization, the qubit will experience decoherence over time. A more comprehensive discussion on modeling decoherence processes can be found in Section 6.1. For the time being, it is sufficient to note that the pure state description may not adequately describe the state of the qubit after a certain period. In such cases, the qubit is said to be in a mixed state. As information about the system is lost over time, for instance due to dephasing or decay of the state, the best description that can be provided may involve assigning equal probabilities for the qubit to be in the states $|0\rangle$ and $|1\rangle$. In theory, there could be several states that the qubit might occupy. If the state begins in $|\psi\rangle$, it will evolve into a mixture of different states $|\psi_k\rangle$ over time, each with probability p_k. This scenario can be captured by defining the general density matrix as:

$$\rho = \sum_k p_k \, |\psi_k\rangle\langle\psi_k| \, . \tag{3.7}$$

Under the more general definition, it can be straightforwardly confirmed that two properties of the pure state density matrix also hold for a general mixed state density matrix, namely, the matrix being Hermitian (Eq. (3.4)) and having a unit trace (Eq. (3.6)). It should be noted that the density matrix is idempotent (Eq. (3.5)) only for pure states, providing a simple method to determine the purity of a state. Within this framework, the expectation value of any observable is given by:

$$\langle A \rangle = \sum_k p_k \, \langle \psi_k | \hat{A} | \psi_k \rangle = \mathrm{Tr}\Big(\rho \hat{A}\Big). \tag{3.8}$$

Decoherence processes are, in practice, attributable to entanglement with the environment. Two distinct scenarios may necessitate the use of a mixed state to describe the system. The first scenario is akin to the Bayesian interpretation of probability, where an observer might lack complete knowledge about the state of the system, and therefore, the mixed state can be employed to describe their uncertainty. In this situation, another observer might use a different density matrix to characterize the same system. The second use of mixed state, is in the description of a system entangled with another. Interestingly, mixed state dynamics allow for the exclusion of information about the larger system. If the larger system is in an entangled state, then even if complete

knowledge exists about the larger system, the subsystem must be described using mixed state dynamics.

Consider the Bell state $|\psi^-\rangle = (|0_A 1_B\rangle - |1_A 0_B\rangle)/\sqrt{2}$, where qubits belonging to systems A and B have been explicitly designated. When the leading qubit, A, is the only one of interest, the focus can be restricted to that system by taking the partial trace over system B on its pure state density matrix, $\rho = |\psi^-\rangle\langle\psi^-| = (|0_A 1_B\rangle - |1_A 0_B\rangle)(\langle 0_A 1_B| - \langle 1_A 0_B|)/2$, resulting in the mixed state density matrix exclusively for qubit A:

$$\begin{aligned}\rho_A = \mathrm{Tr}_B(\rho) &= \sum_{m_B=0}^{1} (I_A \otimes \langle m_B|)\, \rho\, (I_A \otimes |m_B\rangle) \\ &= \frac{1}{2} \sum_{m_B=0}^{1} (I_A \otimes \langle m_B|)\, (|0_A 1_B\rangle - |1_A 0_B\rangle)\, (\langle 0_A 1_B| - \langle 1_A 0_B|)\, (I_A \otimes |m_B\rangle) \\ &= \frac{1}{2}\, (|0_A\rangle\langle 0_A| + |1_A\rangle\langle 1_A|)\,.\end{aligned} \tag{3.9}$$

In this case, $I_A \otimes |m_B\rangle\, (|0_A 1_B\rangle - |1_A 0_B\rangle) = (|0_A\rangle\, \langle m_B|0_B\rangle - |1_A\rangle\, \langle m_B|1_B\rangle)$ was utilized, and similarly for the bra side, resulting in only the terms where m_B was the same state surviving. It is observed that ρ_A is a 50/50 density matrix, indicating that the most accurate description of the system for qubit A is a 50/50 superposition of its two basic states. The same procedure could have been applied for qubit B, yielding an identical density matrix to that of system A. This concept of tracing out the second qubit while preserving only the information on the first qubit is also employed in Sec. 6.1 to derive the Markov-Born master equation, which describes the effects of the larger environment on the qubit state.

3.1.2 Coupling additional qubits to the resonator

Revisiting the Jaynes-Cummings Hamiltonian from earlier, the addition of more qubits to the cavity results in extra spinors. To ensure that each spinor operates on the correct constituent state, i.e., on the appropriate Hilbert spaces, their corresponding operators are defined as follows:

$$\hat{\sigma}_k^{(s)} = \bigotimes_r \begin{cases} \hat{\sigma}_k & \text{if } s = r, \\ I & \text{otherwise,} \end{cases} \tag{3.10}$$

In this context, $\hat{\sigma}_k$ denotes the chosen Pauli matrix, e.g., $\hat{\sigma}_z^{(1)}/2 = \hat{\sigma}_z \otimes I/2$ and $\hat{\sigma}_z^{(2)}/2 = I \otimes \hat{\sigma}_z/2$ represent the spin matrices along the z direction for a pair of qubits. Similar operators can be defined in a straightforward manner. As evidenced by Eq. (3.10), each such operator must commute with a different operator acting on a distinct Hilbert space, while their commutation

with operators on the same Hilbert space must obey the same laws as they otherwise would. In particular, the new spin matrices must obey the following commutation relations:

$$\left[\hat{\sigma}_{\pm}^{(s)}, \hat{\sigma}_{z}^{(q)}\right] = \pm 2\delta_{sq}\hat{\sigma}_{\pm}^{(s)}, \tag{3.11}$$

$$\left[\hat{\sigma}_{+}^{(s)}, \hat{\sigma}_{-}^{(q)}\right] = \delta_{sq}\hat{\sigma}_{z}^{(s)}, \tag{3.12}$$

where δ_{sq} represents the Kronecker delta function. The advantage of such a notation can be readily demonstrated by describing a larger system composed of numerous transmon qubits interacting via a resonator. Each qubit within the cavity can similarly exchange excitation with the resonator by either absorbing a photon from, or emitting a photon to the cavity. Thus, the Hamiltonian of Eq. (2.35) assumes the form:

$$\hat{H} = \hbar\omega_r \hat{a}^\dagger \hat{a} + \sum_s \left[\hbar\omega_s \frac{\hat{\sigma}_z^{(s)}}{2} + g\hbar\left(\hat{a}^\dagger \hat{\sigma}_-^{(s)} + \hat{a}\hat{\sigma}_+^{(s)}\right)\right]. \tag{3.13}$$

Here, the summation is of various qubits within the resonator. The resonator functions as a mediator for these qubits and can be employed to generate entanglements between different qubits. In case of homodyne detection, this setup has been studied extensively in the bad cavity limit. Quantum trajectories for this setup in the bad cavity limit, as well as balanced homodyne detection schemes (see Sec. 6.2.1), have been explored[19]. In this work, these effects will be examined within the context of typical readout regimes.

3.2 Quantum entanglements

As previously mentioned, an entangled state is characterized by the inability to factorize its state-vector or its density matrix (as demonstrated in Eq. (3.3)) into products of its constituent states, signifying that the entangled pair forms a single quantum state. In Sec. 2, it was noted that interactions are necessary to entangle qubits. This becomes readily apparent when examining a pair of Hamiltonians, $\hat{H}_1$ and $\hat{H}_2$, where $\hat{H}_1$ operates solely on the first qubit, and $\hat{H}_2$ on the second, meaning they operate on their respective Hilbert spaces, $\mathcal{H} = \mathcal{H}_1 \otimes \mathcal{H}_2$, and their Hamiltonians commute. If the state is a non-entangled pure-state, the dynamics of the state vector are as follows:

$$\frac{\mathrm{d}\left|\psi(t)\right\rangle}{\mathrm{d}t} = -i\left(\hat{H}_1 + \hat{H}_2\right)\left|\psi_1(t)\right\rangle \otimes \left|\psi_2(t)\right\rangle, \tag{3.14}$$

$$\begin{aligned}\left|\psi(t)\right\rangle &= e^{-i\left(\hat{H}_1+\hat{H}_2\right)t}\left|\psi_1(0)\right\rangle \otimes \left|\psi_2(0)\right\rangle \\ &= e^{-i\hat{H}_1 t}\left|\psi_1(0)\right\rangle \otimes e^{-i\hat{H}_2 t}\left|\psi_2(0)\right\rangle.\end{aligned} \tag{3.15}$$

As the interaction term in the Jaynes-Cummings Hamiltonian is the only component connecting degrees of freedom between the qubits, an evident method

for maintaining a non-entangled state involves detuning the resonator frequency ω_r significantly away from any of the qubit frequencies.

In the realm of quantum information mathematics, entanglements originate from a series of unitary matrices operating on the space of a single qubit, as well as a unitary operation acting on the space of both qubits. These operations are discussed in Sec. 4, along with a brief explanation of the physical mechanisms behind them.

Modeling entanglements constitutes one of the challenging issues in simulating qubit dynamics on a classical computer, as it causes time and memory to scale with an $\mathcal{O}(2^n)$ complexity in relation to the qubit count. Although optimization methods for simulation performance will be discussed later in section 7, the inherent nature of the problem makes it impossible to mitigate the exponential scaling.

For quantum supremacy, algorithms must harness the capability to operate in parallel on a superposition of multiple classical inputs. However, to obtain only the correct output, incorrect answers must undergo destructive interference before readout commences. Quantum entanglement plays a pivotal role in accomplishing this. A few algorithms have already been identified, such as Shor's prime factorization algorithm[20] and Grover's fast search algorithm[21]. Entanglements constitute the singular aspect of quantum computing that cannot be simulated within a reasonable time frame on a classical computer. As a result, entanglement serves as an essential ingredient enabling a quantum computer to achieve quantum supremacy.

4 Logical gates in a quantum Processor

A prerequisite for the performance of universal quantum processing is a method to transform the state of a initialized qubit into any other state the qubits could assume. A single qubit can take on all superpositions of the form:

$$|\psi\rangle = \cos(\theta)|0\rangle + e^{i\phi}\sin(\theta)|1\rangle . \tag{4.1}$$

To reiterate a key aspect of Sec. 3.2, it is insufficient to implement operators that perform transformations on single qubits at a time. Such operators can only produce product states comprised of constituent states of the general form given in Eq. (4.1). To utilize the power of a qubit, multi-qubit gates that can generate entanglements between the qubits is essential. To understand how universal computation in a quantum computer might be achieved, it is beneficial to first familiarize oneself with the mathematical concepts that enable classical computers to attain Turing completeness.

4.1 Universality of classical gates

It is widely recognized that a classical computer operates exclusively on bits of data. Therefore, any logical operation processed on a string of bits, must

Input		Not		OR	AND	XOR
x_1	x_2	$\neg x_1$	$\neg x_2$	$x_1 \vee x_2$	$x_1 \wedge x_2$	$x_1 \oplus x_2$
0	0	1	1	0	0	0
0	1	1	0	1	0	1
1	0	0	1	1	0	1
1	1	0	0	1	1	0

Table 1: The truth table of the three basic logical operations, conjunction (AND), disjunction (OR), negation (NOT), outlining the universality of classical gates, in addition to the secondary operator, the exclusive or (XOR).

have a Boolean function associated with it. Achieving universality in a classical computer thus amounts to identifying generators of the Boolean algebra. Implementing operations for all possible Boolean logical operations is not required. Instead, to achieve classical universality, it is sufficient to implement a universal set of Boolean gates.

In the instance of a single bit, two possible outcomes can be mapped, thereby requiring the existence of two possible unary operations in the Boolean algebra: the trivial identity operator, which maintains the input unchanged, and the complement, also referred to as the not operator, typically denoted as $\neg$, which flips the input bit. It is important to highlight that these operations are entirely reversible, meaning that the input value can be determined from the output.

A binary Boolean operation maps two input bits onto a single output bit. The two bits permute among four input values, each of which can map onto two potential output values. Consequently, there exist $4 \times 2 = 8$ distinct binary operations. In contrast to the unary bit gates, these operations are entirely irreversible from the output values. For the sole purpose of achieving universality, direct implementing all eight gates is deemed an unnecessary complexity. It can be shown that only two binary operators are required, namely the AND operator, denoted as $\wedge$, and the OR operator, denoted as $\vee$.

For universality, it is sufficient to construct a set of three gates, namely AND, OR, and NOT, and utilize combinations of these three to enable the operation of every Boolean function. Table 1 displays the truth table for the OR and AND gates, while also illustrating the truth values for the combined XOR gate, constructed from the other gates as, $(x_1 \vee x_2) \wedge \neg(x_1 \wedge x_2)$, on inputs $x_1, x_2 = \{0, 1\}$. The XOR is used for the addition of two binary numbers on a classical computer, typically denoted with the operator $\oplus$, the modulus 2 of $x_1 + x_2$.

4.2 Single qubit gates

The quantum format of the Boolean operators in classical processing, constitutes a complete set of unitary operators. Quantum information processing

is not generally designed to operate only on strings of binary digits. Rather, a collection of n qubits can represent the superposition of any binary string permutation with a length of n digits. A single qubit, expressed in the general form of Eq.(4.1), can assume a continuum of different values. For this reason, a single qubit gate is inadequate for transforming the qubit into any permitted superposition. Universality requires the capability to execute operations on arbitrary single qubit states, $|\psi\rangle = a\,|0\rangle + b\,|1\rangle$, mapping the infinitely many potential input states onto infinitely many possible output states. This section is devoted to the theoretical considerations for a universal set of quantum gates, as well as how they might be practically implemented for the superconducting qubit.

4.2.1 Bloch-sphere

To built a more insightful intuition of the two-level quantum state, it is convenient to construct a visual geometrical representation. Group theory informs that the two-level system constitutes a 1/2 representation of the special unitary group, $SU(2)$. The group under examination is locally isomorphic to the special orthogonal group, $SO(3)$, which corresponds to the group of rotations for the three-dimensional real-numbered vector space, $\mathbb{R}^3$. Given the isomorphism, it is entirely feasible to represent any two-level state vector, $\psi(t) = \exp\Big(-i\int_0^t \hat{H}\mathrm{d}\tau\Big)\psi(0) = U(t)\psi(0)$, at any time, t, as a point on a sphere, where $U(t)$ consequently signifies a rotation on this sphere.

To construct this mapping, the density matrix of a single qubit must first be considered. Any 2×2 Hermitian traceless matrix can be depicted as a linear combination of the Pauli matrices. To establish a valid density matrix, the trace condition is ensured by adding the identity matrix, multiplied by a factor of 1/2. Therefore, a general representation of the single qubit system density matrix is given by:

$$\rho = \frac{1}{2}\left(I + u_x\hat{\sigma}_x + u_y\hat{\sigma}_y + u_z\hat{\sigma}_z\right) = \frac{1}{2}\left(I + \mathbf{u}\cdot\boldsymbol{\sigma}\right), \tag{4.2}$$

where $\mathbf{u}$ is referred to as the Bloch vector, and $\boldsymbol{\sigma}$ is a three-dimensional vector with the Pauli matrices as elements. The density matrix is a positive semi-definite matrix, implying that for all vectors $\mathbf{v} \in \mathbb{C}^n$, the density matrix complies with the condition $\mathbf{v}^\dagger\rho\mathbf{v} \geq 0$. Demonstrating this condition is straightforward. Any such vector can be written as a linear combination of basis states, $\mathbf{v} = \sum_m v_m\,|m\rangle \in \mathbb{C}^n$. Then, the eigenvalues for the density matrix, $\sum_k p_k = 1$, ensure that $0 \leq p_k \leq 1$, leading to $\sum_k p_k|v_k|^2|c_k|^2 \geq 0$. This introduces an additional constraint on the Bloch vector, specifically that $\|\mathbf{u}\| \geq 1$. All the degrees of freedom for ρ are represented by the Bloch vector, $\mathbf{u}$, and any evolution signifies a change in the vector, which, for pure states where $\mathrm{Tr}(\rho^2) = \mathrm{Tr}(\rho) = 1 \Rightarrow (\mathbf{u}^\intercal\boldsymbol{\sigma})^2 = \|\mathbf{u}\| = 1$, corresponds to a rotation on the Bloch sphere. As a result, this geometrical representation can also

provide a useful visual aid for certain decoherence processes that cause the Bloch vector to gravitate towards the center of the Bloch sphere.

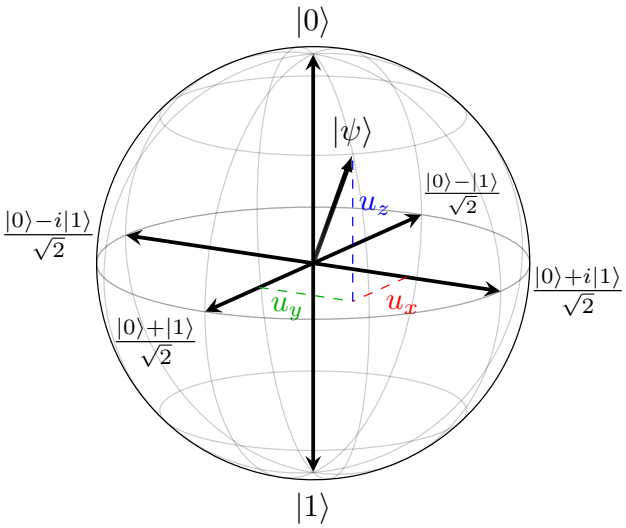


Figure 5: An illustration of the Bloch sphere. The three axis, x, y, and z, along which the states $(|0\rangle + |1\rangle)/\sqrt{2}$, $(|0\rangle + i\,|1\rangle)/\sqrt{2}$, and $|0\rangle$ are respectively located. These axes are designated as such due to their corresponding Pauli matrices. The three components of the Bloch vector $\mathbf{u}$, representing the state $|\psi\rangle$, are explicitly depicted.

The diagonal elements of the density matrix correspond to state amplitudes, signifying that the north and south poles of the sphere represent the computational states, $|0\rangle$ and $|1\rangle$, respectively. Additionally, trajectories along longitudinal lines represent changes in θ. The off-diagonal elements of the density matrix constitute the coherence, with the equator representing an equal superposition of $|0\rangle$ and $|1\rangle$, and rotations along latitudinal lines corresponding to shifts in the qubit phase, ϕ. Refer to Fig.5 for a visualization of the Bloch sphere. The Bloch sphere visually demonstrates that pure states are completely described by the two phases, in accordance with Eq.(4.1).

4.2.2 Universality of single qubit quantum gates

The well-established normalization constraint of quantum state vectors dictates that their evolution is described by unitary operators. In fact, it is relatively simple to show that the generator of any unitary operator is a Hermitian matrix. For the unitary operators of the subgroup $SU(N) \subset U(N)$, the constraint of $\det \hat{U} = 1$ implies that these generators possess additional properties; specifically, they are all traceless, and have anti-Hermitian commutator that obeys the relation[22]:

$$[T_a, T_b] = \sum_c i f_{abc} T_c, \tag{4.3}$$

where f_{abc} represents the anti-symmetric structure constant of the algebra. For a single qubit, whose evolution is described by a matrix in $SU(2)$, three Hermitian traceless matrices that satisfy these conditions are readily identified as the three Pauli matrices. In the Bloch sphere representation, rotations can occur in any of the three major planes, characterized by their rotation angle, denoted by θ_k, around the k'th axis perpendicular to the plane of rotation, where $k = \{x, y, z\}$. As per the Lie algebra, any unitary operation on the Bloch sphere is generated by a linear combination of the three Pauli matrices and can be expressed as:

$$U(\theta_x, \theta_y, \theta_z) = e^{i \sum_k \frac{\theta_k}{2} \sigma_k}, \tag{4.4}$$

where σ_k represents the corresponding Pauli matrix. The implementation of Hamiltonians that enable such gate operations frequently presents additional challenges, such as the potential for one of the three rotations to exist only virtually, as discussed in Sec. 4.2.4. A more direct approach entails rotations around a single axis at a time:

$$\hat{R}_k(\theta_k) = e^{i \frac{\theta_k}{2} \hat{\sigma}_k}. \tag{4.5}$$

The aforementioned gates are commonly referred to as non-Clifford gates, or rather, all but a few angles of $\hat{R}_k(\theta_k)$ results in a Clifford gate, as opposed to Clifford gates, which are discussed in Sec. 4.2.3. In principle, every unitary gate can be constructed using these rotation matrices. This may appear intuitive, as any rotations can be described by three Euler angles.

At first glance, this might seem obvious, since any rotation can be characterized by three Euler angles. However, upon closer inspection, how someone constructs any arbitrary unitary matrix, e.g. $U(2a, 0, 2b)$, may not be immediately evident owing to the non-commuting property of the Pauli matrices, that is:

$$\hat{U} = e^{i(a\hat{\sigma}_x + b\hat{\sigma}_z)} \neq e^{ia\hat{\sigma}_x} e^{ib\hat{\sigma}_z}. \tag{4.6}$$

A straightforward method involves approximating the gate by constructing the right-hand side of the equation using sequences of smaller rotations. If a single small rotation takes place over a time Δt, the unitary evolution characterizing this rotation is approximately represented by $I - iH\Delta t$. Assuming the complete unitary gate requires the time t, the number of constituent gate rotations performed is $n = t/\Delta t$. The unitary operation of a small angle rotation $2a\Delta t$, around x, succeeded by another small angle rotation $2b\Delta t$, around y, is expressed by the unitary:

$$\Delta\hat{U} = (1 + ia\hat{\sigma}_x\Delta t)\,(1 + ib\hat{\sigma}_z\Delta t) = 1 + i\,(a\hat{\sigma}_x + b\hat{\sigma}_z)\,\Delta t + \mathcal{O}(\Delta t^2) \tag{4.7}$$

If each pulse, implementing a rotation, is short, such that $t \gg \Delta t$, the second-order terms in Δt may be neglected. Shortening the time duration of each

pulse increases the number of pulses needed to complete the gate operation, which signifies a large n. By selecting units where $t = 1$, the time is measured in units corresponding to the duration required to complete the entire sequence of rotations. As the limit for $n \to \infty$ is approached, Eq. (4.7) can be reformulated in terms of n in this limit:

$$\hat{U} = \lim_{n\to\infty} \left(1 + \frac{i\,(a\hat{\sigma}_x + b\hat{\sigma}_z)}{n}\right)^n = e^{i(a\hat{\sigma}_x + b\hat{\sigma}_z)}. \tag{4.8}$$

This expression aligns precisely with the gate of interest. Therefore, by applying this procedure, every gate operating exclusively on a single qubit can be realized by approximation. The fidelity of the desired gate should theoretically increase with a sufficiently large number of smaller rotation pulses. For large n, the subsequent approximation is thus deemed to be accurate:

$$\hat{U} = e^{i\sum_k^3 (a_k\hat{\sigma}_k/n)} \approx \prod_k^3 e^{ia_k\hat{\sigma}_k/n}. \tag{4.9}$$

Employing only the rotational gates, enables the mapping of any single qubit input to any position on the Bloch sphere, establishing these gates as a universal set of single qubit quantum gates.

