UNIVERSITÀ DEGLI STUDI DI MILANO

FACOLTÀ DI SCIENZE E TECNOLOGIE

Master degree in Physics

Development of an open-source calibration framework for superconducting qubits

Supervisor:
Prof. Dr. Stefano Carrazza

Co-supervisor:
Dr. Alessandro Candido

Co-supervisor:
Dr. Andrea Pasquale

Co-supervisor:
Dott. Edoardo Pedicillo

Elisa Stabilini
Matricola n° 28326A
A.A. 2024/2025

Contents

Summary

Chapter 1

Superconducting qubits

The electronics that modern computers rely on contain components that operate based on quantum mechanics; however, their computational processes are still governed by classical laws. For this reason, they are referred to as "classical computers."

Quantum computing emerged from Richard Feynman's idea that simulating quantum systems efficiently requires quantum mechanical resources [1]. Classical computers struggle to model complex quantum interactions due to the exponential growth of computational requirements with system size, making exact simulations infeasible beyond small systems [2]. Quantum computers, taking advantage of quantum mechanics phenomena like superposition and entanglement, offer a natural framework for such simulations and have been demonstrated to provide exponential speedups for certain quantum systems [3].

Beyond quantum simulation, curren theoretical advancements suggest that quantum algorithms can outperform classical counterparts in solving specific problems [4].

1.1   Introduction

The physical realization of quantum computing necessitates the development of a system capable of functioning as quantum bits (qubits).
Similar to classical logic, where the bits 0 and 1 are associated with two physical levels, typically represented by high and low voltage states, a qubit can, to a first approximation, be considered as a two-level physical system.

Mathematically, this system is described within a two-dimensional complex Hilbert space, where the basis states $|0\rangle$ and $|1\rangle$ correspond to two orthonormal vectors. Any general state of the qubit can be expressed as a superporition of these basis states:

$$|\psi\rangle = \alpha |0\rangle + \beta |1\rangle \rightarrow \begin{pmatrix} \alpha \\ \beta \end{pmatrix}, \tag{1.1}$$

where $\alpha, \beta \in \mathbb{C}$. If the normalization condition $|\alpha|^2 + |\beta|^2 = 1$ holds, the state $|\psi\rangle$ represents a qubit. The basis $\{|0\rangle, |1\rangle\}$ is called computational basis and the information is stored in the complex numbers α and β.

A possible geometric representation of qubit states is given by the Bloch sphere, which offers a visualization of two level quantum systems as vectors on a unit sphere. A qubit state is depicted as a vector originating from the center of the sphere, with the computational basis states $|0\rangle$ and $|1\rangle$ positioned at the north and south poles, respectively. The axis connecting these states defines the z-axis. The transverse x- and y- axes correspond to the equal superposition states $|\pm\rangle = \frac{1}{\sqrt{2}}(|0\rangle \pm |1\rangle)$ and $|\pm i\rangle = \frac{1}{\sqrt{2}}(|0\rangle \pm i |1\rangle)$, respectively.
A vector of unit length on the Blooch sphere is characterized by the polar angle θ, with $0 \leq \theta \leq \pi$ and the azimuthal angle φ, with $0 \leq \varphi \leq 2\pi$, each unit vector represent a possible pure state of the qubit.

The qubit states $|0\rangle$ and $|1\rangle$ can also be associated with energy eigenstates of a physical system, where $|0\rangle$ represents the ground state with energy E_0 and $|1\rangle$ represents the excited state with energy
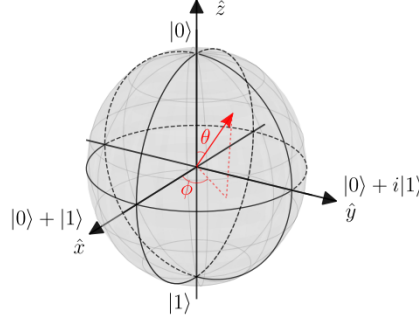
Figure 1.1: Example of qubit state representation on the Bloch sphere
Source: Metrology of Quantum Control and Measurement in Superconducting Qubits [5]