4.2.3 Clifford gates

The Pauli matrices that generate the algebra, are also encompassed within the algebra. It is straightforward to verify, since these operators are Hermitian matrices and they square to the identity, hence they are unitary and also candidates for gate operations. The properties of σ_x, σ_y, and σ_z suggest that ensuring closure is straightforward by incorporating the remaining elements in the multiplication table. These elements form the Pauli group for single qubit. By convention, the gates represented by the Pauli matrices are designated as $\hat{\mathsf{X}}$, $\hat{\mathsf{Y}}$, and $\hat{\mathsf{Z}}$, corresponding to each respective axis. The single-qubit Pauli group, denoted by P_1, is a rank 16 group comprising the following members[23]:

$$P_1 = \{\pm I, \pm\hat{\mathsf{X}}, \pm i\hat{\mathsf{X}}, \pm\hat{\mathsf{Y}}, \pm i\hat{\mathsf{Y}}, \pm\hat{\mathsf{Z}}, \pm i\hat{\mathsf{Z}}\} \tag{4.10}$$

From a theoretical standpoint, the set of Clifford gates constitutes an intriguing collection of gates. Clifford gates represent the set of unitary operators that normalize the Pauli Group. For the Pauli group of the single qubit Hilbert space, the group of Clifford gates are therefore defined as:

$$\mathcal{N}(P_1) = \{\hat{U} \in U(2) \mid \hat{U}P_1\hat{U}^\dagger = P_1\}. \tag{4.11}$$

Utilizing the properties of the Pauli matrices, which include anti-commutation, hermiticity, and squaring to the identity, the following can be readily demonstrated:

$$\hat{\sigma}_k(\pm\hat{\sigma}_l)\hat{\sigma}_k^\dagger = \pm(\hat{\sigma}_k\hat{\sigma}_l)\hat{\sigma}_k = \pm(-\hat{\sigma}_l\hat{\sigma}_k)\hat{\sigma}_k = \mp\hat{\sigma}_l, \tag{4.12}$$

$$\hat{\sigma}_k(\pm I)\hat{\sigma}_k^\dagger = \pm\hat{\sigma}_k\hat{\sigma}_k = \pm I, \tag{4.13}$$

namely, each conjugation maps to a unique element in P_1, confirming the Pauli matrices themselves as members of the Clifford gates. A specific Clifford gate of interest is the Hadamard gate, denoted by $\hat{\mathsf{H}}$. The Hadamard gate executes π rotations about an axis running diagonally through the $x-y$ plane, as illustrated in Fig. 6. The matrix describing this unitary operator is provided by:

$$\hat{\mathsf{H}} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}. \tag{4.14}$$

An interpretation of the Hadamard gate, is that it swaps information accessible along x and z direction, such that eigenvectors of $\hat{\mathsf{Z}}$, is transformed into eigenvectors of $\hat{\mathsf{X}}$. By representing the gate using the outer product of the basis states, the translation between basis states becomes evident:

$$\hat{\mathsf{H}} = \frac{1}{\sqrt{2}} \left(|0\rangle\langle 0| + |0\rangle\langle 1| + |1\rangle\langle 0| - |1\rangle\langle 1| \right) = |+\rangle\langle 0| + |-\rangle\langle 1| \,. \tag{4.15}$$

Here $|\pm\rangle = 1/\sqrt{2}(|0\rangle \pm |1\rangle)$, are the basis states in the x direction, such that the eigenvalue equation yields; $\hat{\sigma}_x |\pm\rangle = \pm |\pm\rangle$. The benefit of the Clifford gates is that their conjugation with the Pauli gates, is by definition a member of the Pauli group. In the case of the Hadamard, this conjugation transforms the three Pauli gates as:

$$\hat{\mathsf{H}}\hat{\mathsf{X}}\hat{\mathsf{H}}^\dagger = \hat{\mathsf{Z}}, \tag{4.16}$$

$$\hat{\mathsf{H}}\hat{\mathsf{Y}}\hat{\mathsf{H}}^\dagger = -\hat{\mathsf{Y}}, \tag{4.17}$$

$$\hat{\mathsf{H}}\hat{\mathsf{Z}}\hat{\mathsf{H}}^\dagger = \hat{\mathsf{X}}. \tag{4.18}$$

Another gate, the $\hat{\mathsf{S}}$ gate, rotates the Bloch-vector $\pi/2$ radians around the z-axis. States along the y direction, gets rotated into the x direction, so that in a similar fashion to the Hadamard gate, the $\hat{\mathsf{S}}$ gate transforms basis states of $\hat{\mathsf{Y}}$ into basis states of $\hat{\mathsf{X}}$:

$$\hat{\mathsf{S}} = |0\rangle\langle 0| + i\,|1\rangle\langle 1| = |+i\rangle\langle +| + |-i\rangle\langle -| \,. \tag{4.19}$$

Here $|\pm i\rangle = 1/\sqrt{2}(|0\rangle \pm i\,|1\rangle)$, are the eigenstates of $\hat{\mathsf{Y}}$, with equivalent eigenvalues to $\hat{\mathsf{Z}}$ and $\hat{\mathsf{X}}$. The conjugation of $\hat{\mathsf{S}}$ on $\hat{\mathsf{X}}$, $\hat{\mathsf{Y}}$ and $\hat{\mathsf{Z}}$ are therefore:

$$\hat{\mathsf{S}}\hat{\mathsf{X}}\hat{\mathsf{S}}^\dagger = \hat{\mathsf{Y}}, \tag{4.20}$$

$$\hat{\mathsf{S}}\hat{\mathsf{Y}}\hat{\mathsf{S}}^\dagger = -\hat{\mathsf{X}}, \tag{4.21}$$

$$\hat{\mathsf{S}}\hat{\mathsf{Z}}\hat{\mathsf{S}}^\dagger = \hat{\mathsf{Z}}. \tag{4.22}$$

These Clifford gates also alters the angles about which rotations of the non-Clifford gates performs. To understand how, consider the conjugation of any

unitary operator $\hat{U}$ on some arbitrary hermitian matrix $\hat{H}$, then raising the conjugation to any power of k, can be expressed in a very straightforward way:

$$\left(\hat{U}\hat{H}\hat{U}^\dagger\right)^k = \hat{U}\hat{H}\hat{U}^\dagger\hat{U}\hat{H}\hat{U}^\dagger \ldots \hat{U}\hat{H}\hat{U}^\dagger = \hat{U}\hat{H}^k\hat{U}^\dagger. \tag{4.23}$$

With the acquired understanding of Eq. (4.23), it is now possible to work out the conjugation of Clifford gates, onto the rotational gates:

$$\begin{aligned}\hat{U}\hat{R}_j(\theta_j)\hat{U}^\dagger = \hat{U}e^{i\frac{\theta_j}{2}\hat{\sigma}_j}\hat{U}^\dagger &= \hat{U}\left[\sum_k^\infty \left(i\frac{\theta_j}{2}\right)^k \frac{\hat{\sigma}_j^k}{k!}\right]\hat{U}^\dagger \\ &= \sum_k \left(i\frac{\theta_j}{2}\right)^j \frac{\left(\hat{U}\hat{\sigma}_j\hat{U}^\dagger\right)^k}{k!} = e^{i\frac{\theta}{2}\hat{U}\hat{\sigma}_j\hat{U}^\dagger}.\end{aligned} \tag{4.24}$$

Substituting $\hat{U}$ with any of the Clifford gates, transforms rotations, as described by equations (4.16)-(4.18) and equations (4.20)-(4.22). Eq. (4.24) demonstrates, that it is sufficient for a universal set of gates, to just include a single rotational matrix, in addition to the Clifford gates. Fig. 6 illustrates the Bloch sphere transformation of the discussed Clifford gates, in addition to the non-Clifford, $\hat{\mathsf{T}}$ gate.

4.2.4 Implementation of single qubit gates

The implementations of non-Clifford gates in Superconducting qubits, primarily fall in either of two categories. This section shall focus on the case, when the qubit is in capacitive coupling to a microwave drive line.

By applying Kirchoff's law to formulate the Lagrangian, the Hamiltonian for this coupling can be derived. Subsequently, by restricting the analysis to the lowest two energy transitions, a Hamiltonian of the following form is obtained[24, p. 28] [25, p. 187-190]:

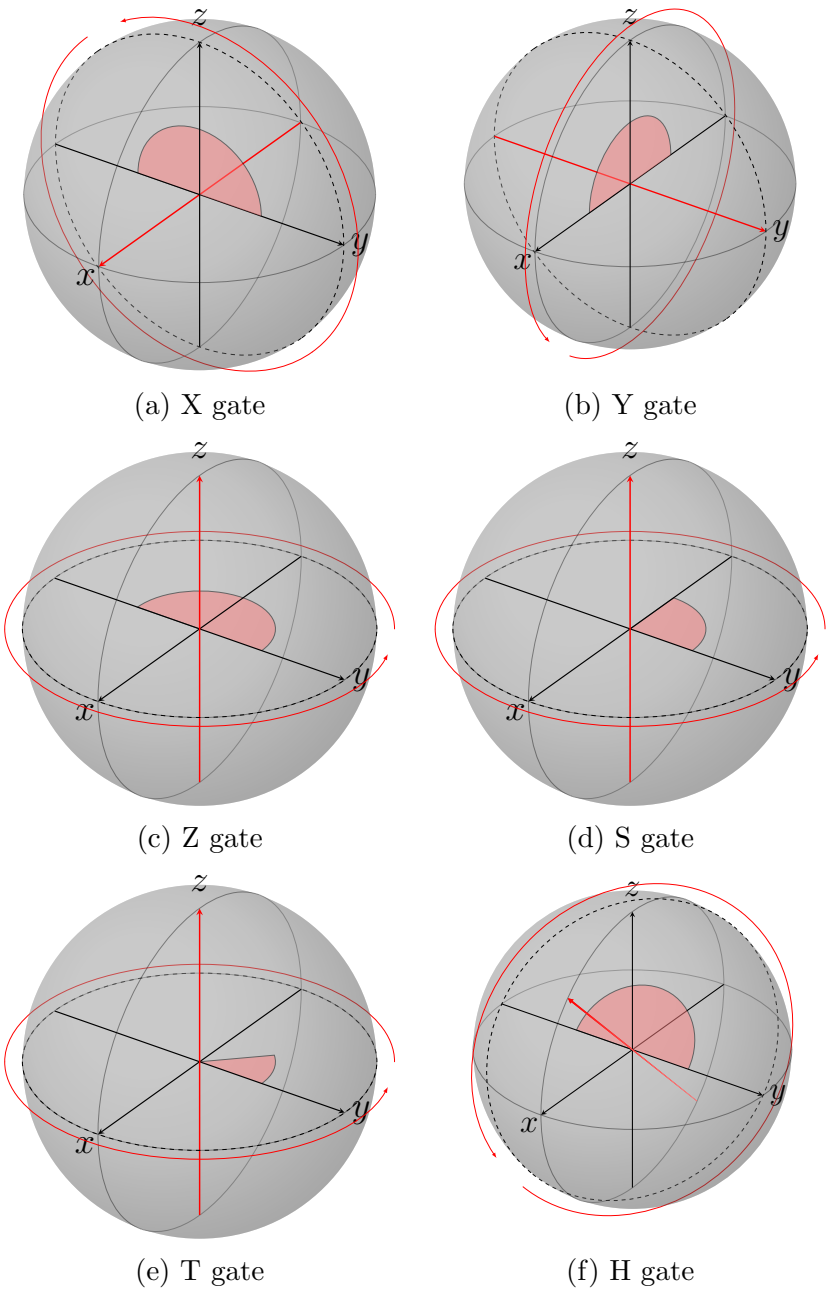
$$\hat{H} = -\frac{\omega_q}{2}\hat{\sigma}_z + \Omega V_d(t)\hat{\sigma}_y. \tag{4.25}$$

Here $\Omega = Q_{zpf}/(1 + (C/C_d))$, where Q_{zpf} is the zero point fluctuation of the charge, C is the capacitance on the capacitor and C_d is the coupling capacitance. Additionally, $V_d(t)$ denotes the time dependent voltage. Transforming the Hamiltonian into the interaction frame, also called the rotating frame, by performing the unitary operation $U_{int} = \exp(-i\omega_q\hat{\sigma}_z/2)$. Written in this frame, the driving term of Eq. (4.25) is written as:

$$\hat{H}_{d,RF} = U_{RF}^\dagger\hat{H}_d U_{RF} = \Omega V_d(t)\left(\cos(\omega_q t)\hat{\sigma}_y - \sin(\omega_q t)\hat{\sigma}_x\right) \tag{4.26}$$

Assuming that the voltage takes a sinusoidal form with some amplitude $V_0 e(t)$, called the envelope. The generic form is obtained:

$$V_d(t) = V_0 e(t)\sin(\omega_d t + \phi) = e(t)\left(\cos(\phi)\sin(\omega_d t) + \sin(\phi)\cos(\omega_d t)\right) \tag{4.27}$$



(a) X gate (b) Y gate

(c) Z gate (d) S gate

(e) T gate (f) H gate

Figure 6: Representations of various gate operations on the Bloch sphere are illustrated. (a) The $\hat{\mathsf{X}}$ gate rotates the qubit π radians in the yz-plane of the Bloch sphere. (b) $\hat{\mathsf{Y}}$ induces a rotation of π radians in the xz-plane. (c)-(e) Operations involving the $\hat{\mathsf{Z}}$, $\hat{\mathsf{S}}$ and $\hat{\mathsf{T}}$, performs rotations within the xy-plane, by π, $\pi/2$ and $\pi/4$ radians, respectively. It is important to mention that the $\hat{\mathsf{T}}$ gate does not fall under the category of a Clifford gate. (f) The Haddamard gate rotates the qubit π radians about an axis diagonal in the xz-plane.

where ω_d is drive frequency. Adopting the in-phase, $I = \cos(\phi)$, and out of phase, $Q = \sin(\phi)$ components, and applying the proper trigonometric identities, the drive Hamiltonian can be expressed in the rotating frame as:

$$\begin{aligned}\hat{H}_{d,RF} = \frac{1}{2}\Omega V_0 e(t)\Big[& I\left(\cos((\omega_q+\omega_d)t) - \cos((\omega_q-\omega_d)t)\right)\hat{\sigma}_x \\ & Q\left(\sin((\omega_q+\omega_d)t) + \sin((\omega_q-\omega_d)t)\right)\hat{\sigma}_x \\ & -I\left(\sin((\omega_q+\omega_d)t) - \sin((\omega_q-\omega_d)t)\right)\hat{\sigma}_y \\ & -Q\left(\cos((\omega_q+\omega_d)t) + \cos((\omega_q-\omega_d)t)\right)\hat{\sigma}_y\Big]\end{aligned} \tag{4.28}$$

Applying the rotating wave approximation, eliminates rapidly oscillating terms, $\omega_q+\omega_d$, as they will average to zero in time spans of the system evolution. The detuning terms, $\Delta\omega = \omega_q - \omega_d$, are kept, and the Hamiltonian is expressed as:

$$\begin{aligned}\hat{H}_{d,RF} = \frac{1}{2}\Omega V_0 s(t)\Big[& -I\cos(\Delta\omega t)\hat{\sigma}_x + Q\sin(\Delta\omega t)\hat{\sigma}_x \\ & +I\sin(\Delta\omega t)\hat{\sigma}_y - Q\cos(\Delta\omega t)\hat{\sigma}_y\Big]\end{aligned} \tag{4.29}$$

In order to comprehend the implications of Eq. (4.29), it is helpful to examine the situation where the drive frequency is tuned to the qubit resonance frequency. This tuning simplifies the Hamiltonian:

$$\hat{H}_{d,RF} = -\frac{\Omega}{2}V_0 s(t)\left(I\hat{\sigma}_x + Q\hat{\sigma}_y\right). \tag{4.30}$$

The Hamiltonian is then controlled, by adjusting the phase ϕ. Qubit state evolution taking place between times $t = 0$ and a later time t is described by the unitary operation:

$$\hat{U}(t) = e^{-i\int_0^t \hat{H}_d \mathrm{d}\tau} = \exp\left(\frac{i}{2}\Omega V_0\left(I\hat{\sigma}_x + Q\hat{\sigma}_y\right)\int_0^t e(\tau)\mathrm{d}\tau\right). \tag{4.31}$$

A phase $\phi = 0$ yields a $\hat{R}_x(\theta_x)$ gate, while $\phi = \pi/2$ leads to a $\hat{R}_y(\theta_y)$ gate. From Eq. (4.31), the rotational angle is given by:

$$\theta = -\int_0^t \Omega V_0 e(\tau)\mathrm{d}\tau. \tag{4.32}$$

For the purpose of simulation, it can be assumed that the envelope formula is a square wave with normalized pulse amplitude, such that rotating the qubit π radians around the x axis in the Bloch sphere, is simply to execute the gate for a duration of $t = \pi/(\Omega V_0)$.

4.3 Gates between multiple qubits

4.3.1 Universality of two qubit quantum gates

In the case of multiple qubits, it is necessary to implement gates that can introduce entanglements between them. The generators for this algebra, is

CNOT	
Input	Output
$\|00\rangle$	$\|00\rangle$
$\|01\rangle$	$\|01\rangle$
$\|10\rangle$	$\|11\rangle$
$\|11\rangle$	$\|10\rangle$

CZ	
Input	Output
$\|00\rangle$	$\|00\rangle$
$\|01\rangle$	$\|01\rangle$
$\|10\rangle$	$\|10\rangle$
$\|11\rangle$	-$\|11\rangle$

Table 2: Truth table of the CNOT and the CZ gates, where the first bit represents the control bit, and the second is the target, i.e. the input state is $|\psi\rangle = |q_{control}\rangle \otimes |q_{target}\rangle$.

the generators of the $SU(4)$ group. This is given by:

$$T_{i+3(j-1)} = \hat{\sigma}_i \otimes \hat{\sigma}_j, \tag{4.33}$$

$$T_{i+9} = \hat{\sigma}_i \otimes I, \tag{4.34}$$

$$T_{i+12} = I \otimes \hat{\sigma}_i. \tag{4.35}$$

That is fifteen in total, where $i, j = 1, 2, 3$, such that hsi represents the Pauli matrices. T_{10} to T_{15} represents the generators operating only on the first or the second qubit, while T_1 to T_9 generates entanglements. The Hamiltonian working on the qubit Hilbert space, is then a linear combination of the generators, and so the unitary operators furnishing $SU(4)$, written in a general form, is given by:

$$\hat{U} = e^{i \sum_k a_k T_k} \tag{4.36}$$

A two qubit unitary gate of theoretical interest, is that of the Clifford gate typically referred to as the controlled not gate, CNOT, also called the controlled $\hat{\mathsf{X}}$ gate, CX. One qubit is assigned as control, the another as target, the unitary gate inverts the target qubit, for the branches of the state vector, where the control bit has the value 1. The truth table for the gate is displayed in Table 2.

The CNOT gate operation, in the matrix representation, is a Hermitian 4×4 matrix of the form:

$$\hat{\mathcal{C}}_{\mathsf{X}} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \tag{4.37}$$

Or written in tensor notation, using the $\hat{\mathsf{X}}$ gate, and the basis vectors, this gate can be written as:

$$\hat{\mathcal{C}}_{\mathsf{X}} = |0\rangle\langle 0| \otimes I + |1\rangle\langle 1| \otimes \hat{\mathsf{X}} \tag{4.38}$$

In this notation, the behavior of the truth table becomes apparent. The interesting property of the gate, is how it transforms the single qubit X Clifford gate, i.e.

$$\hat{\mathcal{C}}_{\mathsf{X}}(I \otimes \hat{\mathsf{X}})\hat{\mathcal{C}}_{\mathsf{X}}^{\dagger} = \hat{\mathcal{C}}_{\mathsf{X}}(|1\rangle\langle 0| \otimes I + |0\rangle\langle 1| \otimes X) \tag{4.39}$$

$$= |0\rangle\langle 1| \otimes \hat{\mathsf{X}} + |1\rangle\langle 0| \otimes \hat{\mathsf{X}} = \hat{\mathsf{X}} \otimes \hat{\mathsf{X}} \tag{4.40}$$

That is, the CNOT gate transforms the generator T_{13}, into the generator T_1. From T_1 it is straightforward, using the Clifford gates, to construct the remaining generators. For example, suppose one is interested in the rotation $\exp\{iaX \otimes Z\}$, simply conjugate the $\exp\{iaX \otimes X\}$ by $\hat{H}$ on the second qubit:

$$(I \otimes \hat{H})e^{ia\hat{\mathsf{X}}\otimes\hat{\mathsf{X}}}(I \otimes \hat{H}^{\dagger}) = e^{ia\hat{\mathsf{X}}\otimes\hat{\mathsf{Z}}} \tag{4.41}$$

The approximation for small angle rotations in $SU(2)$, is similarly true for the generators of $SU(4)$, simply replace the generators of $SU(2)$ in Eq. (4.9), with the generators $\hat{T}_1$ to $\hat{T}_{16}$:

$$\hat{U} = e^{i\sum_k^{16}(a_k\hat{T}_k/n)} \approx \prod_k^{16} e^{ia_k\hat{T}_k/n}. \tag{4.42}$$

4.3.2 Implementation of two qubit gates

The CNOT gate, although theoretically elegant, is not typically the preferred gate for implementation in superconducting circuits. It should be recalled that qubits are capacitively coupled to a resonator, which mediates the interaction between the qubits. A method for achieving two-qubit gates involves utilizing this coupling while temporarily permitting the circuit to access higher energy states. The actual dynamics of the system are characterized by a Hamiltonian of the form, as seen in Sec. 2.2:

$$\hat{H} = \omega_r\hat{a}_r^{\dagger}\hat{a}_r + \sum_s \left(\omega_s\hat{a}_s^{\dagger}\hat{a}_s + \frac{\delta_s}{2}\hat{a}_s^{\dagger}\hat{a}_s^{\dagger}\hat{a}_s\hat{a}_s + g_s\left(\hat{a}_s^{\dagger}\hat{a}_r + \hat{a}_s\hat{a}_r^{\dagger}\right)\right) \tag{4.43}$$

By restricting Eq.(2.21) to the lowest three energy states and denoting $E_n^{(q)}$ as the energy of the n'th state in the q'th qubit, the additional state modifies the Jaynes-Cummings Hamiltonian for the two-qubit system to:

$$\begin{aligned}\hat{H} &= \omega_r\hat{a}_r^{\dagger}\hat{a}_r + \sum_{n=0}^{2}\left(E_n^{(1)}|n\rangle\langle n| \otimes I + I \otimes E_n^{(2)}|n\rangle\langle n|\right) + \sum_s g_k\left(\hat{a}_s^{\dagger}\hat{a}_r + \hat{a}_s\hat{a}_r\right) \\ &= \omega_r\hat{a}_r^{\dagger}\hat{a}_r + \sum_{n=0}^{2}\sum_{m=0}^{2}E_{nm}|nm\rangle\langle nm| + \sum_s g_k\left(\hat{a}_s^{\dagger}\hat{a}_r + \hat{a}_s\hat{a}_r\right),\end{aligned} \tag{4.44}$$

where $E_{nm} = E_n^{(1)} + E_n^{(2)}$. The second line of Eq.(4.44) arises due to the property that $\sum_n |n\rangle\langle n| \otimes I = \sum_n |n\rangle\langle n| \otimes \sum_m |m\rangle\langle m| = \sum_n |n\rangle \sum_m |nm\rangle\langle nm|$. By limiting the interaction term to only allow energy exchanges that conserve the total excitation of the transmon circuit, the states of $|02\rangle$ and $|20\rangle$ can be employed to drive phase transitions, conditioned on the state of $|11\rangle$.

In order to analyze this dynamic, a simpler scenario can be considered, wherein only the second qubit is allowed to access the second excited state. It can be demonstrated that the Hamiltonian of the system can be approximated as:

$$\begin{aligned}\hat{H} = \omega_r a^\dagger a + \sum_s^2 \left(\tilde{\omega}_s \frac{\hat{\sigma}_z^{(s)}}{2} + \chi_s \hat{a}^\dagger \hat{a} \hat{\sigma}_z^{(s)} + S\left(\chi_s, g_s\right) \right) \\ + \omega_{02} |02\rangle\langle 02| + J_{12} \left(|11\rangle\langle 02| + |02\rangle\langle 11| \right),\end{aligned} \tag{4.45}$$

The approximation is valid when the system resides in the dispersive regime, with further details provided in Sec. 5.1. Particular attention should be given to the second line of the expression. The $\omega_{02} |02\rangle\langle 02|$ induces a time-dependent phase on the $|02\rangle$ state, and a coupling of J_{12} drives transitions between $|11\rangle$ and $|02\rangle$.

Consider a pulse targeting this transition between the two states $|11\rangle \leftrightarrow |02\rangle$ and implementing a population inversion between them. Subsequently, the system idles for a period before another pulse reverses the population inversion of the first. During the idling period, a phase shift is induced, enabling the operation of a controlled phase gate. By choosing the appropriate pulse and idling duration, ensuring that the proper leakage and phase conditions are satisfied, it is possible to utilize this procedure to perform a controlled $\hat{\mathsf{Z}}$ gate, which possesses the following matrix representation:

$$\hat{\mathcal{C}}_{\mathsf{Z}} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \end{pmatrix}. \tag{4.46}$$

The CZ gate operation can be transformed into a CNOT gate operation, by conjugation of Haddamard gates on the target qubit:

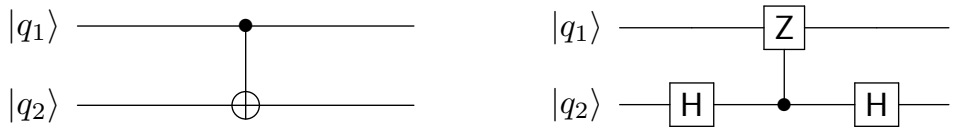
$$(I \otimes \hat{\mathsf{H}}) \hat{\mathcal{C}}_{\mathsf{Z}} (I \otimes \hat{\mathsf{H}}) = I \otimes \hat{\mathsf{H}} (|0\rangle\langle 0| \otimes I + |1\rangle\langle 1| \otimes \hat{\mathsf{Z}}) I \otimes \hat{\mathsf{H}} \tag{4.47}$$

$$= (|0\rangle\langle 0| \otimes I + |1\rangle\langle 1| \otimes \hat{\mathsf{X}}) = \hat{\mathcal{C}}_{\mathsf{X}} \tag{4.48}$$

This transformation is also depicted in a quantum circuit diagram in Fig. 7.

4.3.3 Fidelity of gates

Leakage, cross talking, calibration and decoherence[26][27] are among some of the great challenges in realizing fault tolerant quantum gates. A measure is required to describe how accurately the implementation of a unitary



(a) CNOT gate. (b) Effective CNOT gate.

Figure 7: A quantum circuit diagram of the Hadamard gate is presented. Quantum circuit diagrams share similarities with classical circuit diagrams; however, due to the reversibility of quantum gates, no multi-qubit gates terminate a qubit line, resulting in quantum circuit diagrams being drawn as parallel lines with gates operating between them. (a) Depicts a direct CNOT gate implementation, where the thick dot represents the control qubit and the circled plus denotes the target qubit. (b) Displays an effective CNOT gate operation, where the boxed H represent the Hadamard operation and the boxed Z, connected to a control qubit, represents the CZ gate operation.

gate matches the intended operation, i.e. the extent to which the performed computation of single gates deviates from the theoretical results. Fidelity quantifies how well the final state approximates the intended target state, in other words the closeness of the two quantum states. Completely distinguishable states, such as as pair of orthogonal states, has the fidelity of zero, while two identical states results in a fidelity of one.

Mathematically, fidelity is defined as the squared Bhattacharyya coefficients. Suppose two mixed states are given, with a probability p_k of being in a state $|\psi_k\rangle$ and a probability q_k of being in a state $|\varphi\rangle$. From the definition of the density matrix, p_k and q_k can also be considered as eigenvalues for their respective density matrices $\rho_1 = \sum_k p_k \, |\psi_k\rangle\langle\psi_k|$ and $\rho_2 = \sum_k q_k \, |\varphi_k\rangle\langle\varphi_k|$, where $|\psi_k\rangle$ and $|\varphi_k\rangle$ are the corresponding eigenvectors. The fidelity between the two systems is then given by:

$$F(\rho_1, \rho_2) = \left(\sum_k \sqrt{p_k q_k} \right)^2 = \left(\mathrm{Tr}(\sqrt{\rho_1 \rho_2}) \right)^2 . \tag{4.49}$$

If both states in question are pure state matrices, then Eq. (4.49) has a simple reduction, and the fidelity of the two states can deduced directly from the inner product squared of the state vectors:

$$\left(\mathrm{Tr}(\sqrt{\rho_1 \rho_2}) \right)^2 = \left(\mathrm{Tr}\left(\sqrt{|\psi\rangle\langle\psi|\varphi\rangle\langle\varphi|} \right) \right)^2 = |\langle\psi|\varphi\rangle|^2. \tag{4.50}$$

To reach fault tolerant quantum computers, the fidelity for individual gates must be very close to unity. For most quantum algorithms, when scaled to problems that are intractable on classical hardware, the fidelity of individual gates must lie well above the 0.99 threshold to demonstrate quantum supremacy. This arises from the fact that errors propagate in entangled pairs,

and even small errors on individual qubits can, for sufficiently deep layers of quantum circuits, propagate the errors throughout the entire system[28].

4.4 No cloning theorem

Simulations allow for the initialization of registers to any value, including the inference of previously attained states from a stochastic quantum trajectory. This capability arises from the fact that system evolution can be monitored during simulation run time by accessing computer memory. In contrast, in reality, due to the wave function collapse upon measurement, the qubit state is only read out at the end of computation. Consequently, its state before being projected onto the basis states remains an unknown superposition.

If cloning the qubit state during the computation process were possible, acquiring this knowledge without disrupting the computation of the main set of qubits could be feasible. However, due to the unitary time evolution of quantum states, creating an identical copy of an arbitrarily unknown quantum state is rendered impossible.

In order to prove the no-cloning theorem, consider the scenario where the objective is to clone a state $|\phi_A\rangle$ onto a blank state $|\chi\rangle$, analogous to a printer cloning a painting onto a blank canvas. The mechanism employed should implement a unitary operator, denoted as $\hat{U}$, which transforms the *blank* state $|\chi\rangle$ into the desired cloned state, $|\psi_A\rangle$. If $|\phi_A\rangle$ is truly arbitrary, then the unitary operator should also be capable of cloning any other arbitrary state, $|\phi_B\rangle$ onto $|\chi\rangle$ as well. In other words, the unitary operation should perform the following transformations:

$$\hat{U}\,|\phi_A\rangle\otimes|\chi\rangle = |\phi_A\rangle\otimes|\phi_A\rangle \tag{4.51}$$

$$\hat{U}\,|\phi_B\rangle\otimes|\chi\rangle = |\phi_B\rangle\otimes|\phi_B\rangle\,. \tag{4.52}$$

In the present work, no assumptions have been made regarding the shape of $\hat{U}$. However, it can be demonstrated that, in order to satisfy the given expressions, $|\phi_A\rangle$ and $|\phi_B\rangle$ cannot be considered as two arbitrary states, but must instead be related to each other. To explore this relationship, the inner products between the two states are calculated. Initially, the left-hand side of the equations is examined by taking the bra of (4.51) and applying it to the ket of (4.52), followed by determining the norm of the resulting expression:

$$\begin{aligned}\left|\langle\phi_A|\otimes\langle\chi|\,\hat{U}^\dagger\hat{U}\,|\phi_B\rangle\otimes|\chi\rangle\right| &= |(\langle\phi_A|\otimes\langle\chi|)\,(|\phi_B\rangle\otimes|\chi\rangle)| \\ &= |\langle\phi_A|\phi_B\rangle\otimes\langle\chi|\chi\rangle| \\ &= |\langle\phi_A|\phi_B\rangle|.\end{aligned} \tag{4.53}$$

The same is done on the right-hand side of the equation:

$$|(\langle\phi_A|\otimes\langle\phi_B|)\,(|\phi_B\rangle\otimes|\phi_B\rangle)| = |\langle\phi_A|\phi_B\rangle\otimes\langle\phi_A|\phi_B\rangle| = |\langle\phi_A|\phi_B\rangle|^2, \tag{4.54}$$

Upon combining the two sites, it is found that $|\langle\phi_A|\phi_B\rangle| = |\langle\phi_A|\phi_B\rangle|^2$, yielding the solutions $|\langle\phi_A|\phi_B\rangle| = 0, 1$. This result implies that the two states must belong to the same orthonormal basis set, i.e., either $|\phi_A\rangle \perp |\phi_B\rangle$ or $|\phi_A\rangle = |\phi_B\rangle$. Consequently, it is not possible to clone a general state onto another state. However, if the classical operational regime is considered, where qubits function as bits without superposition, then $|0\rangle$ and $|1\rangle$ belong to the same orthonormal basis set, and cloning of the information onto the state $ket0$ can be performed using the CNOT gate.

5 Measurement Theory and Readout

5.1 Dispersive Regime

For optimal performance, a quantum computer must have the ability to perform a fast and accurate readout of its qubits, as this is crucial for extracting meaningful information from quantum computations and ensuring the reliability of the results obtained. Consequently, it is vital that the act of measuring does not inadvertently and irreversibly alter the states of the output; as a result, a quantum non-demolition (QND) readout is desired. The accuracy of the readout is contingent upon the regime in which the resonator is driven, for instance, when $\Delta_s \ll g_s$, the system coherently swaps excitations between the qubits and the resonator system. These interactions alter qubit population, and excitation of one qubit can uncontrollably be exchanged with the excitation of another through the resonator as a mediator. Therefore, qubit readout occurs in the regimes where the resonator is largely detuned from the qubits, such that $\Delta_s \gg g_s$, and no energy is exchanged between the resonator and the qubits. This is referred to as the dispersive regime, as interactions between the transmon qubits and the resonator system are considered dispersive. Since g_s is low in comparison to the other frequencies, the interaction term can effectively be regarded as a perturbation term. Starting from Eq (3.13), rewritten in units of radiant frequencies, the non-perturbed and perturbed Hamiltonian terms are provided by:

$$\hat{H}_0 = \omega_r \hat{a}^\dagger \hat{a} + \sum_s \omega_s \frac{\hat{\sigma}_z^{(s)}}{2}, \tag{5.1}$$

$$\hat{H}_{int} = \sum_s g_s \left(\hat{a}^\dagger \hat{\sigma}_-^{(s)} + \hat{a} \hat{\sigma}_+^{(s)} \right), \tag{5.2}$$

where evidently $\hat{H} = \hbar\hat{H}_0 + \hbar\hat{H}_{int}$. To analyze the behavior of the dispersive regime, the Hamiltonian is simplified, by changing reference frame into a basis which eliminates the first order interaction term. This change of basis is achieved by applying the Schrieffer-Wolff transformation to diagonalize the Hamiltonian, rotating it with an unitary operator $U = \exp(\hat{\eta})$. The frame

which accomplishes this, is a choice for $\hat{\eta}$, such that $[\hat{\eta}, \hat{H}_0] = -\hat{H}_{int}$, eliminating $\hat{H}_{int}$ in the series expansion. The correct choice of $\hat{\eta}$, satisfying the stated condition, is given by; $\hat{\eta} = \sum_s \frac{g_s}{\Delta_s}(\hat{a}^\dagger \hat{\sigma}_-^{(s)} - \hat{a}\hat{\sigma}_+^{(s)})$. To demonstrate this, it is useful to find the commutation relations of these intermediate terms,

$$\left[\hat{\eta}, \hat{a}^\dagger \hat{a}\right] = -\sum_s \frac{g_s}{\Delta_s}\left(\hat{a}^\dagger \hat{\sigma}_-^{(s)} + \hat{a}\hat{\sigma}_+^{(s)}\right) \tag{5.3}$$

$$\left[\hat{\eta}, \hat{\sigma}_z^{(s)}\right] = -2\sum_s \frac{g_s}{\Delta_s}\left(\hat{a}^\dagger \hat{\sigma}_-^{(s)} + \hat{a}\hat{\sigma}_+^{(s)}\right), \tag{5.4}$$

where the commutation relations of the equations, (2.14), (3.11) and (3.12) were applied. The spin matrices of each distinct qubit commutes, so the two summations in both expressions of $\hat{\eta}$ and $\hat{H}_0$, is reduced to a single sum. The commutation of $\hat{\eta}$ and $\hat{H}_0$ is therefore shown by substitution, to satisfy the desired condition:

$$\begin{aligned}[\hat{\eta}, \hat{H}_0] &= \left[\hat{\eta}, \omega_r \hat{a}^\dagger \hat{a}\right] - \left[\hat{\eta}, \omega_s \frac{\hat{\sigma}_z^{(s)}}{2}\right] \\ &= -\sum_s \frac{g_s}{\Delta_s}\left(\hat{a}^\dagger \hat{\sigma}_-^{(s)} + \hat{a}\hat{\sigma}_+^{(s)}\right)(\omega_r - \omega_s) = -\hat{H}_{int}\end{aligned} \tag{5.5}$$

The Hamiltonian $\hat{H}$ is diagonalized by the unitary operator $\exp(\hat{\eta})$. The new Hamiltonian gets Taylor expanded, $\hat{H}_{disp} = e^{\hat{\eta}}\hat{H}e^{-\hat{\eta}}$, around $\hat{\eta}$, such that first orders in $\hat{H}_{int}$ cancel out, while higher-order terms persist:

$$\begin{aligned}\hat{H}_{disp} = e^{\hat{\eta}}\hat{H}e^{-\hat{\eta}} &= \hat{H} + \left[\hat{\eta}, \hat{H}\right] + \frac{1}{2}\left[\hat{\eta}, \left[\hat{\eta}, \hat{H}\right]\right] + \mathcal{O}(\hat{\eta}^3) \\ &= \hat{H}_0 + \frac{1}{2}\left[\hat{\eta}, \hat{H}_{int}\right] + \mathcal{O}(\hat{H}_{int}^3).\end{aligned} \tag{5.6}$$

In this basis, it is said that the qubit becomes dressed by the field. When disregarding terms of $\hat{H}_{int}^3$ and higher orders, the dispersive Hamiltonian becomes approximately determined by the non-perturbed Hamiltonian, $\hat{H}_0$, in

addition to the commutator term, $\left[\hat{\eta}, \hat{H}_{int}\right]$, which is derived to be:

$$\begin{aligned}
\left[\hat{\eta}, \hat{H}_{int}\right] &= \left[\sum_s \frac{g_s}{\Delta_s}(\hat{a}^\dagger\hat{\sigma}_-^{(s)} - \hat{a}\hat{\sigma}_+^{(s)}), \sum_q g_q(\hat{a}^\dagger\hat{\sigma}_-^{(q)} + \hat{a}\hat{\sigma}_+^{(q)})\right] \\
&= \left[\sum_s \frac{g_s}{\Delta_s}\hat{a}^\dagger\hat{\sigma}_-^{(s)}, \sum_q g_q\hat{a}^\dagger\hat{\sigma}_-^{(q)}\right] + \left[\sum_s \frac{g_s}{\Delta_s}\hat{a}^\dagger\hat{\sigma}_-^{(s)}, \sum_q g_q\hat{a}\hat{\sigma}_+^{(q)}\right] \\
&- \left[\sum_s \frac{g_s}{\Delta_s}\hat{a}\hat{\sigma}_+^{(s)}, \sum_q g_q\hat{a}^\dagger\hat{\sigma}_-^{(q)}\right] - \left[\sum_s \frac{g_s}{\Delta_s}\hat{a}\hat{\sigma}_+^{(s)}, \sum_q g_q\hat{a}\hat{\sigma}_+^{(q)}\right] \\
&= \hat{a}^\dagger\hat{a}\sum_s\sum_q \frac{g_sg_q}{\Delta_s}\hat{\sigma}_-^{(s)}\hat{\sigma}_+^{(q)} - \hat{a}\hat{a}^\dagger\sum_q\sum_s \frac{g_qg_s}{\Delta_s}\hat{\sigma}_+^{(q)}\hat{\sigma}_-^{(s)} \\
&- \hat{a}\hat{a}^\dagger\sum_s\sum_q \frac{g_sg_q}{\Delta_s}\hat{\sigma}_+^{(s)}\hat{\sigma}_-^{(q)} + \hat{a}^\dagger\hat{a}\sum_q\sum_s \frac{g_qg_s}{\Delta_s}\hat{\sigma}_-^{(q)}\hat{\sigma}_+^{(s)} \\
&= -\hat{a}^\dagger\hat{a}\Big(\sum_s\sum_q \frac{g_sg_q}{\Delta_s}\hat{\sigma}_+^{(s)}\hat{\sigma}_-^{(q)} - \sum_q\sum_s \frac{g_qg_s}{\Delta_s}\hat{\sigma}_-^{(q)}\hat{\sigma}_+^{(s)} \\
&+ \sum_q\sum_s \frac{g_qg_s}{\Delta_s}\hat{\sigma}_+^{(q)}\hat{\sigma}_-^{(s)} - \sum_s\sum_q \frac{g_sg_q}{\Delta_s}\hat{\sigma}_-^{(s)}\hat{\sigma}_+^{(q)}\Big) \\
&- \sum_s\sum_q \frac{g_sg_q}{\Delta_s}\hat{\sigma}_+^{(s)}\hat{\sigma}_-^{(q)} - \sum_q\sum_s \frac{g_qg_s}{\Delta_s}\hat{\sigma}_+^{(q)}\hat{\sigma}_-^{(s)} \\
&= -\hat{a}^\dagger\hat{a}\left(\left[\sum_s \frac{g_s}{\Delta_s}\hat{\sigma}_+^{(s)}, \sum_q g_q\hat{\sigma}_-^{(q)}\right] + \left[\sum_q g_q\hat{\sigma}_+^{(q)}, \sum_s \frac{g_s}{\Delta_s}\hat{\sigma}_-^{(s)}\right]\right) \\
&- \sum_s \frac{g_s^2}{\Delta_s}\left(\hat{\sigma}_z^{(s)} + I\right) - \sum_s\sum_{q\neq s} \frac{g_sg_q}{\Delta_s}\hat{\sigma}_+^{(s)}\hat{\sigma}_-^{(q)} - \sum_q\sum_{s\neq q} \frac{g_qg_s}{\Delta_s}\hat{\sigma}_+^{(q)}\hat{\sigma}_-^{(s)} \\
&= \sum_s\left(2\chi_s\hat{\sigma}_z^{(s)}\hat{a}^\dagger\hat{a} + \chi_s\left(\hat{\sigma}_z^{(s)} + I\right) + \sum_{q\neq s}\left(\frac{\chi_sg_q}{g_s} + \frac{\chi_qg_s}{g_q}\right)\hat{\sigma}_+^{(s)}\hat{\sigma}_-^{(q)}\right)
\end{aligned} \tag{5.7}$$

Alternatively, the expression can be repackaged in a more simplified form:

$$\left[\hat{\eta}, \hat{H}_{int}\right] = \sum_s\left(2\chi_s\hat{\sigma}_z^{(s)}\hat{a}^\dagger\hat{a} + \chi_s\left(\hat{\sigma}_z^{(s)} + I\right) + 2S\left(\chi_s, g_s\right)\right) \tag{5.8}$$

where $\chi_s = -g_s^2/\Delta_s$ denotes the dispersive shift, which depends on the state of the qubit. Meanwhile, $S\left(\chi_s, g_s\right)$ represents a flip-flop term, exchanging population between different qubits, and is given by:

$$S\left(\chi_s, g_s\right) = \sum_{q\neq s}\left(\frac{\chi_sg_q}{2g_s} + \frac{\chi_qg_s}{2g_q}\right)\hat{\sigma}_+^{(s)}\hat{\sigma}_-^{(q)} = \sum_{q\neq s} J_{sq}\hat{\sigma}_+^{(s)}\hat{\sigma}_-^{(q)}. \tag{5.9}$$

Injecting (5.8) into (5.6) gives the system's Hamiltonian in the dispersive regime:

$$\hat{H}_{disp} = \omega_r \hat{a}^\dagger \hat{a} + \sum_s \left(\tilde{\omega}_s \frac{\hat{\sigma}_z^{(s)}}{2} + \chi_s \hat{a}^\dagger \hat{a} \hat{\sigma}_z^{(s)} + S\left(\chi_s, g_s\right) \right), \tag{5.10}$$

where $\tilde{\omega}_s = \chi_s - \omega_s$ is called the Lamb shifted qubit frequency. Equation (5.10) also demonstrates, that the qubits picks up a state dependent frequency shift, that is proportional to the number of photons in the cavity, called the ac-Stark shift, originating from oscillating external fields causes slight dephasing of the qubit.

It is important to emphasize that the Hamiltonian presented in Eq. (5.10), is derived from the constraint of considering only the lowest two energy levels of the transmon circuit. While outside the scope of this thesis, these higher energy levels can be exploited, e.g. as a mechanism for creating two gates, as discussed in Sec. 4.3.2, or to implement shelving schemes, effectively increase readout time for logical $|1\rangle$ states[29]. In the case where these higher levels are not ignored, the dispersive shift gets modified to:

$$\chi_s = -\frac{g_s^2}{\Delta_s} \left(\frac{1}{1 + \Delta_s / (\omega_s^{2\to1} - \omega_s^{1\to0})} \right), \tag{5.11}$$

where $\omega_{i\to j}$ represents the transition frequency between states $|i\rangle$ and $|j\rangle$[30]. This indicates that the dispersive shift for Eq. (4.45) corresponds to the dispersive shift when considering the three levels of the transmon circuit.

5.2 I-Q mixing

In the previous section, it was demonstrated that, in the dispersive regime, as described by the Hamiltonian in Eq. (5.10), the resonator undergoes a qubit state-dependent frequency shift, $\chi_q \hat{a}^\dagger \hat{a} \hat{\sigma}_z^{(q)}$. The phase is determined by probing the resonator with a field, $\beta(t)$. After the field interacts with the resonator, the signal is mixed with a classical coherent field, known as the local oscillator. The mixing of the two signals occurs on a microwave I-Q mixer, which employs square-law-type diodes to multiply the signals together. By utilizing demodulation strategies, it becomes possible to identify the two components of the voltage.

At the end of Sec. 2.3 the transmon qubit, capacitively coupled to a resonator was compared to a spinor inside a microwave cavity. To help gain intuition, this chapter shall similarly adopt an optical analogous representation to the I-Q mixing. The first step is to consider the classical version of the outgoing electrical field. Such an electrical signal moving through a vacuum, is a classical wave electromagnetic wave form, and is expressed as:

$$E_{out}(t) = E_0 \cos(\omega_{out} t + \theta) = \mathrm{Re}(E_0 e^{i\omega_{out} t} e^{i\theta})). \tag{5.12}$$

ω_{out} is the frequency of the electrical wave, e.i. the carrier frequency. The field is here presented using complex analysis and has a time independent phase component. The global phase of the electrical field is not what is important. This is because that in addition to the qubit state induced phase shift, the signal also picks up a propogation phase, $\theta_r = -\mathbf{k} \cdot \mathbf{r}$ which is independent of the qubit state. Everything of interest for the readout of the qubit, is the ac-Stark shift, so ignoring the oscillating part of the field, the static phasor component is defined as:

$$c_{out} = E_0 e^{i\theta} = I + iQ. \tag{5.13}$$

As with the phase components for the implementation of single qubit gates in Sec. 4.2.4, the two components of the phasor consist of the in-phase component I, representing the real part, and the out-of-phase component Q, representing the imaginary part. By utilizing the complex conjugate, the two components can be separated into their separate parts:

$$I = \frac{1}{2}\left(c_{out} + c_{out}^*\right), \tag{5.14}$$

$$Q = \frac{1}{2i}\left(c_{out} - c_{out}^*\right). \tag{5.15}$$

By shifting from the classical case to the quantum case, implying that the phasor component must be expressed as a quantum operator $\hat{c}_{out}$. In the general case, the quadrature operators for the single mode of $\hat{c}_{out}$, is defined as

$$\hat{X}_\phi = \frac{1}{2}\left(\hat{c}_{out}e^{-i\phi} + \hat{c}_{out}^\dagger e^{i\phi}\right) \tag{5.16}$$

$$\hat{X}_{\phi+\pi/2} = \frac{1}{2i}\left(\hat{c}_{out}e^{-i\phi} - \hat{c}_{out}^\dagger e^{i\phi}\right). \tag{5.17}$$

Since the out-coupled field originates from the probing of the cavity, it suggests that $\hat{c}_{out} \propto \hat{a}$. Consequently, the commutation relations of the quadratures, $[\hat{X}\phi, \hat{X}\phi + \pi/2] \propto \frac{1}{4}[(\hat{a} + \hat{a}^\dagger), -i(\hat{a} - \hat{a}^\dagger)] = i/2$, lead to uncertainty relations equivalent to those found in Heisenberg's uncertainty between position and momentum. Besides some prefactors, one is free to interpret the two quadratures as a position, $\hat{X}_\phi = \hat{x}$ and momentum $\hat{X}_{\phi+\pi/2} = \hat{p}$ operators. The probability distribution of the vacuum state with respect to the quadrature operators is therefore just the probability distribution of the vacuum state of a harmonic osccilator, i.e. a Gaussian destribution[31, p. 44]:

$$|\langle x|0\rangle|^2 = \sqrt{\frac{2}{\pi}}e^{-2x(t)^2}, \tag{5.18}$$

$$|\langle p|0\rangle|^2 = \sqrt{\frac{2}{\pi}}e^{-2p(t)^2}. \tag{5.19}$$

The derivation of Eq.(5.19) is directly achieved through the application of a Fourier transformation to Eq.(5.18). As the transformation of a Gaussian is itself a Gaussian, it is observed that both quadratures possess analogous distributions.

5.2.1 Homodyne detection

The weak single mode output signal is mixed on the beamsplitter, with the strong coherent state of the local oscillator denoted as:

$$\alpha_{LO}(t) = |\alpha_{LO}| \exp(i(\omega_{LO}t + \phi - \pi/2)), \tag{5.20}$$

where the $-\pi/2$ is chosen to simplify the expression. After entering the port of the mixer, the two signals are either transmitted or reflected. The mixed signal reaching the the photoreceiver incident to the local oscillator, $\hat{a}_+$ and the signal measured on the second one, $\hat{a}_-$ is given by:

$$\hat{a}_+ = T\hat{c}_{out}(t) + R\alpha(t), \tag{5.21}$$

$$\hat{a}_- = R\hat{c}_{out}(t) + T\alpha(t) \tag{5.22}$$

where R and T is the reflection and transmission coefficients of the beamsplitter. In the case of a balanced beamsplitter, the reflection and transmission coefficients are given: $R = 1/\sqrt{2}$ and $T = i/\sqrt{2}$ respectively. Utilizing the appropriate phases, it is possible to factorize out i for the transmitted coefficient, simplifying the expressions for the two fields as:

$$\hat{a}_\pm(t) = \frac{1}{\sqrt{2}} \left(\hat{c}_{out}(t) \pm \alpha_{LO}(t)e^{i\phi}\right) \tag{5.23}$$

The photocurrents measured is proportional to the mean detection rate in the two photoreceivers, and is given by $\eta\hat{a}_+^\dagger(t)\hat{a}_+(t)$ and $\eta(-i\hat{a}_-(t))^\dagger(-i\hat{a}_-(t)) = \hat{a}_-^\dagger(t)\hat{a}_-(t)$, where η denotes the detector efficiency. A method to obtain the value of the quadratures, is by setting $\omega_{LO} = \omega_{out}$. Recall that $\hat{c}_{out}(t) \propto \exp(i\omega_{out}t)$, and $\alpha_{LO}(t) = |\alpha_{LO}| \exp(i\omega_{LO}t)$, therefore in the case of Homodyne detection, the time dependence of the phase drops out. When the photocurrent difference of the two detectors is taken, the strong local oscillator term drops out, and only left cross terms of the photon detections gets measured:

$$i_x(t) \propto \left\langle \hat{a}_+^\dagger\hat{a}_+ - \hat{a}_-^\dagger\hat{a}_- \right\rangle = 2|\alpha_{LO}| \langle x \rangle \tag{5.24}$$

This allows for the measurement or the x quadrature, also called the in-phase I. To measure the $\hat{p}$ quadrature, Q component, by homodyne detection, the signal is separated equally in two, then mixed on a microwave mixer. Fig. 8b depicts an equivalent setup for measuring both quadratures in a quantum optic analog to the I-Q mixir. The non-phase shifted fields entering the ports for the upper right detector, acts equivalently to the scenario just outlined,

except the signal field strength is halved. The fields coherent field at the southern beamsplitter, has had a $\pi/2$ phase shift while the carrier signal has not. This means that the field operators for the field reaching the two southern photo receivers in the figure, is described by:

$$\hat{a}_{y\pm} = \frac{1}{2}\left(\hat{c}_{out}(t) \pm i\alpha_{LO}(t)\right) \tag{5.25}$$

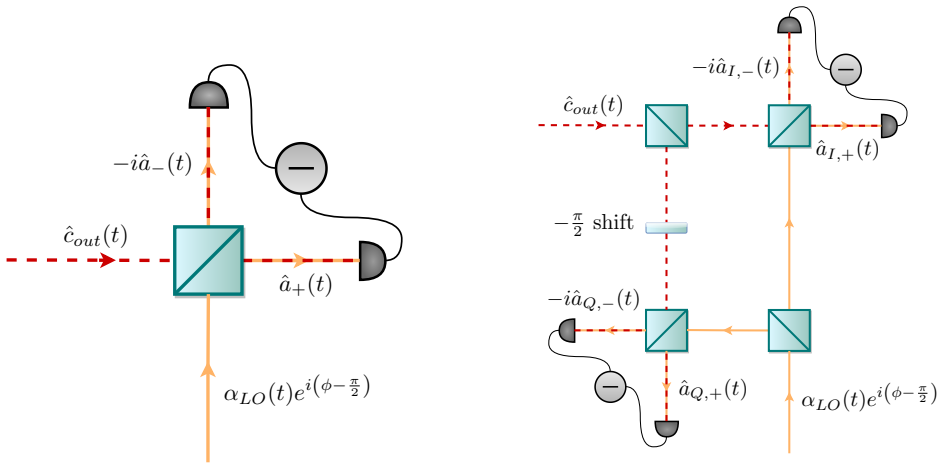
Therefore, the act of subtracting one photocurrent from the other equates to the measurement of the $\hat{p}$ quadrature, or in other words, the Q component of the field.

$$i_y(t) \propto \left\langle \hat{a}^\dagger_{y+}\hat{a}_{y+} - \hat{a}^\dagger_{y-}\hat{a}_{y-} \right\rangle = |\alpha_{LO}\langle p\rangle|. \tag{5.26}$$

In microwave mixers, this is equivalent to integrating the signal over the two signals,

$$I + iQ = \frac{1}{2\pi k}\int_0^{2\pi k} E_{out}(t)\alpha^*(t)/4\mathrm{d}t, \tag{5.27}$$

where the factor $1/4$ comes from splitting the two signals into different mixers.