E_1, assuming $E_0 < E_1$. In this energy eigenbasis, the Hamiltonian of the qubit is given by

$$\hat{H}_q = E_0 \left|0\right\rangle \left\langle 0\right| + E_1 \left|1\right\rangle \left\langle 1\right| = \begin{pmatrix} E_0 & 0 \\ 0 & E_1 \end{pmatrix}. \tag{1.2}$$

Since only energy differences are physically relevant, it is possible to redefine the zero-point energy by subtracting the constant term $E_0(\left|0\right\rangle \left\langle 0\right| + \left|1\right\rangle \left\langle 1\right|)$, leading to the simplified Hamiltonian

$$\hat{H}_q = (E_1 - E_0) \left|1\right\rangle \left\langle 1\right| = \hbar\omega_q \left|1\right\rangle \left\langle 1\right| = \hbar\omega_q \hat{\sigma}^+ \hat{\sigma}^- = \begin{pmatrix} 0 & 0 \\ 0 & \hbar\omega_q \end{pmatrix}, \tag{1.3}$$

where $\omega_q = (E_1 - E_0)/\hbar$ is the qubit transition frequency, and we have used the relation $\hat{\sigma}^+ \hat{\sigma}^- = \left|1\right\rangle \left\langle 1\right|$. For convenience, the Hamiltonian can also be rewritten in terms of the Pauli z-matrix, $\hat{\sigma}_z$, by adding a term proportional to the identity:

$$\hat{H}_q = \hbar\omega_q \left|1\right\rangle \left\langle 1\right| - \frac{\hbar\omega_q}{2}\mathbb{I} = \begin{pmatrix} -\frac{\hbar\omega_q}{2} & 0 \\ 0 & \frac{\hbar\omega_q}{2} \end{pmatrix} = -\frac{\hbar\omega_q}{2}\hat{\sigma}_z. \tag{1.4}$$

Qubits can be implemented through various physical mechanisms; however, their practical realization remains a significant challenge due to their susceptibility to environmental interactions, which lead to decoherence and reduce their coherence time. Despite the diversity of possible physical implementations, any functional quantum computing system must satisfy a set of fundamental criteria. These requirements, known as the DiVincenzo criteria, establish the essential conditions for the construction and operation of a viable quantum computer [6], [7]:

1. The physical system used as quantum computer must comprise a set of qubits, meaning that the quantum system must be well-characterized, and scalable such that quantum computing can be realized.

2. It must be possible to initialize the qubits in a reliable state, such as the ground state.

3. The coherence time of the qubits must be longer than the typical gate time, ideally should be possible to perform $> 10^4$ operations, that is the number which allows for realizing effective error corrections.

4. It must be possible to implement a universal set of quantum gates.

5. It must be possible to measure the qubits in the computational basis.

In the present work, I will focus on superconducting qubits, which constitute the hardware I have worked on and where the experiments were conducted. However, several of the experiments described later can also be implemented using different physical systems.

1.2   Transmon qubits

In this section, I provide a review of the structure and operation of superconducting transmon qubits. The content of this section is based on the *Quantum Information Science* manual [7], *The Metrology of Quantum Control and Measurement in Superconducting Qubits* [5], the notes from quantum computing lectures held by Professor Olivares [8], and the original transmon paper [9].

1.2.1   Josephson Junctions

The Josephson junction (JJ) is formed by a thin oxide layer positioned between the two superconductors which acts as an insulating barriers. An example of Josephson junction is show in figure 1.2, a side view in image 1.2a and a top view in image 1.2b.