(a) I-Q mixing setup

(b) Optical analog for measuring both quadratures in homodyne detection scheme.

Figure 8: (a) The schematic represents quantum optics analogue of an IQ-mixer setup with square law diodes. The outgoing frequency from cryostat is mixed with the frequency of the local oscillator. In typical setup, the two are mixed on a balanced mixer, then the detected photocurrent is demodulated into the two quadrature. (b) The homodyne setup for measurements of both quadratures is depicted. A balanced beamsplitter is employed to mix half the output signal and half the signal of the local oscillator. A relative $pi/2$-radian phase shift is experienced by the local oscillator field reaching the two mixer ports due to the reflection. Conversely, no relative phase difference is experienced between the outgoing field reaching either of the mixer ports.

5.2.2 Heterodyne detection

While homodyne detection provides a straightforward method for understanding the readout process of the qubit, in reality it has some significant drawbacks. The cancellation of the oscillating phase in the demodulating process, lowers the signal to noise ratio. Additionally, with the homodyne detection it is not possible to probe a large set of resonators with different frequencies[24].

In the case of heterodyne detection, both the quadratures can be readout using only the setup shown in Fig. 8a. The time dependent phase of the cross terms leads to an intermediate frequency, $\Delta\omega = |\omega_{out} - \omega_{LO}|$, which alters Eq. (5.24) into:

$$\left\langle \hat{a}_+^\dagger \hat{a}_+ - \hat{a}_-^\dagger \hat{a}_- \right\rangle = 2|\alpha_{LO}| \left\langle \hat{c}_{out} e^{i\Delta\omega t} e^{-i\phi} + \hat{c}_{out}^\dagger e^{-i\Delta\omega t} e^{i\phi} \right\rangle . \tag{5.28}$$

Thus, in the context of heterodyne detection, the demodulation of the two signals, I and Q, results in the following proportionality, when written as a complex number:

$$I + iQ \propto e^{i(\Delta\omega t + \theta)} \tag{5.29}$$

By selecting a phase for ω_{LO} that renders $\Delta\omega$ sufficiently small to facilitate the reasonable digitization of the signal, it becomes feasible to compute the oscillatory exponential and extract the static phasor component of Eq. (5.29). This extraction enables the measurement of the phase and the accurate readout of the qubit state.

6 Master equation of the density matrix

In instances where the qubit system is subjected to measurement or dampening processes, the elementary wave function description, although effective for closed quantum systems, fails to encapsulate the intricate nature of the open quantum system. In the face of such complexity, it becomes evident that a more sophisticated approach is necessary. This requirement gives rise to the need to construct a master equation for the density operator.

This master equation serves a dual purpose. On one hand, it mirrors the functionality of Schrödinger's equation in scenarios where the influence of the environment is negligible, thus allowing for a faithful description of the quantum system's evolution in isolation. On the other hand, it accommodates the effects of decay processes and the non-unitary collapse of quantum superpositions that occur upon measurement, phenomena that are not accounted for within the realm of Schrödinger's equation.

The upcoming section of this paper is committed to the derivation of such a master equation. The master equation will describe a qubit that is coupled to a resonator, and measured using a heterodyne detection scheme. Because of this, a stochastic contribution, arising from the probabilistic nature

of quantum mechanics, will be encompassed in the dynamics of the master equation.

6.1 Markovian Master Equation

When the qubit constitutes a closed system and is not subjected to any measurement, its evolution adheres to Schrödinger's wave equation, $|\dot{\psi_k}\rangle = -i\hat{H}\,|\psi_k\rangle$. Hence, for small time steps, $\mathrm{d}t$, the state evolves approximately with the unitary operator $\hat{U} = I - i\hat{H}\mathrm{d}t$. This evolution, described in the language of the density matrix, leads to a master equation of the system, derived using the conjugation of $\hat{U}$ on ρ:

$$\rho(t+\mathrm{d}t) = \hat{U}\rho(t)\hat{U}^\dagger = \left(I - i\hat{H}\mathrm{d}t\right)\rho(t)\left(I + i\hat{H}\mathrm{d}t\right) \tag{6.1}$$

$$= \rho(t) - i\left[\hat{H}, \rho(t)\right], \tag{6.2}$$

where terms of higher ordered terms, $\mathcal{O}(\mathrm{d}t^2)$, were intentionally removed. The state dynamics, in the Schrödinger picture, of the idealized pure-state evolution given by Eq. (6.2), represent a system that never interacts with its environment and, consequently, does not experience decoherence.

To properly model a physically realistic qubit, the system must first be considered coupled to the environment, also referred to as a *"bath"*. The Hamiltonian of the total system, constituting both the qubit system, and the bath, is most generally expressed by the Hamiltonian $\hat{H}_{tot} = \hat{H}_S + \hat{H}_B + \hat{H}_{int}(t)$. Here, $\hat{H}_S$ is the Hamiltonian of the qubits' system, i.e. the Jaynes-Cummings system under readout, or the Hamiltonian of a gate operation, $\hat{H}_B$ is the Hamiltonian of the bath, and $\hat{H}_{int}(t)$ is the interaction between the two.

The pure state density matrix describing this system, $\rho_{tot}(t)$, evolves in accordance to the master equation $\dot{\rho}_{tot}(t) = -i[\hat{H}_{tot}, \rho_{tot}(t)]$. It can be assumed that after initial preparation of the qubit states, call it time $t = 0$, the state of the qubit is not entangled with the bath, and therefore total state is the product of the two sub components, $\rho_{tot}(0) = \rho_S(0) \otimes \rho_B(0)$. Upon converting this into the interaction frame, the resulting interacting Hamiltonian is split into two terms, $\hat{H}_{int}(t) = \hat{H}_{int,S}(t) + \hat{H}_{int,SB}(t)$, where $\hat{H}_{int,S}(t)$ acts only on the Hilbert space of the qubit. It is posssible to show, using the appropriate approximations, that the dynamics of ρ_S is described by the Markov-Born master equation[32]:

$$\begin{aligned}\dot{\rho}_S(t) = &- i\left[\hat{H}_{int,S}(t), \rho_S(t)\right] \\ &- \int_{-\infty}^{t} \mathrm{Tr}_B\left(\left[\hat{H}_{int}(t), \left[\hat{H}_{int}(\tau), \rho_S(\tau)\otimes\rho_B(0)\right]\right]\right)\mathrm{d}\tau,\end{aligned} \tag{6.3}$$

where Tr_B denotes the partial trace over the bath. The generators for the

Markov master equation, takes the general Lindblad form:

$$\dot{\rho}_S(t) = -i\left[\hat{H}_S, \rho\right] + \sum_j \mathcal{D}\left[\hat{c}_j\right]\rho_S(t). \tag{6.4}$$

In this context, the influence of the entire system on the dynamics of the qubit system, which is the only one of the two systems accessible to measurements, has been encapsulated in the dissipative Lindblad term. In order to simplify the notation, the subscript S will be omitted from this point forward. The superoperator $\mathcal{D}[\hat{c}]\rho$ is defined as follows:

$$\mathcal{D}\left[\hat{c}\right]\rho = \hat{c}\rho\hat{c}^\dagger - \frac{1}{2}\left\{\hat{c}^\dagger\hat{c}, \rho\right\}, \tag{6.5}$$

where $\hat{c}$ is an arbitrary jump operator. The Lindbladian, as expressed in Eq. (6.4), is of a form that ensures the state remains normalized, which is evident from its trace-preserving behavior, $\mathrm{Tr}(\dot{\rho}) = 0$, as is proven in Sec.6.4.1. The Lindbladian master equation does not model the wave function collapse that occurs upon successful measurement of the qubit. Therefore, to comprehend an observer's understanding of the state, the master equation must be replaced by a stochastic master equation (SME).

6.2 Quantum jump and measurement collapse

To model the wave function collapse, the qubit state is considered entangled to the measurement apparatus, and its average dynamics must be described by Lindbladian as well. Considering the two possible branch scenarios. After a very short time, Δt, the qubit can *"exists"* in two distinct realities. One scenario where no photons had been detected, or one with a single photon detection. The change in the number of photons detected, ΔN, is given by the discrete sampling of $\Delta N \in \{0, 1\}$. In addition, the expectation number must reflect the probability of detecting the photon. For $\Delta t \to 0$, these conditions are expressed mathematically as:

$$\mathrm{d}N(t)^2 = \mathrm{d}N(t),, \tag{6.6}$$

$$\langle \mathrm{d}N(t)\rangle = \langle\psi(t)|\hat{c}^\dagger\hat{c}|\psi(t)\rangle\,\mathrm{d}t, \tag{6.7}$$

where $\hat{c}$ is another jump operator, signaling the emission of photon by the qubit upon decay. When no photon is detected, the qubit should evolve without jumps, and its dynamics can be described by the non-Hermitian Hamiltonian of the form:

$$\hat{H}_0 = \hat{H} - \frac{i}{2}\hat{c}^\dagger\hat{c}. \tag{6.8}$$

The jump formalism doesn't guarantee normalization, as measurement operator, $\hat{M}_0$, generated by the non-Hermitian Hamiltonian is a non-unitary matrix,

$\hat{M}_0^\dagger \hat{M}_0 \approx (I + i\hat{H}_0^\dagger \mathrm{d}t)(I - i\hat{H}_0 \mathrm{d}t) = I + i(\hat{H}_0^\dagger - \hat{H}_0)\mathrm{d}t \neq I$. To properly compute the evolution in the case of no detection, the state vector is first evolved with the non hermitian Hamiltonian, and then afterwards renormalize,

$$|\psi_0(t + \mathrm{d}t)\rangle = \frac{\hat{U}_0(\mathrm{d}t)\,|\psi(t)\rangle}{\left\|\hat{U}_0(\mathrm{d}t)\,|\psi(t)\rangle\right\|}. \tag{6.9}$$

The probability $\langle \hat{c}^\dagger \hat{c} \rangle \mathrm{d}t$ of a detection occurring during an infinitesimal time interval is proportional to that same time interval, resulting in an almost zero probability of a measurement taking place during each $\mathrm{d}t$. Nonetheless, at random instances and with a rate of $\langle \hat{c}_{out}^\dagger \hat{c}_{out} \rangle$, a detection does indeed register a click. These detection corresponds to an instantaneous update in an observers knowledge about the system. While it may not describe physical leaps, as Bohr envisioned it, this sudden change in the qubit state after the wave function collapse, is typically referred to as a quantum jump, similar in nature to Bohr's jump mechanism in his 1913 description of the atom. Following the quantum jump, the qubit's excitation has been transferred to the detected photon. Consequently, the projection of $\hat{c}$ is employed to refine the state of knowledge concerning the system, and to re-normalize the state:

$$|\psi_1(t + \mathrm{d}t)\rangle = \frac{\hat{c}\,|\psi(t)\rangle}{\|\hat{c}\,|\psi(t)\rangle\|}. \tag{6.10}$$

Combining the two possible "branches" with the detection condition (6.6), it is discovered that the dynamic of the state vector, is described by the stochastic Schrodinger equation (SSE) of the form:

$$|\psi(t + \mathrm{d}t)\rangle = \mathrm{d}N(t)\,|\psi_1(t + \mathrm{d}t)\rangle + (1 - \mathrm{d}N(t))\,|\psi_0(t + \mathrm{d}t)\rangle\,. \tag{6.11}$$

Expanding (6.9) to the first order in $\mathrm{d}t$ yields a non-linear term proportional to $\mathrm{d}t$. Ignoring the terms of order $\mathrm{d}N(t)\mathrm{d}t$, and explicitly writing out the terms of Eq. (6.11), gives a non-linear SSE of the form:

$$\mathrm{d}\,|\psi(t)\rangle = \left[\mathrm{d}N(t)\left(\frac{\hat{c}}{\sqrt{\langle \hat{c}^\dagger \hat{c} \rangle}} - 1\right) + \left(\frac{\langle \hat{c}^\dagger \hat{c} \rangle}{2} - \frac{\hat{c}^\dagger \hat{c}}{2} - i\hat{H}\right)\mathrm{d}t\right]|\psi(t)\rangle\,. \tag{6.12}$$

The SSE describing the jump formalism is useful when the detections are made directly on the qubits themselves, but fails to consider scenario where measurements of individual photons does not directly project the state onto the computational states, e.g. when dealing with the reflected signal of the resonator cavity, as discussed in Sec. 5.2. Luckily, using the density matrix, it is possible model the dynamics of continuous measurement in homodyne and heterodyne detection schemes.

6.2.1 Homodyne Detection

Readout of the signal during homodyne detection does not count individual photon detections, but rather the photocurrent is measured, after mixture with a strong local oscillator that amplifies the readout signal of the microwave field. Defining the average photocurrent as the photon detection rate, then the photocurrents exiting the mixers are given by:

$$\langle i_+(t)\rangle = \frac{\mathrm{d}\langle n_+(t)\rangle}{\mathrm{d}t}, \tag{6.13}$$

$$\langle i_-(t)\rangle = \frac{\mathrm{d}\langle n_-(t)\rangle}{\mathrm{d}t}. \tag{6.14}$$

In the interest of also simulating the mixed state dynamics, it becomes necessary to shift from the state vector picture to a density matrix picture. This means using the projector $\rho = |\psi(t)\rangle\langle\psi(t)|$, and apply Eq. (6.12) to repackage the dynamics of a stochastic Schrodinger equation, as that of a stochastic master equation (SME):

$$\begin{aligned}\mathrm{d}\rho(t) &= |\mathrm{d}\psi(t)\rangle\langle\psi(t)| + |\psi(t)\rangle\langle\mathrm{d}\psi(t)| + |\mathrm{d}\psi(t)\rangle\langle\mathrm{d}\psi(t)| \\ &= -i\left[\hat{H},\rho(t)\right] - \frac{1}{2}\mathcal{H}\left[\eta\hat{c}^\dagger\hat{c}\right]\rho(t)\mathrm{d}t + \mathcal{G}[\sqrt{\eta}\hat{c}]\rho(t)\mathrm{d}N(t).\end{aligned} \tag{6.15}$$

Here $\mathcal{H}[\hat{c}^\dagger\hat{c}]\rho(t)$ and $\mathcal{G}[\hat{c}]\rho(t)$ are superoperators, and are given, for an arbitrary input operator, as:

$$\mathcal{H}\left[\hat{a}\right]\rho(t) = \hat{a}\rho(t) + \rho(t)\hat{a}^\dagger - \left\langle\hat{a}+\hat{a}^\dagger\right\rangle\rho(t), \tag{6.16}$$

$$\mathcal{G}\left[\hat{a}\right]\rho(t) = \frac{\hat{a}\rho(t)\hat{a}^\dagger}{\langle\hat{a}^\dagger\hat{a}\rangle} - \rho(t). \tag{6.17}$$

Note that since $\mathrm{Tr}(\rho) = 1$, both these super operators are trace preserving as proven in Sec. 6.4.1. The average dynamics of Eq. (6.15), is master equation of the general Lindblad form.

The mixed signal incident to the outgoing field, $\hat{a}_+$, and the mixed signal incident to the other port of the local oscillator, $\hat{a}_-$ gets detected on the two photoreceivers, as illustrated in Fig. 8a. If the two photoreceivers are imperfect detectors, with an efficiency of η, then the SME of the system is written as:

$$\begin{aligned}\mathrm{d}\rho(t) =& -i\left[\hat{H},\rho(t)\right]\mathrm{d}t - \frac{1}{2}\mathcal{H}\left[\eta\hat{a}_+\hat{a}_+^\dagger\right]\rho(t)\mathrm{d}t + \mathcal{G}\left[\sqrt{\eta}\hat{a}_+\right]\rho(t)\mathrm{d}n_+ \\ &- \frac{1}{2}\mathcal{H}\left[\eta\hat{a}_-\hat{a}_-^\dagger\right]\rho(t)\mathrm{d}t + \mathcal{G}\left[\sqrt{\eta}\hat{a}_-\right]\rho(t)\mathrm{d}n_- + (1-\eta)\mathcal{D}[\hat{c}_{out}]\rho(t)\end{aligned} \tag{6.18}$$

Here, the term $(1-\eta)\mathcal{D}[\hat{c}_{out}]\rho(t)$ represents decay due to the outgoing field that is not measured by the receivers. To simplify the derivation, a simple case

is considered in which, instead of using balanced detection, the beamsplitter perfectly transmits the outgoing field such that:

$$\hat{a}_+ \to \hat{c}_{out} + \alpha_{LO}, \tag{6.19}$$
$$\hat{a}_- \to 0. \tag{6.20}$$

In practical applications, the 50/50 beamsplitter proves to be more advantageous, as the squared terms of the strong local oscillator are canceled out, better signal to noise ratio. In the theoretical derivation of the idealized limit, these two cases yield equivalent master equations. Under homodyne detection, the Hamiltonian transforms into a Hamiltonian of the form:

$$\hat{H} \to \hat{H} - i\frac{1}{2}\left(\alpha^*_{LO}\hat{c}_{out} - \alpha_{LO}\hat{c}^\dagger_{out}\right). \tag{6.21}$$

The master equation of the general Lindblad form, Eq. (6.4), is left invariant under the transformations in equations (6.19) and (6.21). This also ensures that the expectation value of the SME, Eq. (6.15), is a Lindbladian master equation as well, such that the effects of adding the local oscillator doesn't change the expected readout values of the qubits.

The detector incident to the outgoing field, has a detection rate:

$$\begin{aligned}\frac{\mathrm{d}\langle n_+\rangle}{\mathrm{d}t} &= \left\langle \eta\hat{a}^\dagger_+\hat{a}_+ \right\rangle = \eta\left\langle \hat{c}^\dagger_{out}\hat{c}_{out} + |\alpha_{LO}|^2 + \alpha_{LO}\hat{c}^\dagger_{out} + \alpha^*_{LO}\hat{c}_{out}\right\rangle \\ &= \eta\left\langle \hat{c}^\dagger_{out}\hat{c}_{out} + |\alpha_{LO}|^2 + 2|\alpha_{LO}|\hat{x}\right\rangle.\end{aligned} \tag{6.22}$$

In the limit where $|\alpha_{LO}|$ is significantly larger than $\langle\hat{c}^\dagger_{out}\hat{c}_{out}\rangle$, the photocurrent is dominated by the field amplitude of the local oscillator, with a smaller contribution from the term $2|\alpha_{LO}|\langle x\rangle$. In the ideal limit of continuous measurement, corresponding mathematically to the case where $|\alpha_{LO}| \to \infty$, the number of photons detected at any finite time interval Δt approaches infinity, while the impact of each detection on the system diminishes.

Letting α_{LO} be real, such that $\phi = 0$, and $\hat{c}_{out} + \hat{c}^\dagger_{out} = 2\hat{x}$. A measurement of the $\hat{x}$ has Gaussian noise fluctuation, as noted in (5.18). In the ideal limit for $\alpha^2_{LO} \gg \hat{c}^\dagger_{out}\hat{c}_{out}$, the photon number of the weak microwave field is neglected on the time scales of Δt, and so the number of detections thus written as:

$$\Delta n_+ = \eta\alpha^2_{LO}\Delta t\left[1 + \frac{2\langle x\rangle}{\alpha_{LO}}\right] + \sqrt{\eta}\alpha_{LO}\Delta W(t). \tag{6.23}$$

Here, $\Delta W(t)$ represents the Wiener increment, which causes the distribution of ΔN to be Gaussian. The Wiener increment is a number sampled from a normal distribution, $\Delta W(t) \propto \mathcal{N}(0,t)$, with a mean of $\langle\Delta W(t)\rangle = 0$ and a variance of $\langle\Delta W(t)^2\rangle = \Delta t$. These Wiener increments can be considered a random walk, therefore, in accordance with the central limit theorem, remains

Gaussian. Eq. (6.15) under this transformation is given by:

$$\begin{aligned}\mathrm{d}\rho(t) = &- i\left[\hat{H},\rho(t)\right] - \eta\frac{1}{2}\mathcal{H}\left[\hat{c}_{out}^{\dagger}\hat{c}_{out} + 2\alpha_{LO}\hat{c}_{out}\right]\rho(t)\mathrm{d}t \\ &+ \mathcal{G}\left[\sqrt{\eta}\hat{c}_{out} + \sqrt{\eta}\alpha_{LO}\right]\rho(t)\mathrm{d}n_{+}(t) + (1-\eta)\,\mathcal{D}[\hat{c}_{out}]\rho(t)\mathrm{d}t\end{aligned} \tag{6.24}$$

The change in the system over this time interval is very small, even though the number of detections are large. Expanding Eq. (6.24) in powers of α_{LO}^{-1}, yields the systematic change during this time:

$$\begin{aligned}\Delta\rho(t) = &- i\left[\hat{H},\rho(t)\right]\Delta t - \eta\frac{1}{2}\mathcal{H}\left[\hat{c}_{out}^{\dagger}\hat{c}_{out} + 2\alpha_{LO}\hat{c}_{out}\right]\rho(t)\Delta t \\ &+ \Delta n_{+}(t)\left(\frac{\mathcal{H}[\hat{c}_{out}]}{\alpha_{LO}} + \frac{\left\langle\hat{c}_{out}^{\dagger}\hat{c}_{out}\right\rangle\mathcal{G}[\hat{c}_{out}] - 2\left\langle x\right\rangle\mathcal{H}[\hat{c}_{out}]}{\alpha_{LO}^{2}}\right)\rho(t) \\ &+ (1-\eta)\,\mathcal{D}[\hat{c}_{out}]\rho(t)\Delta t + \mathcal{O}\left(\alpha_{LO}^{-3}\right).\end{aligned} \tag{6.25}$$

In the continuum limit, where $\Delta t \to \mathrm{d}t$, and ignoring terms that are orders smaller than $\alpha_{LO}^{-1/2}$, the SME for the case of Homodyne detection is given by:

$$\mathrm{d}\rho(t) = -i\left[\hat{H},\rho(t)\right]\mathrm{d}t + \mathcal{D}[\hat{c}_{out}]\rho(t)\mathrm{d}t + \sqrt{\eta}\mathcal{H}[\hat{c}_{out}]\rho(t)\mathrm{d}W(t). \tag{6.26}$$

Here the Wiener increment have become infinitesimal, but otherwise following the same rules as previously, i.e. $\langle\mathrm{d}W(t)\rangle = 0$ and $\langle\mathrm{d}W(t)^2\rangle = \mathrm{d}t$. As discussed previously, the SME is equivalent for a 50/50 beamsplitters.

Returning to a balanced homodyne detection scheme, the photon detections detections are made on both detectors, such that the detection made in the time Δt is given by:

$$\Delta n_{+} = \frac{1}{2}\left(\eta\alpha_{LO}^{2}\Delta t\left[1 + \frac{2\left\langle x\right\rangle}{\alpha_{LO}}\right] + \sqrt{\eta}\alpha_{LO}\Delta W_{+}(t)\right) \tag{6.27}$$

$$\Delta n_{-} = \frac{1}{2}\left(\eta\alpha_{LO}^{2}\Delta t\left[1 - \frac{2\left\langle x\right\rangle}{\alpha_{LO}}\right] + \sqrt{\eta}\alpha_{LO}\Delta W_{-}(t)\right). \tag{6.28}$$

Such that the difference of the two currents are given by:

$$i(t) = \lim_{\Delta t\to 0}\frac{\Delta n_{+} - \Delta n_{-}}{\sqrt{\eta}\alpha_{LO}\Delta t} = 2\left\langle x(t)\right\rangle + \frac{\mathrm{d}W(t)}{\mathrm{d}t} \tag{6.29}$$

Note, that the two independent Wiener increments with mean zero, has the property that $\langle\Delta W_{+} \pm \Delta W_{-}\rangle = \langle\Delta W\rangle$ and $\langle(\Delta W_{+} \pm \Delta W_{-})^2\rangle = 2\langle\Delta W^2\rangle$.

6.2.2 Heterodyne Detection

As discussed in Sec. 5.2, the schemes typically used in superconducting circuits, is one where the frequencies ω_{out} and ω_{LO} are detuned from one another. To understand the dynamics of this problem, the first order of business

is considering the crossterm $\hat{c}_{out}(t)\alpha_{LO}(t) = \hat{c}_{out}\alpha_{LO}e^{i\Delta\omega t}$, which now includes a time dependent rotation, with the detuning of $\Delta\omega = \omega_{out} - \omega_{LO}$. In the transformation of Eq. (6.15), any terms of $\hat{c}_{out}$ would result in either complex squared terms, or cross terms between $\hat{c}_{out}$ and α_{LO}. Hence, the Homodyne SME (6.26) is transformed into a Heterodyne SME, by replacing terms of $\hat{c}_{out}$ with the remaining cross term contribution, $\hat{c}_{out}e^{i\Delta\omega t}$, in the limit of very strong coherence field, $\alpha_{LO} \to \infty$, giving:

$$\begin{aligned} \mathrm{d}\rho(t) &= -i\left[\hat{H}, \rho(t)\right]\mathrm{d}t + \mathcal{D}[\hat{c}_{out}]\rho(t)\mathrm{d}t \\ &+ \sqrt{\eta}\mathrm{d}W(t)\left(e^{i\Delta\omega t}\left[\hat{c}_{out}\rho(t) - \langle\hat{c}_{out}\rangle\rho(t)\right] + e^{-i\Delta\omega t}\left[\rho(t)\hat{c}_{out}^{\dagger} - \langle\hat{c}_{out}\rangle\rho(t)\right]\right), \end{aligned} \tag{6.30}$$

note the symmetry of the Lindblad operator, $\mathcal{D}[\hat{c}_{out}e^{i\Delta\omega t}]\rho(t) = \mathcal{D}[\hat{c}_{out}]\rho(t)$. The rotation introduces noise of the $\hat{p}$ quadrature, which is larger than the speed at which rotations average to zero, and so a new set of Gaussian random variables are defined. When averaging over a lot of detuning cycles, these variables becomes two independent Wiener increments for the two quadratures, and the SME is given by:

$$\mathrm{d}\rho(t) = -i\left[\hat{H}, \rho(t)\right]\mathrm{d}t + \mathcal{D}[\hat{c}_{out}]\rho(t)\mathrm{d}t + \mathcal{H}[\mathrm{d}W^*(t)\hat{c}_{out}]\rho(t), \tag{6.31}$$

where the normalized complex Wiener process is defined by:

$$\mathrm{d}W(t) = \frac{1}{\sqrt{2}}\left(\mathrm{d}W_x(t) + i\mathrm{d}W_p(t)\right). \tag{6.32}$$

In this case, the two photoreceivers measure each their own photocurrent, given by the stochastic equations:

$$i_x(t) = 2\langle x(t)\rangle + \sqrt{2}\frac{\mathrm{d}W_x}{\mathrm{d}t} \tag{6.33}$$

$$i_p(t) = 2\langle p(t)\rangle + \sqrt{2}\frac{\mathrm{d}W_p}{\mathrm{d}t} \tag{6.34}$$

The heterodyne detection thus allows for measurement of both quadratures, using the simpler setup of Fig. 8a, rather than the more complicated setup needed in case of homodyne detection, shown in Fig. 8b.

6.3 Cavity-field

Upon measurement, the cavity is subject to a coherent driving field, $\beta_{in} = \beta e^{-i\omega_d t}$. The effects of this field, is understood by writing the Hamiltonian in a frame rotating with the driving field:

$$\hat{H} = \Delta_{dr}\hat{a}^{\dagger}\hat{a} + \sum_s\left(\Delta_{ds}\frac{\hat{\sigma}_z^{(s)}}{2} + \chi_s\hat{a}^{\dagger}\hat{a}\hat{\sigma}_z^{(s)} + S\left(\chi_s, g_s\right)\right) + i\sqrt{2\kappa_1}\left(\beta\hat{a}^{\dagger} - \beta^*\hat{a}\right) \tag{6.35}$$

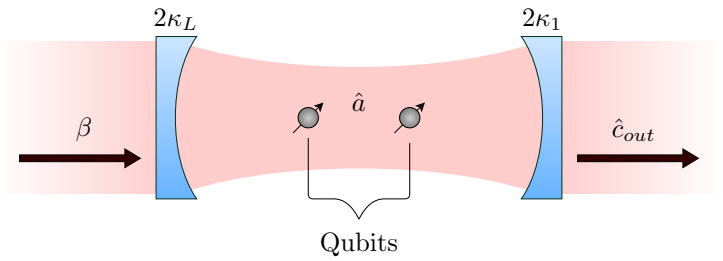


Figure 9: The cavity is driven externally by a coherent field, β. The field decays with a loss of κ_L, and a transmission of κ_1 through the cavity mirrors. The outcoupled field $\hat{c}_{out}$ is mixed with a strong local oscillator, before being readout using a heterodyne scheme.

where $\Delta_{dr} = \omega_r - \omega_d$, is the detuning between the cavity, and driving field, $\Delta_{ds} = \tilde{\omega_s} - \omega_d$ is the detuning between the spin and the driving field, and κ_1 is the amplitude decay rate of the cavity field, which due to transmission to the outgoing field, see Fig. 9.

The cavity leaks the photon field by a rate $\kappa = \kappa_1 + \kappa_L$, where κ_L represents losses out of the cavity. The system is hence subjected to dampening processes, yielding the following Jump operators:

$$\hat{c}_1 = \sqrt{2\kappa}\hat{a}, \tag{6.36}$$

$$\hat{c}_2^{(s)} = \sqrt{\frac{\gamma_\phi}{2}}\hat{\sigma}_s^{(z)}, \tag{6.37}$$

where $\hat{c}_1$ represents the field losses out of the cavity, and $\hat{c}_2$ is rate of dephasing of the qubits, which can be understood as undetected spontaneous emission of the qubits, as demonstrated in [33].

The transmitted outgoing field is described by the field leaving the cavity, in the path of the beamsplitter:

$$\hat{c}_{out} = \sqrt{2\kappa_1}(\hat{a} + \beta). \tag{6.38}$$

At this stage, the relative phase between the outgoing field and the local oscillator, as shown in equations (5.16) and (5.17), has not been considered. To address this, the relative phase can be assigned to the operator through the following transformation:

$$\hat{c}_{out} = \hat{c}_3 \rightarrow \hat{c}_{out}e^{i\phi} \tag{6.39}$$

6.4 Adiabatic elimination of the cavity

When the interactions between the qubit and the cavity field mode are dispersive, causing the energy exchanges between the qubit and the cavity to freeze, the cavity stabilizes. The implications of this can be examined by transitioning to the Heisenberg picture and observing the time evolution of the cavity

operator $\hat{a}$. By applying the Heisenberg equations of motion, the evolution of the cavity is characterized by:

$$\dot{\hat{a}} = -i\left[\hat{a}, H\right] - \kappa\hat{a} + \hat{F}. \tag{6.40}$$

Here the operator $\hat{F}$ describes the a Langevin noise originating from the vacuum-field, and has vanishing expectation value. The damping term, $-\kappa a$, stems from a non-Hermitian term $-i\kappa\hat{a}^\dagger\hat{a}$ to account for the leakage out of the cavity[19]. The dynamics of the cavity, which is assumed to rapidly average to zero, is uninteresting for the readout of the qubits, and therefore is adiabatically eliminated, setting $\dot{\hat{a}} = 0$. It is assumed in this study that utilizing the drive field's reference frame leads to a decent approximation for the elimination of the cavity in the dispersive regime. However, it must be recognized that in practice, superior approximations could potentially be achieved through the selection of alternative reference frames[34]. Isolating $\hat{a}$ on the left hand side yields the equation for the constant cavity-field:

$$\hat{a} = \frac{\sqrt{2\kappa_1}\beta}{i\Delta_{dr} + \kappa + i\sum_s \chi_s\hat{\sigma}_z^{(s)}}. \tag{6.41}$$

Computation of the inverse matrix in the denominator is possible, but such algorithms are typically laborious and more inefficient than merely computing the eigenvalues. Fortunately, when analyzing the qubit dynamics within the Schrödinger picture, where the only parameter of relevance to the state's evolution are the eigenvalues of the operator, when applied to the state.

Getting the cavity field operator is straight forward, since for an eigenequation $\hat{M}\left|m\right\rangle = \lambda_m\left|m\right\rangle$, if the operator gets mapped to a power, then the mapping obeys $f(\hat{M})\left|m\right\rangle = f(\lambda_m)\left|m\right\rangle$. This allows for diagonalization the cavity field operator:

$$\hat{a} = \sum_m \frac{\sqrt{2\kappa_1}\beta}{i\Delta_{dr} + \kappa + i\lambda_m}\left|m\right\rangle\!\left\langle m\right|, \tag{6.42}$$

where $\left|m\right\rangle\!\left\langle m\right|$ represents the outer product of the basis state vectors. The eigenvalue λ_m is the eigenvalue of $\sum_s \chi_s\hat{\sigma}_z^{(s)}$, applied to the eigenvectors of the computational basis, $\left|m\right\rangle = \bigotimes_q \left|m_q\right\rangle$, for some $m_q = \{0, 1\}$.

The eigenvalues of $\hat{\sigma}_z^{(s)}$ of course depends on the state of the s'th qubit. It is 1 in case $m_s = 0$ and -1, when $m_s = 1$. The eigenvalue, λ_m, is then given for any arbitrary basis state as:

$$\lambda_m = \sum_q (-1)^{m_q}\chi_q. \tag{6.43}$$

6.4.1 Trace preservation

This purpose of this section is to prove the trace preserving properties of the stochastic master equation. This can function as a sanity check since

the trace preservation ensures the normalized evolution of the density matrix, guaranteeing a physically realistic model. To accomplish this, the trace of each term is separately examined.

$$\mathrm{d}\rho(t) = -i\left[\hat{H}, \rho(t)\right]\mathrm{d}t + \sum_j \mathcal{D}[\hat{c}_j]\rho\mathrm{d}t + \mathcal{H}[\mathrm{d}W^*(t)\hat{c}_{out}]\rho(t) \tag{6.44}$$

Trace preservation of the term: $-i[H, \rho]$

The proof begins by writing the commutator

$$\begin{aligned} \mathrm{Tr}[-i[H,\rho]] &= -i\,\mathrm{Tr}[H\rho - \rho H] \\ &= -i\left(\mathrm{Tr}[H\rho] - \mathrm{Tr}[\rho H]\right). \end{aligned} \tag{6.45}$$

By utilizing the cyclic property of trace, the proof can be completed:

$$\mathrm{Tr}[-i[H,\rho]] = -i\left(\mathrm{Tr}[H\rho] - \mathrm{Tr}[H\rho]\right) = 0. \tag{6.46}$$

Consequently, it is deduced that this term preserves the trace.

Trace preservation of the term: $\mathcal{D}[\hat{c}]\rho\mathrm{d}t$

By inserting the definition of the Lindblad operator as detailed in Eq. (6.5), and computing its trace, the following result is obtained:

$$\begin{aligned} \mathrm{Tr}[\mathcal{D}[\hat{c}]\rho\mathrm{d}t] &= \mathrm{Tr}\left[\hat{c}\rho\hat{c}^\dagger - \frac{1}{2}\left\{\hat{c}^\dagger\hat{c}, \rho(t)\right\}\right]\mathrm{d}t \\ &= \mathrm{Tr}\left[\hat{c}\rho\hat{c}^\dagger\right]\mathrm{d}t - \frac{1}{2}\mathrm{Tr}\left[\left\{\hat{c}^\dagger\hat{c}, \rho\right\}\right]\mathrm{d}t \end{aligned} \tag{6.47}$$

The anti-commutator is then explicitly written out as:

$$\begin{aligned} \mathrm{Tr}[\mathcal{D}[\hat{c}]\rho\mathrm{d}t] &= \mathrm{Tr}\left[\hat{c}\rho\hat{c}^\dagger\right]\mathrm{d}t - \frac{1}{2}\mathrm{Tr}\left[\hat{c}^\dagger\hat{c}\rho + \rho\hat{c}^\dagger\hat{c}\right]\mathrm{d}t \\ &= \mathrm{Tr}\left[\hat{c}\rho(t)\hat{c}^\dagger\right]\mathrm{d}t - \frac{1}{2}\mathrm{Tr}\left[\hat{c}^\dagger\hat{c}\rho\right]\mathrm{d}t - \frac{1}{2}\mathrm{Tr}\left[\rho\hat{c}^\dagger\hat{c}\right]\mathrm{d}t. \end{aligned} \tag{6.48}$$

The cyclic property of the trace permits the rearrangement of its contents, yielding:

$$\begin{aligned} \mathrm{Tr}[\mathcal{D}[\hat{c}]\rho\mathrm{d}t] &= \mathrm{Tr}\left[\hat{c}^\dagger\hat{c}\rho(t)\right]\mathrm{d}t - \frac{1}{2}\mathrm{Tr}\left[\hat{c}^\dagger\hat{c}\rho\right]\mathrm{d}t - \frac{1}{2}\mathrm{Tr}\left[\hat{c}^\dagger\hat{c}\rho\right]\mathrm{d}t \\ &= \mathrm{Tr}\left[\hat{c}^\dagger\hat{c}\rho\right]\mathrm{d}t - \mathrm{Tr}\left[\hat{c}^\dagger\hat{c}\rho\right]\mathrm{d}t = 0. \end{aligned} \tag{6.49}$$

Therefore, it is deduced that this term is unconditionally trace-preserving.

Trace preservation of the term: $\mathcal{H}[\mathrm{d}W^*(t)\hat{c}_{out}]\rho(t)$

The definition of the super-operator is explicitly written out as defined in equation (6.16),

$$\begin{aligned}\mathrm{Tr}[\mathcal{H}\left[\mathrm{d}W^*(t)\hat{c}_{out}\right]\rho] = \mathrm{Tr}[\mathrm{d}W^*(t)\hat{c}_{out}\rho] + \mathrm{Tr}\Big[\rho(\mathrm{d}W^*(t)\hat{c}_{out})^\dagger\Big] \\ - \mathrm{Tr}\Big[\mathrm{Tr}\Big(\mathrm{d}W^*(t)\hat{c}_{out}\rho + \rho(\mathrm{d}W^*(t)\hat{c}_{out})^\dagger\Big)\rho\Big].\end{aligned} \tag{6.50}$$

Computing the dagger is trivial,

$$(\mathrm{d}W^*(t)\hat{c}_{out})^\dagger = \mathrm{d}W(t)\hat{c}_{out}^\dagger. \tag{6.51}$$

This, coupled with the understanding that the traces are composed of scalars, enables the rearrangement of the factors.

$$\begin{aligned}\mathrm{Tr}[\mathcal{H}\left[\mathrm{d}W^*(t)\hat{c}_{out}\right]\rho] = \mathrm{Tr}[\mathrm{d}W^*(t)\hat{c}_{out}\rho] + \mathrm{Tr}\Big[\rho\mathrm{d}W(t)\hat{c}_{out}^\dagger\Big] \\ - \Big(\mathrm{Tr}[\mathrm{d}W^*(t)\hat{c}_{out}\rho] + \mathrm{Tr}\Big[\rho\mathrm{d}W(t)\hat{c}_{out}^\dagger\Big]\Big)\,\mathrm{Tr}[\rho].\end{aligned} \tag{6.52}$$

Assuming that $\mathrm{Tr}[\rho(t)] = 1$, which it is defined to be, the rest becomes trivial.

$$\begin{aligned}\mathrm{Tr}[\mathcal{H}\left[\mathrm{d}W^*(t)\hat{c}_{out}\right]\rho] = \mathrm{Tr}[\mathrm{d}W^*(t)\hat{c}_{out}\rho] + \mathrm{Tr}\Big[\rho\mathrm{d}W(t)\hat{c}_{out}^\dagger\Big] \\ - \mathrm{Tr}[\mathrm{d}W^*(t)\hat{c}_{out}\rho] - \mathrm{Tr}\Big[\rho\mathrm{d}W(t)\hat{c}_{out}^\dagger\Big].\end{aligned} \tag{6.53}$$

Therefore,

$$\mathrm{Tr}[\mathcal{H}\left[\mathrm{d}W^*(t)\hat{c}_{out}\right]\rho] = 0. \tag{6.54}$$

While this term is guaranteed to be trace preserving within the bounds of the definition of the density matrix, it is important to note that within the context of numerical precision, an assumption has been made, which is why the word "assuming" was written in bold. This term is trace preserving if and only if the condition $\mathrm{Tr}[\rho(t)] = 1$ is satisfied. This condition will become important in Sec. 7.7.

7 Implementation

All code written for the simulator and plotting, along with how to run it, can be found here:
`https://github.com/anas2660/MSc-quantum-trajectories-simulator`

7.1 Implementation Considerations

Numerous design considerations have been made in the course of implementing the simulator, all aiming towards preserving high performance, thus facilitating the simulation of larger systems on the same hardware. This has necessitated keeping scalability at the forefront of design decisions, which, in essence, implies that the implementation should be able to efficiently leverage hardware enhancements and the growing number of processing cores introduced with each new generation. Ideally, the design should be such that it could be effectively run on GPUs as well as CPUs, considering that GPUs are typically engineered to yield significantly higher throughput for floating point operations. In this thesis, a GPU implementation hasn't been made, only architecture considerations with respect to it.

To design an efficient system, one must first understand what tools are available in the underlying platform. It is easy to forget that processors are not complete blackboxes, but in fact physical machines with capabilities and functions like any other machine. They just happen to be solid state. For every generation the design of the chip changes slightly, however, to retain backwards compatibility in software, such that the same software can be run on newer generation processors, chips are designed to run the same instruction set architecture (`ISA`). This means that the encoding for the same core instructions will remain the same through generations. As processors advance, chip designers will add new instructions, for the sake of performance, in the form of extensions.

In the context of this project, the prevalent desktop architecture, `x86_64` (as of the time of writing), has been selected as the target. This architecture has seen the addition of a multitude of extensions over the years, some of which have gained widespread support across any recent generation of processors, while others have not. The design of our implementation is structured around the vector instruction set extensions up to and including `AVX2`, as well as the fused multiply-add (`FMA3`) extensions. This blend of instruction set architecture and instruction set extensions ensures that the implementation will be widely supported on any desktop processor released over the past decade. Nevertheless, this does not imply that the exact solution is completely incompatible with older hardware, as discussed in Section 7.1.1.

7.1.1 Choosing the right tools

Results relevant to our objectives are primarily Monte Carlo-like composites of multiple runs. Consequently, the entire system needs to be custom-built to enable a configuration that is highly optimized for the specific requirements of the simulation described in this thesis.

Firstly, a programming language, in which to write our implementation is chosen. The natural choice would be to choose the most efficient language, which is C, as can be seen in Table 3. However, Rust was chosen for the

Language	Energy
C	1.0
Rust	1.03
C++	1.34
Ada	1.7
Java	1.98
⋮	⋮
Ruby	69.91
Python	75.88
Perl	79.58

Language	Time
C	1.00
Rust	1.04
C++	1.56
Ada	1.85
Java	1.89
⋮	⋮
Perl	65.79
Python	71.90
Lua	82.91

Table 3: Approximate normalized energy efficiency and execution time for programming languages[35]. The list is shortened to only show the top and bottom of the original table.

following reasons:

- Rust nearly matches the speed of C.
- It is a compiled language, and therefore giving access to machine instructions, such as the vector instructions in `AVX2`.
- Rust has higher level features such as operator overloading which in this case can make the code more understandable.

No libraries or packages were used other than `rand`, which generates random numbers, and `rand_distr`, which is for sampling on the normal distribution using the given random numbers. However, the implementation relies on the `portable_simd` module of the standard library, which is, as of writing, in nightly only. This means that the interface to it is subject to change, and therefore our implementation might need minor changes in order to run in the future. It compiles and runs as of `rustc 1.69.0-nightly (34e6673a0 2023-02-25)`. A benefit of using `portable_simd` rather than using explicit instructions for `AVX2` and `FMA3`, is that the compiler can replace the code with something more portable, if the target architecture does not support those instructions.

7.1.2 Eliminating common inefficiencies

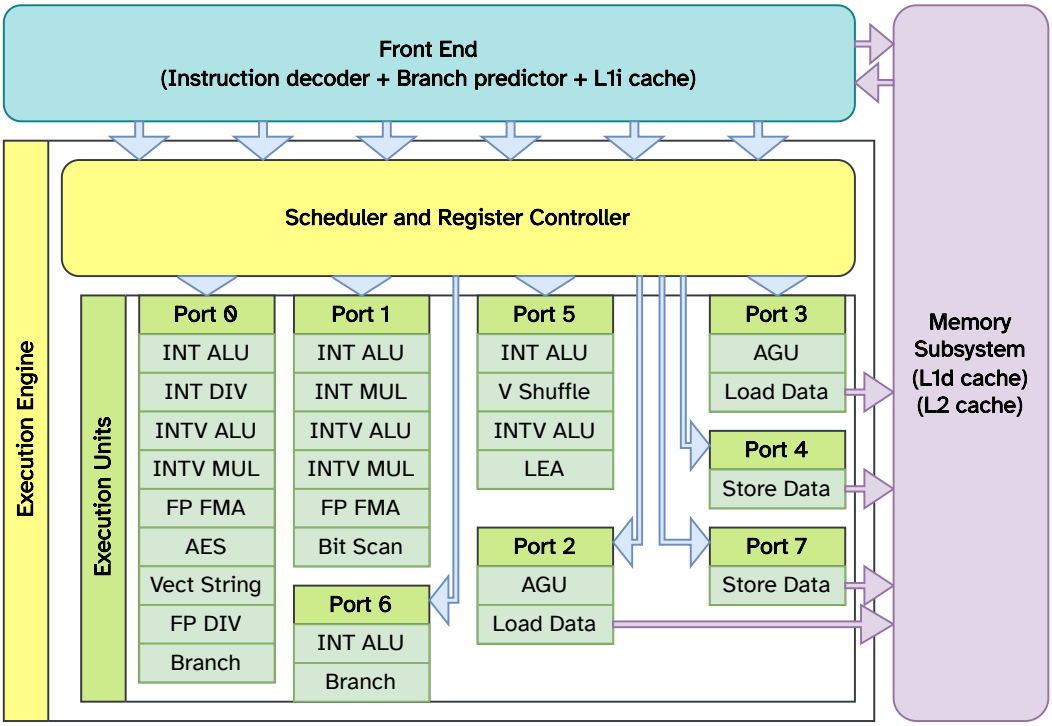


Figure 10: Simplified view of a Kabylake/Skylake processor[36, 37]. The blue arrows represent a μOP[1] transfer, and purple arrows represent data transfers. Everything in this diagram runs in parallel every cycle, except for the individual operations within each execution unit, which represents the capabilities of that specific execution unit. Each execution unit can only perform one operation every cycle. For example, there are a maximum of 6 μOP transfers from the front end to the execution engine per cycle. There are 8 execution units that execute with respect to their capabilities in parallel per cycle. The relevant execution unit capabilities include: `ALU`, arithmetic-logic unit, which does logical operations such as `and` and arithmetic. `DIV`, does division. `MUL`, does multiplication. `FMA`, fused multiply-add, does vector floating point multiplication and addition in a single μOP. Where `V` and `Vect` are short for "vector". `INT` is short for "integer", and `FP` is short for "floating point". For example, `INTV MUL` means "integer vector multiplication".

[1] A μOP is an instruction that can be executed in at least one of the execution units. A normal instruction can expand to multiple μOPs when decoded in the front end.

The first common inefficiency, which has been addressed, is the execution of redundant instructions. For instance, consider the instructions required for running the Python interpreter. More specifically, the ones for interpreting types in expressions, by reading metadata, to compute what necessary instructions to execute. This inefficiency has been simply been addressed by using a statically typed language which compiles to native machine code, such that

all reasoning about which specific instructions to execute has been done at compile time by the compiler.

The second common inefficiency, is branch misprediction. Modern processors are highly parallel machines, even within each core. This can be observed in Fig. 10, wherein every component executes in parallel per cycle, except for the capabilities listed within each execution unit, which represent said capabilities of each respective execution unit. The reason this is important, is that instructions cannot be executed in parallel if there are dependencies between them, such as conditional jumps introduced by an `if`-statement in the programming language. The solution to this is partially embedded in the CPU in the form of a branch predictor, as can be seen in the front end in Fig. 10. The job of the branch predictor is to choose a code-path before getting a definitive answer to the jump condition. If it predicts correctly, this lets instructions be executed as if it was linear unconditional code. However, there is also a chance of it predicting incorrectly, in which case it has to undo the instructions that were wrongfully executed. This, of course, has a cost. For our target processor, the cost is approximately 16 to 20 cycles[38], depending on if the relevant instructions are still contained within the L1i cache, which can also be seen in Fig. 10. The most effective strategy to minimize this cost is to reduce code branching to the greatest extent possible, particularly in performance-critical code. However, branching is sometimes required, and regarding these branches, they should be as predictable as possible. This could be having a much higher probability of going down one of the code paths. Some compilers let the programmer specify approximate probability of expressions evaluating to true, and the compiled code layout will then be appropriately optimized, such that the branch predictor is more likely to choose the most likely branch[39]. However, this feature is, as of writing, unstable in Rust, and therefore has not been used.

At first glance, the cost of a wasted cycle, might not seem significant. After all, a cycle is only 238 picoseconds for a 4.2 GHz processor. However, calculating the computational potential for each cycle can be a more effective means of communicating its value. This can be done by counting the number of execution units, which have the `FP FMA` capability. For our target processor, two execution units have such a capability as can be observed in Fig. 10. A single `FMA` instruction can perform 8 multiplications and 8 additions simultaneously, for single precision floating point numbers. This means, there is a potential of,

$$2 \cdot (8 \text{ FLOP} + 8 \text{ FLOP})/\text{cycle} = 32 \text{ FLOP}/\text{cycle} \tag{7.1}$$

wasted for a single unused cycle. Therefore, the potential computational value wasted by a branch misprediction could be up to,

$$20 \text{ cycle} \cdot 32 \text{ FLOP}/\text{cycle} = 640 \text{ FLOP}. \tag{7.2}$$

The last, and by far the worst within the domain of compiled languages, is cache misses. This occurs when data required for computation is not within

any of the levels of cache and has to be transferred from the system RAM. The cost of this happening is approximately 256 wasted cycles for a 4.2 GHz Skylake processor[38]. This means the approximate maximum computational value lost is,

$$256 \text{ cycle} \cdot 32 \text{ FLOP/cycle} = 8192 \text{ FLOP}. \tag{7.3}$$

To avoid this happening for every data load, modern processors employ techniques such as having multi-level memory caches, and prefetching.

Multi-level memory caches are memory storage, built into the processor itself, and is much faster to access than system RAM. The capacity of these caches are, however, much lower than system RAM, and the faster the cache, the lower the capacity. A data transfer from one these caches is done 64 bytes at a time[38], in what is called a *cache line*. So to efficiently use these caches, it would be optimal to keep the amount of actively used data low, and keep it as densely packed as possible. Ideally all 64 bytes of a cache line should be relevant to the computation.

Prefetching is the ability to fetch data before it needs to be accessed, by analyzing data access patterns. An example of this would be sequential constantly strided data accesses, which means that there is a constant offset between the locations of consecutive data accesses. The prefetcher would recognize this access pattern and subsequently prefetch the data at the next location in the series. Hence, it is considered ideal to maximize the ordering and predictability of data accesses as much as possible.

Another option is to employ manual prefetching using prefetch intrinsics. However, it should be noted that this approach can potentially impact performance negatively, as excessive prefetching can displace valuable data from the cache.

7.2 Number Representation and Vectorization

As everything is implemented from the ground up, it gives opportunity to make assumptions for the sake of performance. The processor is capable of doing computation on vectors, and one should therefore design their system to fully leverage this fact. If the system would be written purely using scalar mathematics, it might look like the sketch drawn in figure 11. It is drawn to illustrate that vector math potential goes unused, as element, two, three, and four remain empty.