(a)

(b)

Figure 1.2: Figure 1.2a: Side viwe of a Josephson junction, the two superconducting pads are coloured in red and blue and indicating by the letter S. In grey, indicated by letter I is represented the insulating barrier of oxide. The superconductors and the oxide are layered over a substrate.
Figure 1.2b: Electron microscope image of a $2\mu m \times 2\mu m$ cross-type junction: I. Josephson junction. II. Base electrode. III. Contact to the top electrode.
Source:https://www.ims.kit.edu/english/2551.php

Superconductivity is a phenomenon observed in certain materials where, when cooled well below a critical temperature T_c, which depends on the material, their electrical resistance drops to zero, allowing them to behave as perfect conductors. According to the BCS (Bardeen-Cooper-Schrieffer) theory, superconductivity arises, from the formation of Cooper pairs, which are bound states of electrons with opposite momenta and spins. These pairs collectively forms a macroscopic quantum states described by a single waveform $\psi(\mathbf{r})$ which can be expressed as

$$\psi(\mathbf{r}) = \sqrt{\rho(\mathbf{r})}e^{i\theta(\mathbf{r})} \tag{1.5}$$

where $\rho(\mathbf{r})$ is the density of Cooper pairs in the metal, which is typically uniform in the bulk of the superconductor, and $\theta(\mathbf{r})$ is the macroscopic phase of the superconducting wavfunction.

For this reason the wavefunctions on the two sides of the JJ can be denoted as

$$\psi_1(\mathbf{r},t) = \sqrt{\rho_1(\mathbf{r},t)}e^{i\theta_1(\mathbf{r},t)}, \psi_2(\mathbf{r},t) = \sqrt{\rho_2(\mathbf{r},t)}e^{i\theta_2(\mathbf{r},t)} \tag{1.6}$$

The dynamics of the system can be described by the two equations

$$i\hbar\frac{d\psi_1}{dt} = E_1\psi_1 + K\psi_2, \tag{1.7}$$

$$i\hbar\frac{d\psi_2}{dt} = E_2\psi_2 + K\psi_1. \tag{1.8}$$

By substituting the expression of ψ_i into the Schrödinger equation 1.7, 1.8 we obtain

$$\frac{d\rho_1}{dt} = \frac{2K}{\hbar}\sqrt{\rho_1\rho_2}\sin(\theta_2 - \theta_1), \tag{1.9}$$

$$\frac{d\rho_2}{dt} = -\frac{2K}{\hbar}\sqrt{\rho_1\rho_2}\sin(\theta_2 - \theta_1). \tag{1.10}$$

Since the derivative of the charge density is the current, from equations 1.9 and 1.10 we obtain the first Josephson equation

$$I = I_c\sin\phi \tag{1.11}$$

where $I_c = \frac{2K}{\hbar}\sqrt{\rho_1\rho_2}$ is the critical current and ϕ is the superconducting phase difference $\theta_2 - \theta_1$.

Instead, from the real part of the Schrödinger equation 1.7, 1.8 and a few calculations, we obtain the second Josephson equation

$$\frac{d\phi}{dt} = \frac{2e}{\hbar}V(t). \tag{1.12}$$

which can be rewritten as

$$\frac{d\phi}{dt} = \frac{2\pi}{\Phi_0} V(t). \tag{1.13}$$

where $\Phi_0 = \frac{h}{2e}$ is the superconducting flux quantum, with h is the Planck's constant and $2e$ is the charge of a Cooper pair.

The time derivative of the first Josephson equation 1.11 yields:

$$\dot{I}_J = I_C \cos\phi \frac{\partial\phi}{\partial t}, \tag{1.14}$$

equation 1.14 suggests a nonlinear relation between the current the voltage. Using the Josephson voltage-phase relation and the fact that $\dot{I} = \frac{V}{L}$ it is possible to define an effective nonlinear inductance for the Josephson junction:

$$L_J = \frac{1}{\cos\phi}\frac{\Phi_0}{2\pi I_c}. \tag{1.15}$$

In addition to ths inductive behaviour the Josephson junction also exibits capacitive properties due to its inherent capacitance C_J with a corresponing energy of

$$E_{C_J} = \frac{Q^2}{2C_J} \tag{1.16}$$

.