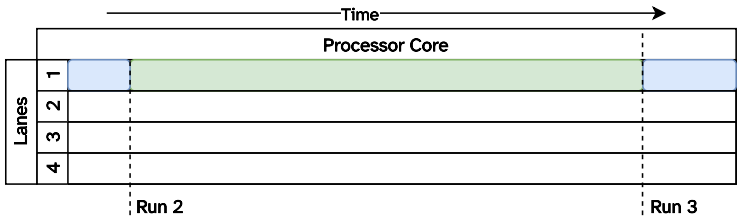


Figure 11

In a more traditional sense, it is common to optimize the slow parts, such as matrix multiplication or even addition. However, sometimes scalar values are needed, and in such a case, no gain can be had from manually optimizing operations using vectors. In other cases, horizontal operations are needed, where the elements of a single vector should be added or multiplied together. This cannot be done in a single instruction. This approach should produce something akin to the sketch in figure 12. Note that there is increased vector utilization, and the total time of a run is shorter than the purely scalar case.

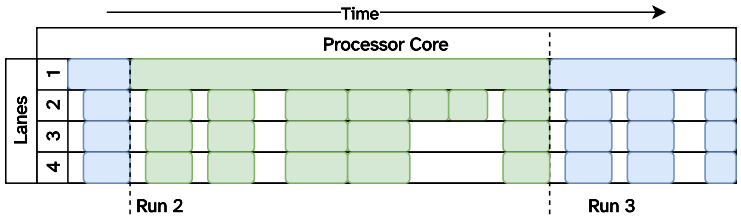


Figure 12

In this case, the knowledge of the total system can be applied to design a more efficient architecture. It is known that there needs to be simulations of trajectories happening multiple times. Additionally, no trajectory can interact with, or require the results of another trajectory. This Monte Carlo-like configuration is known as *embarrassingly parallel*[40]. It is established that there is no branching code when simulating a single trajectory[1]. Consequently, the definition of a number can be reformulated such that every element represents a scalar value for an separate trajectory. This redefinition has been accomplished as follows,

```
pub[2] type Real = std::simd::f64x4;
```

The rest of the mathematical constructs will then look standard on the surface. For example,

[1]Branching does not necessarily exclude vectorization. It would, however, reduce the potential efficiency gains[41].

[2]The Rust keyword `pub` means public, which implies that whatever it is applied to can be used outside of the file in which it is defined.

```
pub struct Complex {
    pub real: Real,
    pub imag: Real
}
```

This means that in reality every element of our operator is in itself a vector. When an operator, vector, or scalar is referred to in the code, it is referring to it for multiple trajectories. Therefore, any operation between any mathematical constructs, will be using vector instructions. The goal is that it would produce results similar to the sketch in Fig. 13. In reality it will not be as perfect as in the sketch for a variety of reasons, such as stack management, histogram computations, and random number calculations.

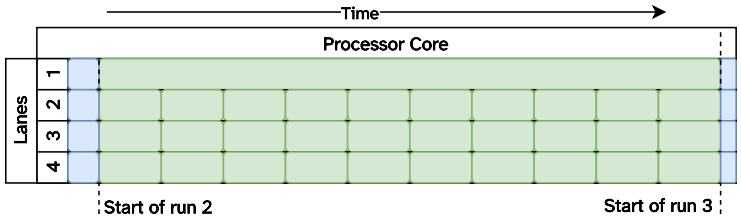


Figure 13

Note that the overall time of a run is longer than even the scalar case, as there is higher execution latency associated with vector instructions. However, due to the chaining of vector instructions, and instruction pipelining, the cost of high instruction latency should be dramatically reduced. Although the time of a run is longer, it is more efficient as it is simultaneously completing four separate trajectories instead of one. The downside of this approach, is that the system would be incapable of simulating a single trajectory, or any number that is not a multiple of the vector size in the definition of `Real`.

7.3 Fused Multiply Add

An important optimization to consider is utilizing the `FMA` instructions. These instructions allow for both vector multiplication and addition to be performed simultaneously within a single instruction, without any increase in latency compared to a single vector multiplication[42]. This means that the vector additions become effectively free if both operations are required. These instructions also have the same throughput as normal vector additions. In figure 10, it can be observed that a floating point fused multiply-add (`FP FMA`) can be performed by the same number of execution units as those capable of performing floating point multiplications. However, this capability is not listed separately because the multiplications are decoded to `FMA` μOPs, wherein zero is added.

7.4 Theoretical Computation Limit of a Core

The theoretical computation limit of a core, at least for our purposes, can be calculated almost purely by looking at the processor block diagram in figure 10. As there are multiple execution units able to perform computation per core, the ones that can execute the required instructions are simply counted. The instructions that this implementation has been based is the `FMA` extension. There are two execution units able to perform those instructions, port 0 and port 1. Now, simply multiplying by the clock speed, which is 4.2 GHz in our case, as well as the number of floating point numbers within a vector, yields:

$$\texttt{perf}_{\texttt{f32}} = 2 \cdot 4.2\text{GHz} \cdot (2 \cdot 8)\ \text{FLOP} = 134.4\ \text{GFLOP/s}. \tag{7.4}$$

Note that the extra 2 within the FLOP number is because there are 2 floating point operations done per `FMA` instruction, a multiply and add[3]. Another thing to note is that this calculation was done with 32 bit floating point numbers in mind. Doing the same for 64 bit floating point numbers would yield half the throughput,

$$\texttt{perf}_{\texttt{f64}} = 2 \cdot 4.2\text{GHz} \cdot (2 \cdot 4)\ \text{FLOP} = 67.2\ \text{GFLOP/s}. \tag{7.5}$$

To get the total computation limit of the whole processor would just be to multiply by the number of cores in the same processor.

However, this limit can never be hit in reality, as useful programs do other things, such as stack management, data loading, data writing, logic, and branching.

7.5 Stochastic Numerical Solutions

The noise term of Eq. (6.30), represents a challenge for computing a solutions to the density matrix. A stochastic equation, in its differential form, written as:

$$\mathrm{d}Y(t) = a(t, Y(t))\mathrm{d}t + b(t, Y(t))\mathrm{d}W(t), \tag{7.6}$$

is known as an Itô process. Itô calculus is more tricky than regular calculus, as key ideas of regular calculus cannot be assumed to hold with a stochastic process. In addition, the Wiener process is stationary, such that the distribution of $W(t)$ is not dependent on the time. The Wiener increment at time $t = t_1$, is also completely independent of the Wiener increment at time $t = t_2$, for any $t_1 \neq t_2$. Given the inherent randomness of the process, it is understood that providing an exact singular solution to Eq. (7.6) is not feasible. Instead, the best that can be hoped for, is the probability distribution of the solutions.

[3]This can also be a subtract.

The noise is independent of the system, and therefore the expectation value of the system, is simply the evolution of the non-stochastic term:

$$\langle Y(t)\rangle = \int_0^\tau a(t, Y(t))\mathrm{d}\tau \tag{7.7}$$

This aligns with the anticipated physical interpretation of the density matrix evolution, as expressed in Eq. (6.30), which expected density evolution is then described by the Lindbladian master equation:

$$\langle \mathrm{d}\rho\rangle = -i\left[\hat{H}, \rho(t)\right]\mathrm{d}t + \mathcal{D}[\hat{c}_{out}]\mathrm{d}t. \tag{7.8}$$

It suggests that irrespective of whether the system divulges its information to a measurement apparatus, the system continues to release that information into its environment. This implies that an observer doesn't need to be sentient.

Modifications to conventional numerical integration schemes are necessary for their application to Itô processes. This requirement primarily arises due to certain characteristics, such as the chain rule, not being compatible with Itô calculus. A viable approach could be to operate directly on the Euler-Maruyama method, which is the stochastic differential equivalence (SDE) of the Euler numerical integration method.

At a cursory glance, one might naively assume that the numerical error of this method is of the order $\mathcal{O}(\Delta t)$, an ordering referred to as strong order convergence. However, this assumption holds true only for the diffusive non-stochastic term. Another scaling of interest is introduced by the noise term, which contributes an additional error to the solution. In the context of the Euler-Maruyama method, this is a weak order convergence of $\mathcal{O}(\sqrt{\Delta t})$. In the interest of performance, it is ideal to avoid the selection of step sizes orders of magnitude smaller just to offset the poor scaling of the numerical method. Opting for a more robust numerical scheme with improved weak-scaling helps. However, it is important to underline that conventional Runge-Kutta schemes do not yield solutions for Itô diffusion; rather, they converge to what is referred to as a Stratonovich integral, which typically yields a different result[43].

A few different numerical schemes have been identified. Simpler schemes tend only to work when the functions commute[44], a trait that is generally untrue for the heterodyne SME, or is complicated when applied to matrices. The method used in this thesis, is a Runge-Kutta scheme which has strong and weak ordering of $\mathcal{O}(\Delta t)$. The numerical solution for finding $Y(t_k+\Delta t) = Y_{k+1}$, where $t_{k+1} - t_k = \Delta t$, is given by the algorithm:

$$k_1 = \Delta t\ a(t_k, Y_k) + \left(\Delta W_k - S_k\sqrt{\Delta t}\right) b(t_k, Y_k), \tag{7.9}$$

$$k_2 = \Delta t\ a(t_{k+1}, Y_k + k_1) + \left(\Delta W_k + S_k\sqrt{\Delta t}\right) b(t_{k+1}, Y_k + k_1), \tag{7.10}$$

$$Y_{k+1} = Y_k + \frac{1}{2}\left(k_1 + k_2\right). \tag{7.11}$$

Here $S_k = \pm 1$, sampled with equal probability. A more comprehensive analysis of the numerical scheme has been done in previous works[45].

7.6 Multi-core parallelism

As explained in an earlier section, the computational task of simulating quantum trajectories is almost entirely parallel, as no trajectory may interact with another. This means, that as well as being able to accelerate the task using vector instructions, it is also trivial to accelerate the task by running it on multiple cores in parallel. The number of cores in a system, is of course easier to scale than the individual core efficiency. In fact, in many-core systems, such as super computers, it can be observed that the processor core clock speed is lower than that of an ordinary desktop processor[46]. Therefore, the ability to scale well to multiple cores is tremendously important, if the goal, apart from simulating correctly, is maximum performance. If the raw data of all trajectories was the only thing to be saved, the simulation could be implemented as entirely parallel. However, it quickly becomes unfeasible, from a data storage perspective, to run simulations for multiple different cavity and qubit configurations. For example, the data storage requirements of a single simulation of 10^5 trajectories with 10^4 steps, for a two qubit system, computed at 32-bit precision, would be,

$$10^5 \cdot 10^4 \cdot 4\text{B} \cdot 4 = 16\text{GB} \tag{7.12}$$

The extra 4 is for the probabilities of each of the possible computational states for a 2 qubit system, 00, 01, 10, and 11.

A more natural approach is to reduce the data to only the needed parts before writing to disk. For example, to get an overview over all trajectories simulated, a 2D histogram is compiled from them, with steps in time being horizontal, and measurement probability being vertical, with labeled rows dedicated for each possible computational state. Meaning the bottom of a row would represent zero probability of measuring the labeled computational state, and the top of the same row would represent a probability of one for the same measurement. An example of this histogram can be seen in figure 14.

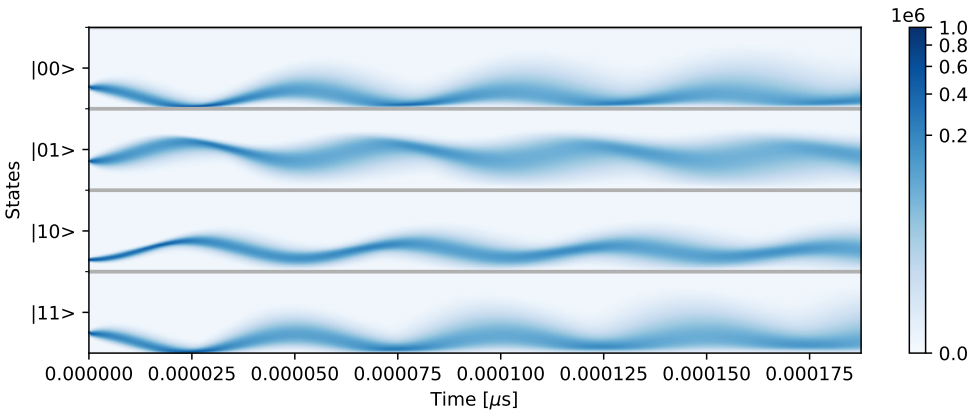


Figure 14: Example histogram of 10^6 trajectories.

This data reduction process is not parallel when combining data from multiple threads. Therefore, the simulator simply keeps a separate histogram for each thread, and simply combines only the histograms at the end. As a consequence, a reduction in the total amount of data that needs to be communicated between threads is facilitated, thereby resulting in increased parallelism, albeit at the expense of higher memory usage.

It may seem inconsequential optimizing small tasks, such as histogram generation, for parallelism, however, small gains in parallelism can provide exponential gains in maximum parallel performance. This is what can be inferred from Amdahl's law, which relates the speedup of a fixed workload to the number of cores and parallel proportion of the task, as follows[47],

$$S = \frac{1}{1 - p + pn^{-1}}, \tag{7.13}$$

where S is the speedup, p is parallel proportion and n is the number of cores. This is plotted in figure 15 to visually demonstrate the scaling.

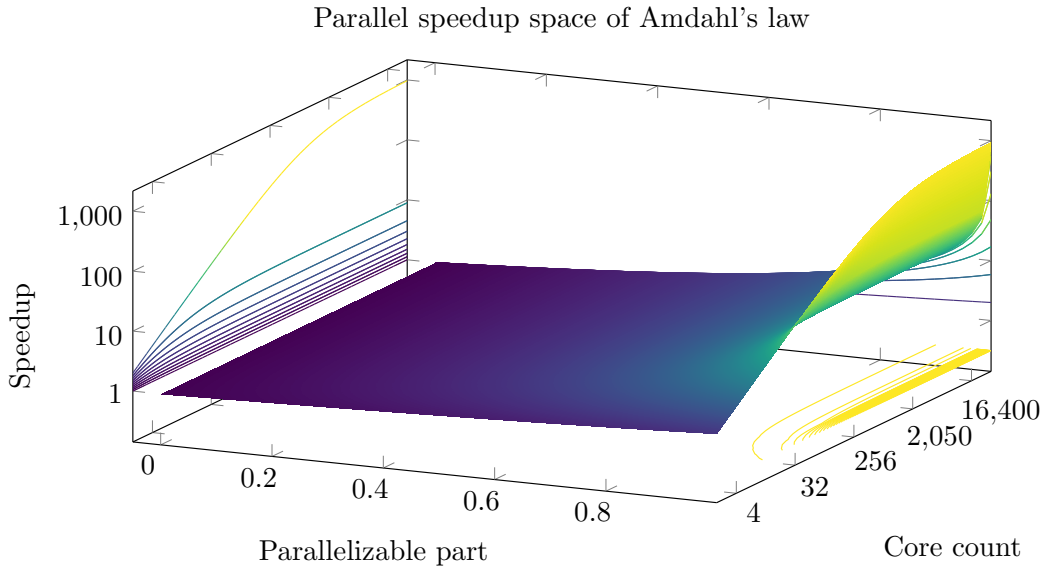


Figure 15: Amdahl's law plotted as a function of (p, n). The core count, n, as well as the speedup, S, are plotted with logarithmic scaling. In the speedup-core count plane are contour lines with interval of 0.1 parallelizable proportion.

7.7 Numerical Error

When computation is done on a processor with non-integer values, it is usually done with floating point numbers. Floating point works as scientific notation in binary, consisting of an exponent for an order of magnitude, a sign, and a fraction. As computation is done in radix two, the order of magnitude

is specified as a power of two. The floating point implementation of our target architecture is based on the IEEE 754 standard[48], giving us 52 bits of fraction, which is approximately 16 decimal digits of precision, and 11 bits of exponent for the order of magnitude. This is number of bits should keep numerical errors low, relative to our needs. However, as results of previous computations get reused for next operations, this error has the ability to add up as the number of chained computations increases. This error should not completely overwhelm the desired result for our purposes.

As demonstrated in subsection 6.4.1, the error induced by the floating point computations could render the system unstable. This is due to the inability to guarantee that the trace of the density matrix is one. Only an approximation can be asserted, $\mathrm{Tr}[\rho] \approx 1$.

The term amplifying our error is,

$$\mathrm{Term}_3(\mathrm{d}\rho(t)) = \mathcal{H}[\mathrm{d}W^*(t)\hat{c}_{out}]\rho(t). \tag{7.14}$$

The floating point error is first introduced into the density matrix factor, $\rho(t)$. Therefore, correcting our error locally here should introduce the least amount of side-effects in our system,

$$\mathcal{H}[\mathrm{d}W^*(t)\hat{c}_{out}]\rho(t) \to \mathcal{H}[\mathrm{d}W^*(t)\hat{c}_{out}]\frac{\rho(t)}{\mathrm{Tr}[\rho(t)]} \tag{7.15}$$

In the implementation, a less precise method of normalizing the density matrix between integration steps was opted for. This decision was based on the employed numerical integration scheme, which would necessitate the computation of the trace and division twice for each step, rather than once.

$$\rho(t) \to \frac{\rho(t)}{\mathrm{Tr}[\rho(t)]} \tag{7.16}$$

This adds error to the coherence elements of the density matrix as the error of those elements do not necessarily have the same sign as the errors of the diagonal. It was determined that this additional minor error was qualitatively acceptable, as the stability of the system was regained with less computation needed.

7.7.1 Rapid oscillations and high Δt

In our stochastic master equation, the term,

$$\mathrm{Term}_1(\mathrm{d}\rho(t)) = -i\left[\hat{H}, \rho(t)\right]\mathrm{d}t, \tag{7.17}$$

induces a frequency which can potentially be multiple orders of magnitude higher than the rest of the terms. This means that Δt has to be set to a number which is much lower than the length of a period, such that the oscillations of this term can be captured properly in our results.

Ideally the Δt should be set as high as possible while still giving accurate results. To increase the maximum Δt value the error of this term alone, had to be removed. This has been done by solving that term analytically. By ansatz the solution is found,

$$\rho(t) = e^{-iHt}\rho(0)e^{iHt}. \tag{7.18}$$

Using the analytical solution for this term alone, allows Δt to be orders of magnitude higher in some more extreme cases while maintaining identical results.

7.8 Potential improvements

As with any system, there are flaws and imperfections. Here are some potential improvements that have not been implemented.

- When writing computation results to memory there is an opportunity to use stream intrinsics, which would let us write to memory without loading that memory into cache. This would make it less likely that a cache miss happens when loading data which is actually needed for computation.
- When computing histogram data there is an opportunity to use gather and scatter intrinsics rather than have reading and writing of memory be with scalar instructions. However, scatter intrinsics are in the `AVX512` family of instruction set extensions, which is beyond the capabilities of our target hardware.
- In our system the results of all threads are waited for and then combined in a single thread. This could be improved by combining the result data on multiple threads. This would improve scalability when increasing core count.
- The plotting is done using a separate plotting program written in python which loads the result files from the simulation, and plots it using `matplotlib`. This could be improved by using a Rust-native plotting library, as the simulation data would already be in memory. This would also give the option to not write the data to disk at all.
- There could be a graphical user interface, making it faster to iterate through different system configurations, including changing the quantum circuit which is applied to the qubits.
- When scaling up to more qubits, it becomes less likely that it will fit in cache, and it could therefore be beneficial to manually place prefetch intrinsics, when it is known that memory accesses might not be contiguous, such as in operator/matrix multiplication.

Some performance metrics of our final implementation can be found in appendix A.

8 Results

The subsequent section is dedicated to the presentation and analysis of examples of data extracted from the simulation. Natural units of $\hbar = 1$, are used throughout the simulations, to simplify expressions, and because the probability densities, remains unaltered. The simulation model provides a powerful framework to quickly and efficiently obtain new data, due to the optimization considerations detailed in Sec. 7.

8.1 First simulations

Unless indicated otherwise, the simulation is run on a default set of values, selected to mirror physically realistic values employed in experimental setups. It is assumed that the entirety of cavity decay transpires to the measured signal, thereby setting $\kappa_L = 0$, and $\kappa_1 = \kappa$. This premise ensures that the detection losses can be solely attributed to detector inefficiencies, which are assumed minimal, setting the detector efficiency at $\eta = 0.96$. The relative phase is chosen such that $\phi = 0$, allowing for the integrated current to be symmetrically separated about the $\hat{x}$ quadrature. If the complex part of β, is incorporated into the phase ϕ, then without loss of generality, β is assumed to be real, with the default value chosen as $\beta = 4\sqrt{\text{MHz}}$. To amplify the strength of the outgoing field, the cavity is probed in proximity to the resonator frequency, with $\Delta_{dr} = 50\text{MHz} \approx 0$. Furthermore, to validate the dispersive regime, $\chi_0 = 0.6\text{MHz} = 0.06g$ is chosen, corresponding to $\Delta_{dq} - \Delta_{dr} = \omega_q - \omega_r = g^2/\chi_0 \gg g$.

Depictions of measurements for a two-qubit system across 4000 trajectories can be found in Figures 16 and 17. The initial state subject to measurement is an entangled state, defined as follows:

$$|\psi\rangle = \frac{1}{2}\left(|00\rangle + \sqrt{\frac{3}{2}}\,|01\rangle + \frac{1}{\sqrt{2}}\,|10\rangle + |11\rangle\right). \tag{8.1}$$

The dispersive shifts of the two qubits are specified as $\chi_1 = \chi_0 + 0.1\text{MHz}$ and $\chi_2 = \chi_0 - 0.1\text{MHz}$. The separation between χ_1 and χ_2 is necessary to distinguish between the two states of $|01\rangle$ and $|10\rangle$ in the integrated signal. The simulation illustrates that an extension of the measurement duration is instrumental in projecting a larger number of trajectories onto the classical computational basis states. This observation is further corroborated by the integrated currents, seen from the difference between the figures 16a-16d and

17a-17d. The separation between the qubit state distributions expands linearly with time, while their standard deviation increases as $\sqrt{t}$, in accordance with the Wiener increments.

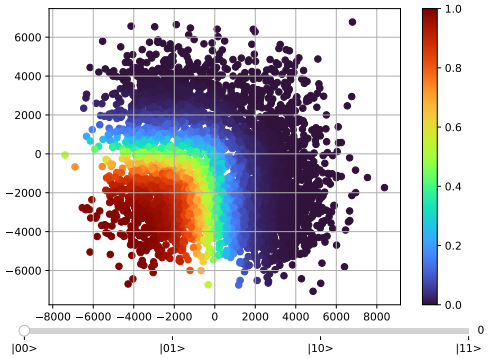
8.2 Effects of the dispersive shift

Envision a scenario involving the measurement of a single qubit undergoing a gate operation. The readout procedure will project the qubit onto its basis state, thereby introducing error of the gate operation. To circumvent this, the qubit could be significantly detuned from the cavity, resulting in a decrease in χ which causes the ac-Stark term, $\chi\hat{a}\hat{a}^\dagger$, to reduce. The utility of such an operation might lie in the potential for parallelization of the computation, allowing for computation to be performed on one subsystem while simultaneously enabling the readout of another. This can be achieved by driving the qubit with a square pulse, implementing the following Hamiltonian:

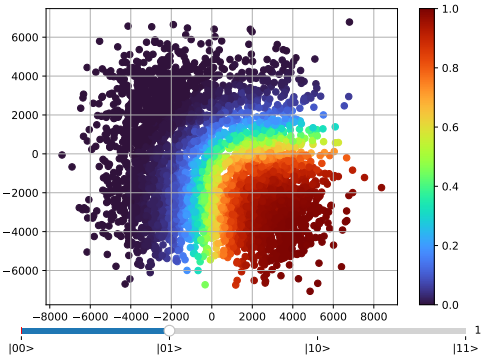
$$\hat{H} = \Omega\hat{\sigma}_x. \tag{8.2}$$

When the Hamiltonian is run for a time $t = \pi/(2\Omega)$, the qubit, assuming no decay, has successfully completed an $\hat{\mathsf{X}}$ gate operation. Adding Eq. (8.2) to the dispersive Hamiltonian, gives a description for the dynamics of the measured qubit under rotations about x. When the detuning between the qubit frequency, and the resonator frequency, $\Delta = \omega_s - \omega_r$, decreases, then the measurement of the qubit increases. Fig. 18a illustrates the fidelity of the gate, as a function of the dispersive shift, χ, performed on a single qubit. It is evident from the figure, that the fidelity doesn't tend towards zero at higher values of χ. The dispersive shift works as a sort of detuning, and therefore at large values of χ de-excites the cavity away from the resonance, reducing the strength of the measurement. When adding another qubit into the cavity, while still performing a $\hat{\mathsf{X}}$ gate on the first, the interaction term destroys the fidelity of the gate, as dipicted in Fig. 18b.

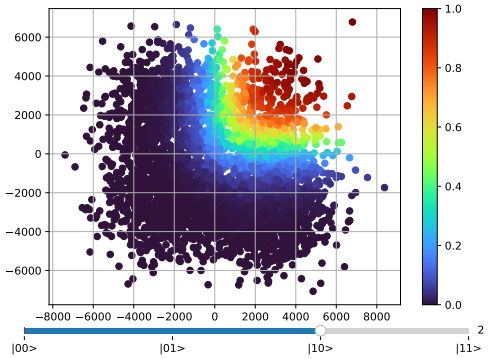
To then properly run gates on one qubit, both qubits will have to be measured only weakly, which makes the parallelization of the readout on one qubit, and computation on other qubits difficult. Additionally, this scheme does not allow gate operation between two qubits, which are required for achieving quantum supremacy. One can circumvent this issue, by coupling qubits to multiple resonators, such that measurement can be performed independently on a subset of qubits[49].