Fom equation 1.15 it is possible to compute the energy stored in the nonlinear inductance as

$$E_{L_J} = \int_0^t \mathrm{d}\tau I_J(\tau)V(\tau) = \int_0^t \mathrm{d}\tau I_c \sin\phi(\tau)\frac{\partial\phi(\tau)}{\partial\tau}\frac{\Phi_0}{2\pi} \tag{1.17}$$

$$= \frac{\Phi_0 I_c}{2\pi}(1 - \cos\phi) = E_J(1 - \cos\phi) \tag{1.18}$$

where E_J represents the energy due to the behaviour of the junction as nonlinear inductor .

1.2.2   CPB qubit

A first example of superconducting qubit is the Cooper Pair Box (CPB), which consists of a small superconducting island connected to a reservoir of superconducting electrons through a Josephson junction [10], with an external gate voltage controlling the charge state. The circuit corresponding to CPB is similar to the circuit of a parallel resonator where the linear inductance is subsituted by a Josephson junction which simply acts as a nonlinear inductance.

Combining the energy associated to the capacitance C and the energy of the Josephson junction 1.17 it is possible to write the classical Hamiltonian of the circuit

$$H_J = 4E_C n^2 - E_J \cos\phi \tag{1.19}$$

where the constant term was ignored as it acts simply as a constant offset without influencing the dynamics of the system and where E_C is the charging energy defined as

$$E_C = \frac{e^2}{2C}. \tag{1.20}$$

To control the number of Cooper pairs on the island, it is possible to connecnt a DC voltage source V_g to the system through a gate capacitor C_g, as shown in Figure 1.3b. When $V_g = 0$, both the gate and qubit capacitors remain uncharged. As V_g increases, a charge $Q_g = C_g V_g$ accumulates on the gate capacitor, inducing an equal and opposite charge on the island to maintain charge neutrality. When $Q_g \approx 2e$, a Cooper pair tunnels from the reservoir to the island, discharging the qubit capacitor.

The presence of the external voltage source introduces an additional control parameter for the number of Cooper pairs on the island, modifying the system's Hamiltonian. The resulting Hamiltonian of the Cooper Pair Box (CPB) takes the form

$$\hat{H} = 4E_C(\hat{n} - n_g)^2 - E_J \cos\hat{\phi} \tag{1.21}$$

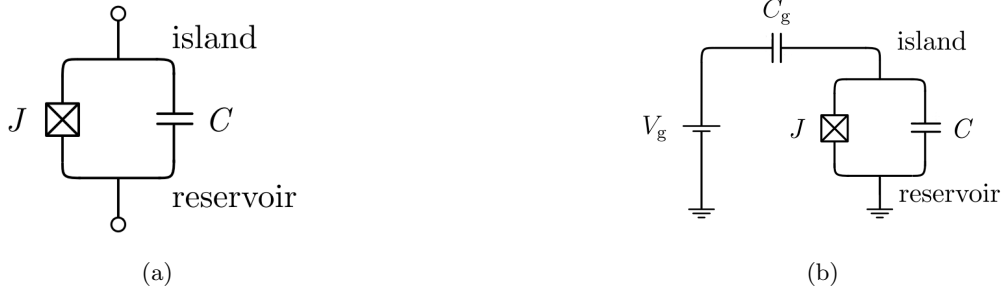where $n_g = \frac{C_g V_g}{2e}$ represents the normalized gate charge.

4

Figure 1.3: Figure 1.3a: corresponding circuit of a CPB which consists of Josephson junction shunted by a capacitor. Surce: [7]. Figure 1.3b: electrical circuit of a CPB capacitively coupled to a DC voltage source through a capacitor C_g. Source: [7].

A key feature of the system is the presence of a Josephson junction in place of a linear inductance. Unlike a standard LC circuit—corresponding to a quantum harmonic oscillator—this results in a non-equidistant energy level spectrum. In particular, the energy levels are anharmonic, allowing the first two levels to be spectrally isolated from the higher ones. This anharmonicity enables the use of the subspace spanned by the ground state $|0\rangle$ and the first excited state $|1\rangle$ aas a qubit.