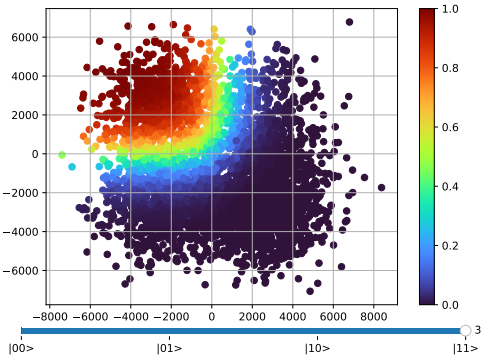
(a) State $|00\rangle$ currents.



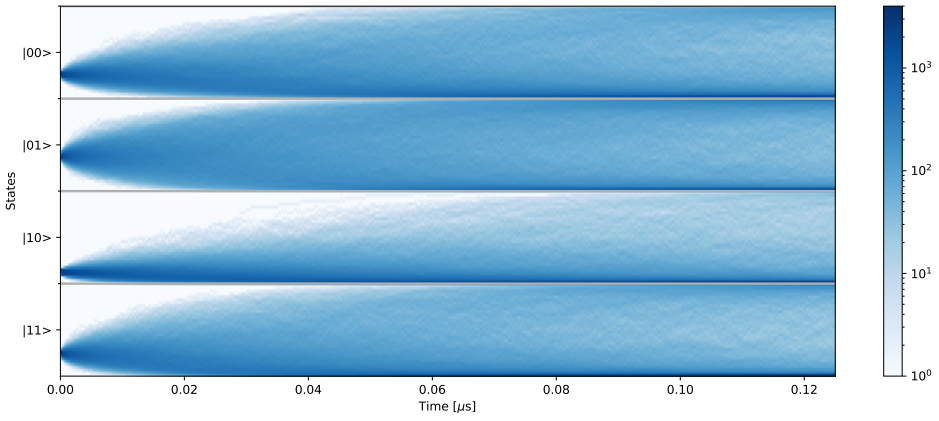
(b) State $|01\rangle$ currents.



(c) State $|10\rangle$ currents.

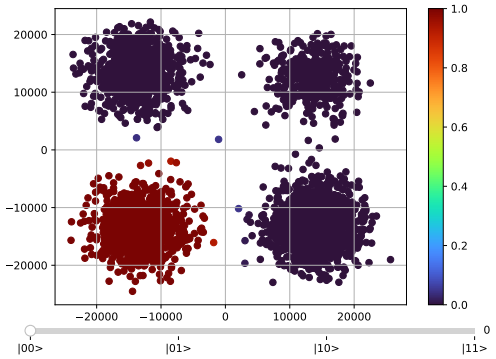


(d) State $|11\rangle$ currents.

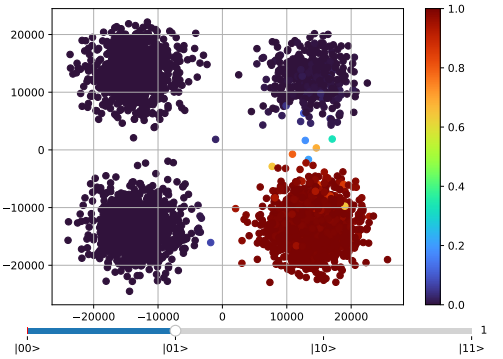


(e) All 4000 trajectories.

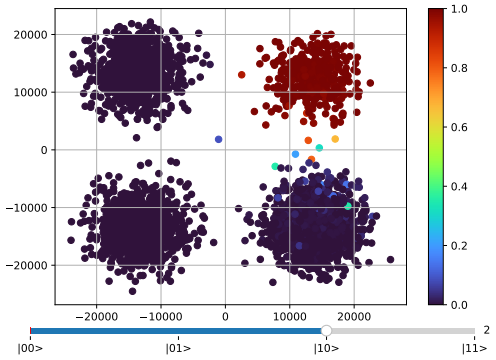
Figure 16: 4000 trajectory simulations and their simulated integrated currents, on a time duration of $0.125\mu s$. The colors in the current plots represent the probability of measuring the respective computational state.



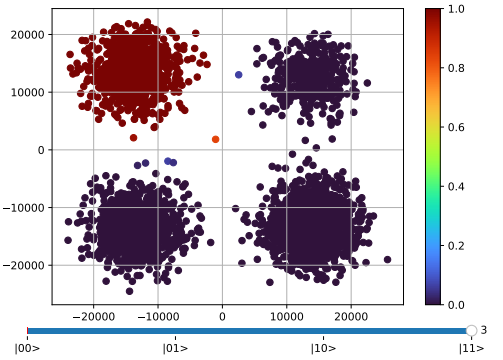
(a) State $|00\rangle$ currents.



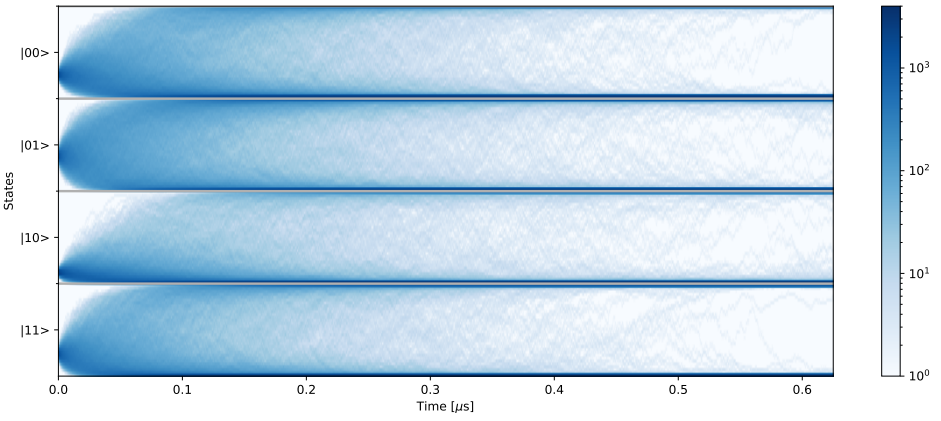
(b) State $|01\rangle$ currents.



(c) State $|10\rangle$ currents.



(d) State $|11\rangle$ currents.



(e) All 4000 trajectories.

Figure 17: 4000 trajectory simulations and their simulated integrated currents, on a time duration of $0.625\mu s$. The colors in the current plots represent the probability of measuring the respective computational state.

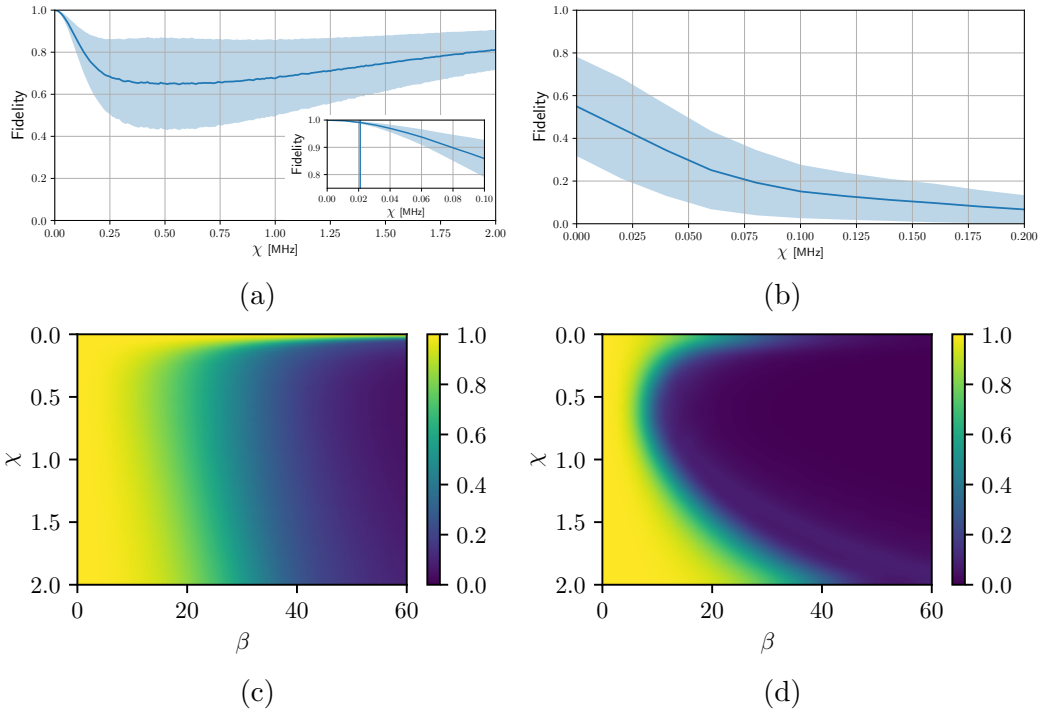


Figure 18: (a) and (b) depicts the mean fidelity and standard deviation of the $\hat{\mathsf{X}}$ gate from 12 800 trajectories, when varying χ for 200 samples. The inserted subfigure in (a) is a zoomed in version, with the 0.99 fidelity market by the blue line. The two figures are plotted with a $\beta = 21\sqrt{\text{MHz}}$, in a rotating frame moving with the frequency of the qubit under gate operation. (c) and (d) illustrates the fidelity while varying both χ and β with 4 000 and 400 samples for (c) and (d) respectively. Note in (d), that the fidelity dips when the qubits are at resonance to each other. (a) and (c) represents the fidelity of a single qubit in the cavity, while (b) and (d) represents two qubits, with the second qubits χ held constant. The Ω frequency of the gate operation is 1 GHz.

If the state of a system is completely unknown, i.e. the state is described by a completely mixed state density matrix, then measurement should update an observer's knowledge about the state. To quantify this ignorance, consider the purity of the state, defined by $P = \mathrm{Tr}(\rho^2)$. For a completely mixed state density matrix, with k qubits, and 2^k distinct basis states, the purity at the time when measurement commences is:

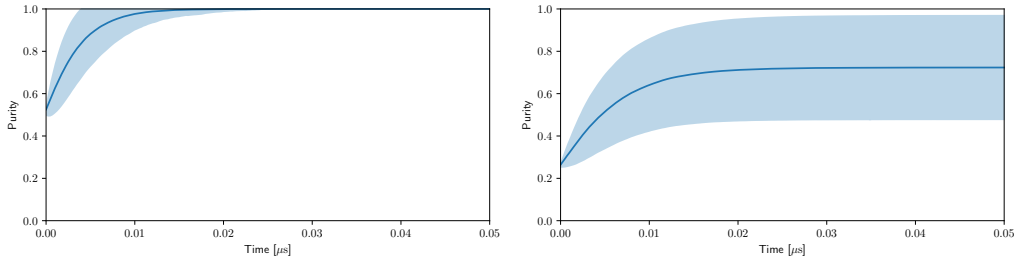
$$P = \mathrm{Tr}\big(\rho^2\big) = \mathrm{Tr}\left(\frac{1}{2^{2k}} I\right) = \frac{1}{2^k}. \tag{8.3}$$

Measurement by heterodyne detection, has a limit to the information that can be extracted for a set of qubits placed within the same resonator cavity, with the same frequency of each of the qubits. The qubits are distinguished by the ac-Stark shift, $\chi_s \hat{a}\hat{a}^\dagger \hat{\sigma}_z^{(s)}$. If the value of χ is the same for each qubit, then two computational basis states, $|a\rangle$ and $|b\rangle$, are mutually indistinguishable (degenerate), if the binary representation of one is a permutation of the other's digits. In this case they will experience the same state dependent shift, contributing to the current equally and providing no distinguishing information. An example to this are the indistinguishable states $|01\rangle$ and $|10\rangle$, because their corresponding eigenvalue when operating on the ac-Stark shift is the same:

$$\sum_s \chi \hat{a}\hat{a}^\dagger \hat{\sigma}_z^{(s)} |01\rangle \otimes |n\rangle = \lambda |01\rangle \otimes |n\rangle , \tag{8.4}$$

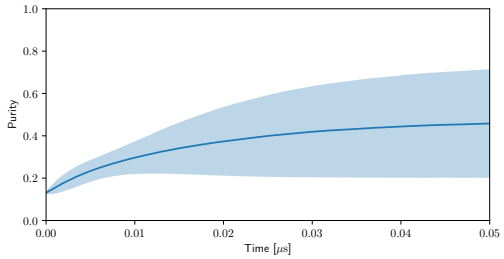
$$\sum_s \chi \hat{a}\hat{a}^\dagger \hat{\sigma}_z^{(s)} |10\rangle \otimes |n\rangle = \lambda |10\rangle \otimes |n\rangle , \tag{8.5}$$

where $\lambda = (\chi - \chi)n = 0$. That means that the mixed state density matrix $\rho = \sum_m^{2^k} |m\rangle\langle m|$ will half the time evolve to a pure state density matrix, where $p_{00} \to 1$, or $p_{11} \to 1$, and the other half the time, will evolve into a mixed state density matrix, where $p_{01} = p_{10} = 1/2$, due to the degeneracy of the two states, where k is the number of qubits. Therefore the average purity can at most be determined to be $\langle P_{max} \rangle < 2^{-k}(k+1)$. This is illustrated in Fig. 19, for 1, 2, 3 and 4 qubits in the cavity. When the χ's are varied it becomes possible to properly distinguish between every qubit state, with greater variation, leading to faster distinction between the states. Note that the standard deviation does not capture the idea, that some trajectories, no matter the number of qubits in the resonator, will obtain a maximum purity of one. This is demonstrated in Fig. 20, where the two $\chi_\pm = \chi_0 \pm \Delta\chi/2$ is varied. To properly distinguish between the two states over a measurement time of 0.625μs, the difference has to be about $\Delta\chi = 0.04$MHz.

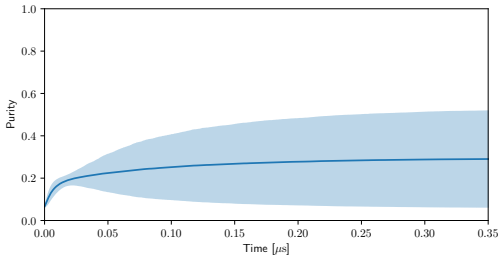


(a) Single qubit

(b) Two Qubits



(c) Three Qubits



(d) Four Qubits

Figure 19: The average purity and standard deviation from 4 480 trajectories of a cavity with one (a), two (b), three (c), and four (d) qubits.

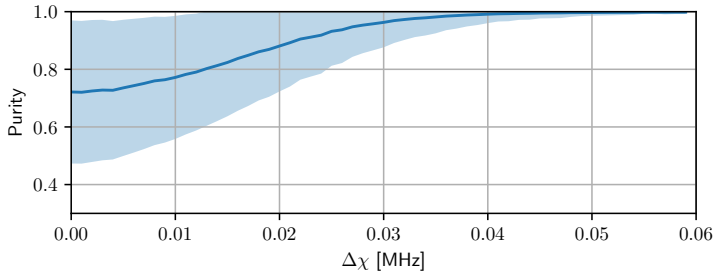


Figure 20: The maximum purity after a time of 0.625μs from 12 800 simulated trajectories, over 60 samples of $\Delta\chi$, with the probing strength of $\beta = 15\sqrt{\text{MHz}}$. With number of qubits, $k = 2$. The shaded area corresponds to one standard deviation from the mean, and the initial state of the system was prepared in a completely mixed state, $\rho_{initial} = 2^{-k}I$.

8.3 The Zeno effect

An interesting property of the simulation, is seen when the system is subject to strong measurements, while also under the operation of a gate. The application is not what is strongly considered here, which has been analyzed in previous works[50]. The measurements during gate operation collapses the state, before the desired unitary operation can be executed. This exercise is to gain insight into the behavior of a driven qubit, under the influence of strong measurements. The expectation is, that the time evolution of the system, will experience quantum Zeno effect. To comprehend how, consider a qubit under

the influence of $\hat{R}_y$ rotations, with the Hamiltonian:

$$\hat{H} = \Omega\hat{\sigma}_x \tag{8.6}$$

The state evolution of the system given by:

$$|\psi(t)\rangle = e^{-i\hat{H}t}\,|\psi(0)\rangle = (\cos(\Omega t)I - i\sin(\Omega t)\hat{\sigma}_x)\,|\psi(0)\rangle \tag{8.7}$$

Or in terms of the state coefficients:

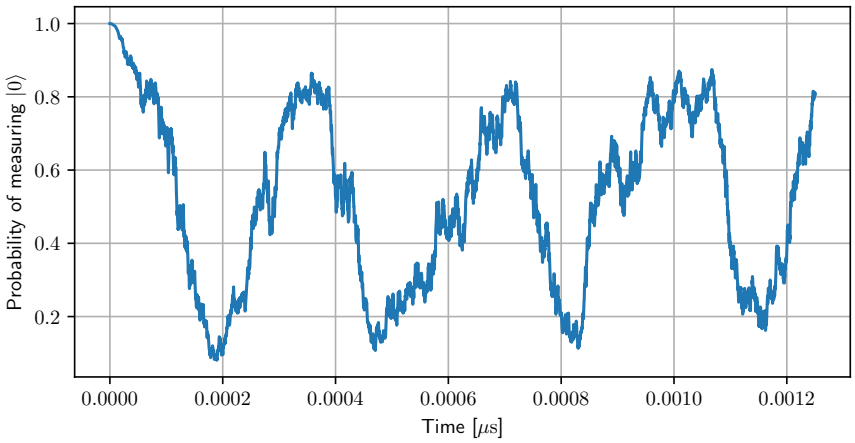
$$a(t) = \cos(\Omega t)a_0 - i\sin(\Omega t)b_0, \tag{8.8}$$

$$b(t) = \cos(\Omega t)b_0 - i\sin(\Omega t)a_0. \tag{8.9}$$

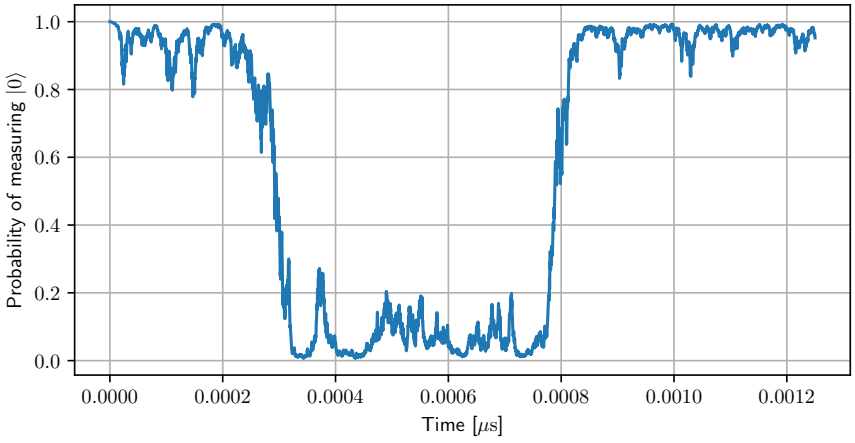
Suppose the state is initialized to $a(t=0) = a_0 = 1$, and instead of heterodyne detection, the qubit was measured directly such that a measurement resulted in a projection onto the basis-states, in accordance with the quantum jump formalism. If n measurements are made, equally spaced during the time T, then the probability of each measurement collapsing the state into $|0\rangle$, after time T, is given by $P(m=0) = |\cos(\Omega T/n)|^{2n}$. In the limit for large n this probability will approach unity, and the constant measurement would have frozen the evolution of the system.

$$\begin{aligned} P(x=0) &= \lim_{n\to\infty} |\cos(\Omega T/n)|^{2n} = \lim_{n\to\infty} \left|1 - \tfrac{1}{2}\left(\tfrac{\Omega T}{n}\right)^2\right|^{2n} \\ &= \lim_{n\to\infty} \left|\left(1 - \tfrac{1}{\sqrt{2}}\tfrac{\Omega T}{n}\right)^n \left(1 + \tfrac{1}{\sqrt{2}}\tfrac{\Omega T}{n}\right)^n\right|^2 = e^{\frac{1}{\sqrt{2}}\frac{\Omega}{n}} e^{-\frac{1}{\sqrt{2}}\frac{\Omega}{n}} = 1 \end{aligned} \tag{8.10}$$

Under regular measurements, the time that the system is expected to be in the ground state is proportional to $\tau \propto n/\Omega$. In the case of heterodyne detection, where each measurement isn't a direct projection, the Zeno regime is reached when the cavity is strongly probed and the time it takes to project the trajectory onto its computational basis states, is much faster than the rabi oscillation. Similarly, the time that the system is expected to remain primarily in a specific state, is proportional to $\tau \propto \beta/\Omega$. This is illustrated in Fig. 21a and Fig. 21b, where the strong probing disturbs the Rabi oscillations.



(a) $\beta = 60\sqrt{\text{MHz}}$



(b) $\beta = 140\sqrt{\text{MHz}}$

Figure 21: A single trajectory in the Zeno regime, under different probing strengths. The qubit is subject to Rabi oscillations, of the frequency $\Omega = 10\text{GHz}$, and are prepared in an initial state $|\psi\rangle = |0\rangle$.

8.4 State prediction from measurement records

Another use of the stochastic master equation, given that the physical constants of the system are known, is to feed it a measurement record from the physical system. The simulator implements the SME, allowing it to be used as an analytical tool for experimental data. It can be used to compute the probabilities for the physical system being in certain computational states. This can simply be done, by solving for the random values within the current calculation,

$$\mathrm{d}I_x = \langle x \rangle + \frac{\sqrt{2}}{\mathrm{d}t}\mathrm{d}W_x \rightarrow \mathrm{d}W_x = \frac{\mathrm{d}t}{\sqrt{2}}(\mathrm{d}I_x - \langle x \rangle) \tag{8.11}$$

$$\mathrm{d}I_y = \langle y \rangle + \frac{\sqrt{2}}{\mathrm{d}t}\mathrm{d}W_y \rightarrow \mathrm{d}W_y = \frac{\mathrm{d}t}{\sqrt{2}}(\mathrm{d}I_y - \langle y \rangle). \tag{8.12}$$

This is built into the simulator as an option. For each trajectory simulated in Fig. 22, a corresponding measurement record has been generated. The simulations here are done with $\beta = 50\sqrt{\text{MHz}}$, and $3 \cdot 10^4$ steps.

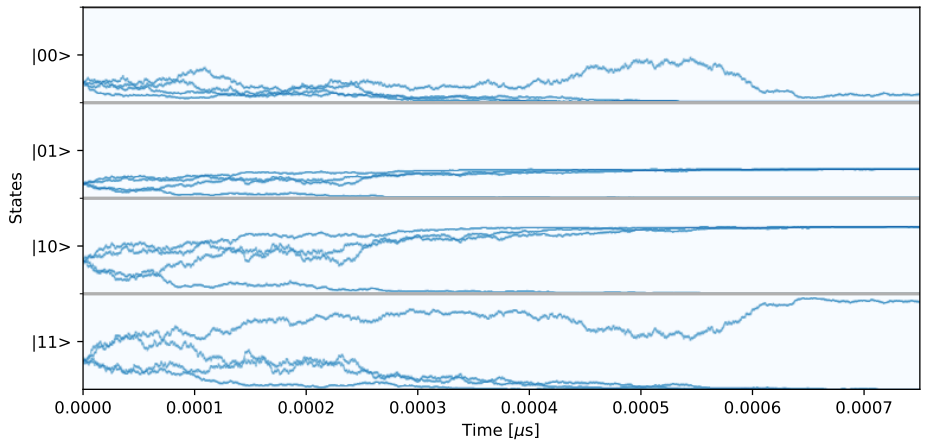


Figure 22: Four example trajectories, which could have physically occurred in the lab, with which current measurement records are generated.

As for the initial state when simulating with specific measurement records, it can simply be set to the completely mixed density matrix, i.e. the normalized identity matrix, $\rho_{initial} = 2^{-k}I$, where k is the number of qubits. This indicates that the observer has no knowledge of the system as it could equally be in any state. Feeding the measurement records from the trajectories generated in Fig. 22, generates new trajectories as can be observed in Fig. 23. These new trajectories represent the knowledge of the original system, given that only information in the form of current measurement records had been extracted, which is the case for real physical systems.

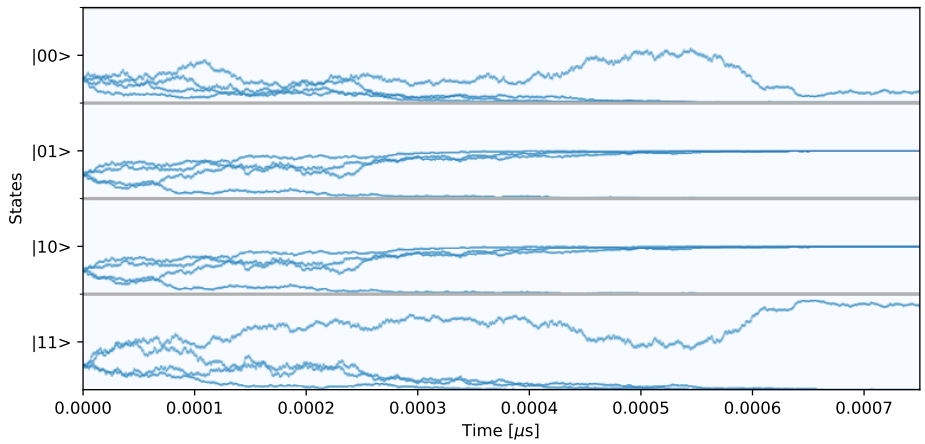


Figure 23: Four recreated trajectories from the current measurement records which were generated from the trajectories in figure 22.

A notable feature is that, while the states $|00\rangle$ and $|11\rangle$ are recreated nearly perfectly, the states $|01\rangle$ and $|10\rangle$ look identical. This is again is again due to identical values of χ, which means that they are indistinguishable in the measurement record, as described at the end of section 8.2. So both indistinguishable states in the reconstructed trajectory would in this case be an average of the two states.

9 Conclusion

This thesis presented a simulator for the stochastic evolution of a qubit system coupled to a resonator. Designed with scalability in mind, the simulator can efficiently simulate a large number of quantum trajectories. It does this by leveraging parallel computing through vector instructions and multi-core processing, facilitating the exploration of higher state dynamics. The simulation conditions include continuous heterodyne detection when tuning frequencies to the dispersive regime.