The qubit frequency is defined as the frequency associated with the energy difference between these two states: $f_{01}(n_g) = f_Q = \frac{(E_1 - E_0)}{h}$. This frequency can be tuned by varying the externally applied DC voltage, which modifies the system's parameters and, consequently, its energy level spacing.

1.2.3   Transmon qubit

One of the main drawbacks of the Cooper Pair Box (CPB) qubit, which ultimately led to its replacement by other qubit architectures, is its limited coherence time. The transmon qubit was introduced specifically to address this issue, with the goal of improving the dephasing time of the CPB. The key idea behind the transmon is to reduce the sensitivity of the energy levels to fluctuations in the gate charge—effectively flattening the energy bands—by increasing the ratio between the Josephson energy $E_J/E_C \gtrsim 1$, this architecture was first proposed in [9], the first and more straightforward method to increase this ratio is to enlarge the capacitance of the qubit, which reduces the charging energy E_C.

Since the CPB and the transmon qubit have the same electrical circuit they are also described by the same hamiltonian 1.19. The difference is that in this case the transmon satisfies the condition $E_J/E_C \gtrsim 1$ it is possible to expand the cosine term in 1.19 with a Taylor series and neglect the higher order terms:

$$\hat{H} \approx 4E_C\hat{n}^2 + \frac{1}{2}E_J\hat{\phi}^2 - \frac{E_J}{4!}\hat{\phi}^4 \tag{1.22}$$

where the last term, proportional to $\hat{phi}^4$, makes the potential of the transmon slightly anharmonic.

As happens in the standard harmonic oscillator case, the operators $\hat{phi}$ and $\hat{n}$ satisfy the cnaonical commutation relation $[\hat{\phi}, \hat{n}] = i\mathbb{I}$, it is possible to introduce the raising and lowering operators $\hat{b}, \hat{b}^\dagger$ as

$$\hat{\phi} = \sqrt{\xi}(\hat{b} + \hat{b}^\dagger), 2 \quad \hat{n} = -\frac{i}{2\sqrt{xi}}(\hat{b} - \hat{b}^\dagger), \tag{1.23}$$

where $\xi = \sqrt{2E_C/E_J}$.

Substituting equations 1.23 in the hamiltonian, equation 1.22 becomes

$$\hat{H} = \sqrt{8E_JE_C}\hat{b}^\dagger\hat{b} - \frac{E_C}{12}(\hat{b} + \hat{b}^\dagger)^4. \tag{1.24}$$

Given equation 1.24 it is possible to solve the eigenvalue proble $\hat{H}|k\rangle = E_k|k\rangle$ and calculate the energy levels E_k. The first term of hamiltonian 1.24 is the harmonic oscillator contribution with eigenstates $|k\rangle$ and eigenvalues $\sqrt{8E_JE_C}k$. Since $E_C \ll E_J$, the second term $\hat{V} = -\frac{E_C}{12}(\hat{b} + \hat{b}^\dagger)^4$ represents a small perturbative contribution to the Hamiltonian and can be treated using perturbation theory. The first-order correction to the energy levels is given by the diagonal matrix elements of the perturbation operator:

$\Delta E_k^{(1)} = \langle k|\hat{V}|k\rangle$. It is possible to verify that $\langle k|\hat{V}|k\rangle = -\frac{E_C}{12}(6k^2 + 6k + 3)$. Thus the eigenenergies of the transmon hamiltonian are

$$E_k \approx \sqrt{8E_J E_C}k - \frac{E_C}{2}(k^2 + k). \tag{1.25}$$

As mentioned before, the qubit frequency is defined as $f_Q = f_{01} = (E_1 - E_0)/h$ which yields

$$f_01 \approx (\sqrt{8E_J E_C} - E_C)/h \tag{1.26}$$

As explained at the beginning of this section, a large E_J/E_C ratio makes the transmon qubit significantly less sensitive to charge noise. However, this improvement comes at the expense of reduced anharmonicity in the energy level spectrum. The anharmonicity η is defined as the difference between the second and first transition energies, relative to the first transition energy:

$$\eta = \frac{(E_2 - E_1) - (E_1 - E_0)}{\hbar} = \omega_{12} - \omega_{01}. \tag{1.27}$$

For a transmon, the anharmonicity η is negative, reflecting the fact that the level spacing decreases with increasing energy. Ideally, the absolute value $|\eta|$ should be sufficiently large to allow external microwave drives to selectively address the $|0\rangle \leftrightarrow |1\rangle$ transition without inadvertently exciting higher-energy states.