An analysis of the effects of the dispersive shift and the strength of the resonator probing was carried out using the simulator. It highlighted the degeneracy between certain computational states and proposed a method to assess the impact of varying the separation between qubit frequencies (i.e., changing χ). Furthermore, the simulation demonstrated the manifestation of the quantum Zeno effect when the cavity is strongly probed and the system is subjected to a rotation about x.

The simulator, can be seamlessly integrated with experimental data in the form of current measurement records. As it implements the stochastic master equation presented in this thesis, it has the capacity to solve the stochastic components utilizing the provided experimental data. This integration allows for the extraction of all possible information from the signal, thereby providing insightful predictions of quantum states. This conjunction of simulated and experimental data can significantly enhance our understanding of quantum systems.

The simulator's capability can be extended to incorporate two-qubit gates into the simulation, such as the CZ gate, due to its inherent flexibility in modifying the Hamiltonian and including additional states. This adaptability positions the simulator as a valuable tool for advancing understanding and exploration in quantum computing. Furthermore, it can implement entire quantum circuits and used for the the analysis of error correcting schemes upon such circuits.

References

[1] Hannes Bernien et al. "Probing many-body dynamics on a 51-atom quantum simulator". In: *Nature* 551.7682 (Nov. 2017), pp. 579–584. DOI: 10.1038/nature24622. URL: https://doi.org/10.1038%2Fnature24622.

[2] Howard M. Wiseman and Gerard J. Milburn. *Quantum Measurement and Control*. Cambridge University Press, 2009. DOI: 10.1017/CBO9780511813948.

[3] David P. DiVincenzo. "The Physical Implementation of Quantum Computation". In: *Fortschritte der Physik* 48.9-11 (Sept. 2000), pp. 771–783. DOI: 10.1002/1521-3978(200009)48:9/11<771::aid-prop771>3.0.co;2-e. URL: https://doi.org/10.1002%2F1521-3978%28200009%2948%3A9%2F11%3C771%3A%3Aaid-prop771%3E3.0.co%3B2-e.

[4] Prakash Murali et al. *Architecting Noisy Intermediate-Scale Trapped Ion Quantum Computers*. 2020. arXiv: 2004.04706 [quant-ph].

[5] Loïc Henriet et al. "Quantum computing with neutral atoms". In: *Quantum* 4 (Sept. 2020), p. 327. DOI: 10.22331/q-2020-09-21-327. URL: https://doi.org/10.22331%2Fq-2020-09-21-327.

[6] M. Kiczynski et al. "Engineering topological states in atom-based semiconductor quantum dots". In: *Nature* 606.7915 (June 2022), pp. 694–699. ISSN: 1476-4687. DOI: 10.1038/s41586-022-04706-0. URL: https://doi.org/10.1038/s41586-022-04706-0.

[7] Panagiotis Kotetes, Gerd Schön, and Alexander Shnirman. "Engineering and manipulating topological qubits in 1D quantum wires". In: *Journal of the Korean Physical Society* 62.10 (May 2013), pp. 1558–1563. DOI: 10.3938/jkps.62.1558. URL: https://doi.org/10.3938%2Fjkps.62.1558.

[8] Dominik Niemietz et al. "Nondestructive detection of photonic qubits". In: *Nature* 591.7851 (Mar. 2021), pp. 570–574. ISSN: 1476-4687. DOI: 10.1038/s41586-021-03290-z. URL: https://doi.org/10.1038/s41586-021-03290-z.

[9] R. Hanson, O. Gywat, and D. D. Awschalom. "Room-temperature manipulation and decoherence of a single spin in diamond". In: *Physical Review B* 74.16 (Oct. 2006). DOI: 10.1103/physrevb.74.161203. URL: https://doi.org/10.1103%2Fphysrevb.74.161203.

[10] Sanskriti Joshi and Sajjad Moazeni. *Scaling up Superconducting Quantum Computers with Cryogenic RF-photonics*. 2022. arXiv: 2210.15756 [quant-ph].

[11] S. Krinner et al. “Engineering cryogenic setups for 100-qubit scale superconducting circuit systems”. In: *EPJ Quantum Technology* 6.1 (May 2019). DOI: 10.1140/epjqt/s40507-019-0072-0. URL: https://doi.org/10.1140%2Fepjqt%2Fs40507-019-0072-0.

[12] M. H. Devoret, A. Wallraff, and J. M. Martinis. *Superconducting Qubits: A Short Review*. 2004. DOI: 10.48550/ARXIV.COND-MAT/0411174. URL: https://arxiv.org/abs/cond-mat/0411174.

[13] Aaron M. Kraft, Christoph Rupprecht, and Yau Chuen Yam. “Superconducting Quantum Interference Device (SQUID)”. In: 2017.

[14] M. D. Hutchings et al. “Tunable Superconducting Qubits with Flux-Independent Coherence”. In: *Phys. Rev. Appl.* 8 (4 Oct. 2017), p. 044003. DOI: 10.1103/PhysRevApplied.8.044003. URL: https://link.aps.org/doi/10.1103/PhysRevApplied.8.044003.

[15] Jeffrey Birenbaum. “The C-shunt Flux Qubit: A New Generation of Superconducting Flux Qubit”. In: 2014.

[16] Marco Cattaneo and Gheorghe Sorin Paraoanu. “Engineering Dissipation with Resistive Elements in Circuit Quantum Electrodynamics”. In: *Advanced Quantum Technologies* 4.11 (Sept. 2021), p. 2100054. DOI: 10.1002/qute.202100054. URL: https://doi.org/10.1002%2Fqute.202100054.

[17] Aaron Somoroff et al. *Millisecond coherence in a superconducting qubit*. 2021. DOI: 10.48550/ARXIV.2103.08578. URL: https://arxiv.org/abs/2103.08578.

[18] Matthias Steffen. “Superconducting Qubits Are Getting Serious”. In: *Physics* 4 (Dec. 2011). DOI: 10.1103/physics.4.103. URL: https://doi.org/10.1103/physics.4.103.

[19] Brian Julsgaard and Klaus Mølmer. “Measurement-induced two-qubit entanglement in a bad cavity: Fundamental and practical considerations”. In: *Phys. Rev. A* 85 (3 Mar. 2012), p. 032327. DOI: 10.1103/PhysRevA.85.032327. URL: https://link.aps.org/doi/10.1103/PhysRevA.85.032327.

[20] Peter W. Shor. “Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer”. In: *SIAM Journal on Computing* 26.5 (Oct. 1997), pp. 1484–1509. DOI: 10.1137/s0097539795293172. URL: https://doi.org/10.1137%2Fs0097539795293172.

[21] Lov 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*. ACM Press, 1996. DOI: 10.1145/237814.237866. URL: https://doi.org/10.1145%2F237814.237866.

[22] Anthony Zee. *Group theory in a nutshell for physicists*. en. In a Nutshell. Princeton, NJ: Princeton University Press, Mar. 2016.

[23] Daniel Gottesman. *The Heisenberg Representation of Quantum Computers*. 1998. arXiv: quant-ph/9807006 [quant-ph].

[24] Philip Krantz et al. "A quantum engineer's guide to superconducting qubits". In: *Applied Physics Reviews* 6 (June 2019), p. 021318. DOI: 10.1063/1.5089550.

[25] Evan Jeffrey et al. "Fast Accurate State Measurement with Superconducting Qubits". In: *Phys. Rev. Lett.* 112 (19 May 2014), p. 190504. DOI: 10.1103/PhysRevLett.112.190504. URL: https://link.aps.org/doi/10.1103/PhysRevLett.112.190504.

[26] Peng Zhao et al. *Quantum crosstalk analysis for simultaneous gate operations on superconducting qubits*. 2022. arXiv: 2110.12570 [quant-ph].

[27] M. Werninghaus et al. "Leakage reduction in fast superconducting qubit gates via optimal control". In: *npj Quantum Information* 7.1 (Jan. 2021), p. 14. ISSN: 2056-6387. DOI: 10.1038/s41534-020-00346-2. URL: https://doi.org/10.1038/s41534-020-00346-2.

[28] Guillermo González-García, Rahul Trivedi, and J. Ignacio Cirac. "Error Propagation in NISQ Devices for Solving Classical Optimization Problems". In: *PRX Quantum* 3.4 (Dec. 2022). DOI: 10.1103/prxquantum.3.040326. URL: https://doi.org/10.1103%2Fprxquantum.3.040326.

[29] Liangyu Chen et al. *Transmon qubit readout fidelity at the threshold for quantum error correction without a quantum-limited amplifier*. 2022. arXiv: 2208.05879 [quant-ph].

[30] Jens Koch et al. "Charge-insensitive qubit design derived from the Cooper pair box". In: *Physical Review A* 76.4 (Oct. 2007). DOI: 10.1103/physreva.76.042319. URL: https://doi.org/10.1103%2Fphysreva.76.042319.

[31] David J. Griffiths and Darrell F. Schroeter. *Introduction to Quantum Mechanics*. 3rd ed. Cambridge University Press, 2018. DOI: 10.1017/9781316995433.

[32] Philip Pearle. "Simple derivation of the Lindblad equation". In: *European Journal of Physics* 33.4 (Apr. 2012), pp. 805–822. DOI: 10.1088/0143-0807/33/4/805. URL: https://doi.org/10.1088%2F0143-0807%2F33%2F4%2F805.

[33] Jean Dalibard, Yvan Castin, and Klaus Mølmer. "Wave-function approach to dissipative processes in quantum optics". In: *Phys. Rev. Lett.* 68 (5 Feb. 1992), pp. 580–583. DOI: 10.1103/PhysRevLett.68.580. URL: https://link.aps.org/doi/10.1103/PhysRevLett.68.580.

[34] E Brion, L H Pedersen, and K Mølmer. "Adiabatic elimination in a lambda system". In: *Journal of Physics A: Mathematical and Theoretical* 40.5 (Jan. 2007), pp. 1033–1043. DOI: 10.1088/1751-8113/40/5/011. URL: https://doi.org/10.1088%2F1751-8113%2F40%2F5%2F011.

[35] Rui Pereira et al. “Energy efficiency across programming languages: how do energy, time, and memory relate?” In: Oct. 2017, pp. 256–267. DOI: 10.1145/3136014.3136031.

[36] David Schor et al. *Core i7-7700K - Intel*. Apr. 2019. URL: https://en.wikichip.org/wiki/intel/core_i7/i7-7700k (visited on 05/22/2023).

[37] Andreas Abel and Jan Reineke. “uops.info: Characterizing Latency, Throughput, and Port Usage of Instructions on Intel Microarchitectures”. In: *ASPLOS*. ASPLOS ’19. Providence, RI, USA: ACM, 2019, pp. 673–686. ISBN: 978-1-4503-6240-5. DOI: 10.1145/3297858.3304062. URL: http://doi.acm.org/10.1145/3297858.3304062.

[38] Igor Pavlov. *7-Zip LZMA Benchmark - Intel Skylake*. 2023. URL: https://www.7-cpu.com/cpu/Skylake.html (visited on 05/22/2023).

[39] GCC team et al. *Other Built-in Functions Provided by GCC*. Apr. 2023. URL: https://gcc.gnu.org/onlinedocs/gcc/Other-Builtins.html (visited on 05/22/2023).

[40] M. Herlihy and N. Shavit. *The Art of Multiprocessor Programming, Revised Reprint*. Elsevier Science, 2012. ISBN: 9780123977953.

[41] Brandon Falk. *Vectorized Emulation: Hardware accelerated taint tracking at 2 trillion instructions per second*. Oct. 2018. URL: https://gamozolabs.github.io/fuzzing/2018/10/14/vectorized_emulation.html (visited on 05/22/2023).

[42] Intel. *Intel® Intrinsics Guide*. May 2023. URL: https://www.intel.com/content/www/us/en/docs/intrinsics-guide/index.html (visited on 05/22/2023).

[43] Bernt Øksendal. *Stochastic Differential Equations: An Introduction with Applications*. 6th ed. Universitext. Springer, 2003. ISBN: 978-3-5400-4758-2.

[44] Yoshio Komori. “Weak order stochastic Runge–Kutta methods for commutative stochastic differential equations”. In: *Journal of Computational and Applied Mathematics* 203.1 (2007), pp. 57–79. ISSN: 0377-0427. DOI: https://doi.org/10.1016/j.cam.2006.03.010. URL: https://www.sciencedirect.com/science/article/pii/S0377042706001737.

[45] A. J. Roberts. *Modify the Improved Euler scheme to integrate stochastic differential equations*. 2012. arXiv: 1210.0933 [math.NA].

[46] Erich Strohmaier et al. *TOP500 - June 2023*. May 2023. URL: https://www.top500.org/lists/top500/2023/06/ (visited on 05/22/2023).

[47] Robert Robey and Yuliana Zamora. *Parallel and high performance computing*. New York, NY: Manning Publications, July 2021. ISBN: 978-1-6172-9646-8.

[48] IEEE. “IEEE Standard for Binary Floating-Point Arithmetic”. In: *ANSI-/IEEE Std 754-1985* (1985), pp. 1–20. DOI: 10.1109/IEEESTD.1985.82928.

[49] William P. Livingston et al. “Experimental demonstration of continuous quantum error correction”. In: *Nature Communications* 13.1 (Apr. 2022), p. 2307. ISSN: 2041-1723. DOI: 10.1038/s41467-022-29906-0. URL: https://doi.org/10.1038/s41467-022-29906-0.

[50] Anton L. Andersen and Klaus Mølmer. “Quantum Nondemolition Measurements of Moving Target States”. In: *Phys. Rev. Lett.* 129 (12 Sept. 2022), p. 120402. DOI: 10.1103/PhysRevLett.129.120402. URL: https://link.aps.org/doi/10.1103/PhysRevLett.129.120402.

[51] Andreas Abel and Jan Reineke. “uiCA: Accurate Throughput Prediction of Basic Blocks on Recent Intel Microarchitectures”. In: *ICS '22: 2022 International Conference on Supercomputing, Virtual Event, USA, June 27-30, 2022*. Ed. by Lawrence Rauchwerger et al. ICS '22. ACM, June 2022, pp. 1–12. URL: https://dl.acm.org/doi/pdf/10.1145/3524059.3532396.

Appendix

A Performance metrics of the simulator

A.1 Performance analysis of 4x4 operator multiply

The disassembly of a matrix multiplication for 4x4 operators in our simulator can be seen in figure 24. Note that every instruction is a vector instruction. Normally 8 matrix multiplications for 4x4 matrices take 896 floating point operations including both multiplications and additions. Analyzing the disassembly using uiCA[51], we learn that this code can be executed in 56 cycles. Which means this code is effectively doing 16 floating point operations per cycle.

```
vmovupd ymm6, [rsp+864]
vmulpd  ymm4, ymm6, ymm3
vmovupd ymm5, [rsp+896]
vfmsub231pd ymm4, ymm2, ymm5
vmulpd  ymm5, ymm5, ymm3
vfmadd231pd ymm5, ymm2, ymm6
vaddpd  ymm4, ymm4, [r12+rbx]
vaddpd  ymm5, ymm5, [r12+rbx+32]
vmovupd ymm9, [rsp+736]
vmulpd  ymm6, ymm9, ymm1
vmovupd ymm8, [rsp+768]
vfmsub231pd ymm6, ymm0, ymm8
vaddpd  ymm7, ymm4, ymm6
vmulpd  ymm4, ymm8, ymm1
vfmadd231pd ymm4, ymm0, ymm9
vaddpd  ymm5, ymm4, ymm5
vmovapd ymm4, [r14+rbx+128]
vmovapd ymm6, [r14+rbx+160]
vmovupd ymm10, [rsp+672]
vmulpd  ymm8, ymm10, ymm6
vmovupd ymm9, [rsp+704]
vfmsub231pd ymm8, ymm4, ymm9
vaddpd  ymm8, ymm8, ymm7
vmulpd  ymm7, ymm9, ymm6
vfmadd231pd ymm7, ymm4, ymm10
vaddpd  ymm9, ymm7, ymm5
vmovapd ymm5, [r14+rbx+192]
vmovapd ymm7, [r14+rbx+224]
vmovupd ymm12, [rsp+608]
vmulpd  ymm10, ymm12, ymm7
vmovupd ymm11, [rsp+640]
vfmsub231pd ymm10, ymm5, ymm11
vaddpd  ymm8, ymm8, ymm10
vmulpd  ymm10, ymm11, ymm7
vfmadd231pd ymm10, ymm5, ymm12
vaddpd  ymm9, ymm10, ymm9
vmovapd [r12+rbx], ymm8
vmovapd [r12+rbx+32], ymm9
vmovupd ymm10, [rsp+800]
vmulpd  ymm8, ymm10, ymm3
vmovupd ymm9, [rsp+832]
vfmsub231pd ymm8, ymm2, ymm9
vmulpd  ymm3, ymm9, ymm3
vfmadd231pd ymm3, ymm10, ymm2
vaddpd  ymm2, ymm8, [r12+rbx+64]
vaddpd  ymm3, ymm3, [r12+rbx+96]
vmovupd ymm10, [rsp+544]
vmulpd  ymm8, ymm10, ymm1
vmovupd ymm9, [rsp+576]
vfmsub231pd ymm8, ymm0, ymm9
vaddpd  ymm2, ymm8, ymm2
vmulpd  ymm1, ymm9, ymm1
vfmadd231pd ymm1, ymm10, ymm0
vaddpd  ymm0, ymm1, ymm3
vmovupd ymm8, [rsp+480]
vmulpd  ymm1, ymm8, ymm6
vmovupd ymm3, [rsp+512]
vfmsub231pd ymm1, ymm4, ymm3
vaddpd  ymm1, ymm2, ymm1
vmulpd  ymm2, ymm3, ymm6
vfmadd231pd ymm2, ymm8, ymm4
vaddpd  ymm0, ymm2, ymm0
vmovupd ymm4, [rsp+416]
vmulpd  ymm2, ymm4, ymm7
vmovupd ymm3, [rsp+448]
vfmsub231pd ymm2, ymm5, ymm3
vaddpd  ymm1, ymm1, ymm2
vmulpd  ymm2, ymm3, ymm7
vfmadd231pd ymm2, ymm4, ymm5
vaddpd  ymm0, ymm2, ymm0
vmovapd [r12+rbx+64], ymm1
vmovapd [r12+rbx+96], ymm0
vmovapd ymm2, [r14+rbx]
vmovapd ymm3, [r14+rbx+32]
vmovapd ymm0, [r14+rbx+64]
vmovapd ymm1, [r14+rbx+96]
vmovupd ymm6, [rsp+352]
vmulpd  ymm4, ymm6, ymm3
vmovupd ymm5, [rsp+384]
vfmsub231pd ymm4, ymm2, ymm5
vmulpd  ymm5, ymm5, ymm3
vfmadd231pd ymm5, ymm2, ymm6
vaddpd  ymm4, ymm4, [r12+rbx+128]
vaddpd  ymm5, ymm5, [r12+rbx+160]
vmovupd ymm9, [rsp+288]
vmulpd  ymm6, ymm9, ymm1
vmovupd ymm8, [rsp+320]
vfmsub231pd ymm6, ymm0, ymm8
vaddpd  ymm7, ymm4, ymm6
vmulpd  ymm4, ymm8, ymm1
vfmadd231pd ymm4, ymm0, ymm9
vaddpd  ymm5, ymm4, ymm5
vmovapd ymm4, [r14+rbx+128]
vmovapd ymm6, [r14+rbx+160]
vmovupd ymm10, [rsp+224]
vmulpd  ymm8, ymm10, ymm6
vmovupd ymm9, [rsp+256]
vfmsub231pd ymm8, ymm4, ymm9
vaddpd  ymm8, ymm8, ymm7
vmulpd  ymm7, ymm9, ymm6
vfmadd231pd ymm7, ymm4, ymm10
vaddpd  ymm9, ymm7, ymm5
vmovapd ymm5, [r14+rbx+192]
vmovapd ymm7, [r14+rbx+224]
vmovupd ymm12, [rsp+160]
vmulpd  ymm10, ymm12, ymm7
vmovupd ymm11, [rsp+192]
vfmsub231pd ymm10, ymm5, ymm11
vaddpd  ymm8, ymm8, ymm10
vmulpd  ymm10, ymm11, ymm7
vfmadd231pd ymm10, ymm5, ymm12
vaddpd  ymm9, ymm10, ymm9
vmovapd [r12+rbx+128], ymm8
vmovapd [r12+rbx+160], ymm9
vmovupd ymm10, [rsp+96]
vmulpd  ymm8, ymm10, ymm3
vmovupd ymm9, [rsp+128]
vfmsub231pd ymm8, ymm2, ymm9
vmulpd  ymm3, ymm9, ymm3
vfmadd231pd ymm3, ymm10, ymm2
vaddpd  ymm2, ymm8, [r12+rbx+192]
vmovupd ymm10, [rsp+32]
vmulpd  ymm8, ymm10, ymm1
vmovupd ymm9, [rsp+64]
vfmsub231pd ymm8, ymm0, ymm9
vaddpd  ymm2, ymm8, ymm2
vmulpd  ymm1, ymm9, ymm1
vfmadd231pd ymm1, ymm10, ymm0
vaddpd  ymm0, ymm3, [r12+rbx+224]
vaddpd  ymm0, ymm1, ymm0
vmulpd  ymm1, ymm13, ymm6
vmovupd ymm3, [rsp]
vfmsub231pd ymm1, ymm4, ymm3
vaddpd  ymm1, ymm2, ymm1
vmulpd  ymm2, ymm3, ymm6
vfmadd231pd ymm2, ymm13, ymm4
vaddpd  ymm0, ymm2, ymm0
vmulpd  ymm2, ymm15, ymm7
vfmsub231pd ymm2, ymm5, ymm14
vaddpd  ymm1, ymm1, ymm2
vmulpd  ymm2, ymm14, ymm7
vfmadd231pd ymm2, ymm15, ymm5
vaddpd  ymm0, ymm2, ymm0
vmovapd [r12+rbx+192], ymm1
vmovapd [r12+rbx+224], ymm0
```

Figure 24: Disassembly of the matrix multiplication for 4x4 operators.

A.2 Performance analysis with *perf*

Here we use the Linux *perf* tool to get access to performance counters. Simulations were done with 5000 steps, 1000 simulations per thread and 10 threads, unless indicated otherwise. The *perf* command was,

```
perf stat -e \
    branches,branch-misses,cycles,instructions,LLC-loads,LLC-load-misses,fp_arith_inst_retired.256b_packed_single\
    ./target/release/qubitsim
```

This command counts overall branches, branch misses, cycles, instructions, last level cache loads/misses, and single precision floating point vector operations. It counts both the multiplication and addition in fused multiply adds as separate operations.

A.2.1 Single qubit

```
Performance:  111868993767 × 8FLOP / 4.397132270s = 203.5 GFLOP/s

 Performance counter stats for './target/release/qubitsim':

     7.962.269.467      branches:u                                                    (71,44%)
        49.860.839      branch-misses:u                 #    0,63% of all branches    (57,15%)
   135.674.857.659      cycles:u                                                      (57,15%)
   214.607.258.766      instructions:u                  #    1,58  insn per cycle     (71,43%)
         5.385.218      LLC-loads:u                                                   (71,42%)
           182.576      LLC-load-misses:u               #    3,39% of all LL-cache accesses  (71,43%)
   111.868.993.767      fp_arith_inst_retired.256b_packed_single:u                    (71,44%)

       4,397132270 seconds time elapsed

      30,860163000 seconds user
       1,476724000 seconds sys
```

A.2.2 Two qubits

```
Performance:  1029197324501 × 8 FLOP / 35.470138850s = 232.1 GFLOP/s

 Performance counter stats for './target/release/qubitsim':

    67.767.103.208      branches:u                                                    (71,43%)
        50.711.879      branch-misses:u                 #    0,07% of all branches    (57,14%)
 1.202.705.654.435      cycles:u                                                      (57,14%)
 1.688.824.450.844      instructions:u                  #    1,40  insn per cycle     (71,43%)
       655.345.926      LLC-loads:u                                                   (71,43%)
         2.037.097      LLC-load-misses:u               #    0,31% of all LL-cache accesses  (71,43%)
 1.029.197.324.501      fp_arith_inst_retired.256b_packed_single:u                    (71,43%)

      35,470138850 seconds time elapsed

     273,480074000 seconds user
       1,469019000 seconds sys
```

A.2.3 Three qubits

The step count was reduced to 500 per simulation.

```
Performance:  963926014784 × 8 FLOP / 30.906494601s = 249.5 GFLOP/s

 Performance counter stats for './target/release/qubitsim':

    52.548.454.254      branches:u                                                    (71,43%)
       274.843.662      branch-misses:u                 #    0,52% of all branches    (57,14%)
 1.066.665.206.701      cycles:u                                                      (57,14%)
 1.674.511.074.114      instructions:u                  #    1,57  insn per cycle     (71,43%)
     2.428.971.578      LLC-loads:u                                                   (71,43%)
         1.571.487      LLC-load-misses:u               #    0,06% of all LL-cache accesses  (71,43%)
   963.926.014.784      fp_arith_inst_retired.256b_packed_single:u                    (71,43%)

      30,906494601 seconds time elapsed

     242,588937000 seconds user
       0,196753000 seconds sys
```

A.2.4 Four qubits

Step count was reduced to 50 per simulation.

```
Performance: 893369806941 × 8 FLOP / 31.831373848s = 224.5 GFLOP/s

     9.034.729.100      branches:u                                                    (71,43%)
        61.799.305      branch-misses:u                 #    0,68% of all branches    (57,14%)
 1.092.345.298.672      cycles:u                                                      (57,14%)
 1.170.691.913.142      instructions:u                  #    1,07  insn per cycle     (71,43%)
     4.100.359.905      LLC-loads:u                                                   (71,42%)
        34.912.509      LLC-load-misses:u               #    0,85% of all LL-cache accesses  (71,43%)
   893.369.806.941      fp_arith_inst_retired.256b_packed_single:u                    (71,43%)

      31,831373848 seconds time elapsed

     248,381686000 seconds user
       0,090720000 seconds sys
```