Flux-tunable transmon   To implement certain two-qubit gate schemes, such as swap interactions, it is essential to tune the qubit frequency. A common approach to achieving this is by adding an extra junction to the transmon, the most common configuration is the SQUID (Superconducting QUantum Interference Device). In the SQUID configuration two Josephson junctions are connected in parallel on a superconducting loop as shown in Figure [??].

Starting from the Hamiltonian of the single Josephson junction it is possible to write the hamiltonian of a SQUID:

$$\hat{H} = 4E_C\hat{n}^2 - E_{J1}\cos\hat{\phi}_1 - E_{J2}\cos\hat{\phi}_2 \tag{1.28}$$

where E_{J1} and E_{J2} are the Josephson energies of the two junctions, and the operators $\hat{\phi}_1$ and $\hat{\phi}_2$ are the phase differences across the junctions.

Because od the quantization of the magnetic flux through the SQUID, the quantities $\hat{\phi}_1$ and $\hat{\phi}_2$ are not independet. In particular, as show in [7], the difference between $\hat{\phi}_1$ and $\hat{\phi}_2$ follows the following relation:

$$\hat{\phi}_1 - \hat{\phi}_2 = \frac{2\pi}{\Phi_0}\Phi_{\text{ext}}\mathbb{I}(\text{mod}2\pi) \tag{1.29}$$

where Φ_{ext} is the flux of external magnetic field defined as the integral of the magnetic field over the SQUID area.

Equation 1.29 can be simplified and rewritten as

$$\hat{H} = 4E_C\hat{n}^2 - E_J(\Phi_{\text{ext}})\cos\hat{\varphi} \tag{1.30}$$

where $\hat{\varphi} = \frac{\hat{\phi}_1 + \hat{\phi}_2}{2}$, and the Josephson energy is flux-dependent:

$$E_J(\Phi_{\text{ext}}) = (E_{J1} + E_{J2})\left|\cos\left(\pi\frac{\Phi_{\text{ext}}}{\Phi_0}\right)\right|\sqrt{1 + d^2\tan^2\left(\pi\frac{\Phi_{\text{ext}}}{\Phi_0}\right)}, \tag{1.31}$$

where $d = \frac{E_{J1} - E_{J2}}{E_{J1} + E_{J2}}$ is the relative junction asymmetry.

Then, it's easy to see that the frequency f_Q of a two-junction transmon dependends on the magnetic flux $\Phi_Q(t)$ through the SQUID loop, for symmetric junctions is given by

$$f_Q(\Phi_Q) \approx \frac{1}{h}\left(\sqrt{8E_J E_C \cos\left(\pi\frac{\Phi_Q}{\Phi_0}\right)} - E_C\right). \tag{1.32}$$

1.3   Quantum operations

A quantum operation is a mathematical transformation that describes how a quantum state changes as a consequence of a physical process. Formally, it is a map $\mathcal{E}$ that transforms a quantum state described by a density operator $\hat{\rho}$ into another state described by a new density operator $\hat{\rho}'$:

$$\mathcal{E}(\rho) = \rho'. \tag{1.33}$$

The simplest example of a quantum operation is the evolution of a quantum state $\hat{\rho}$ of a closed quantum system, under a unitary operator $\hat{U}$, which can be written as $\mathcal{E} \equiv \hat{U}\hat{\rho}\hat{U}^\dagger$.

Depolarizing chennel   A depolarizing channel describes a process in which the current state of the n-qubit system ρ, is replaced by $\frac{\mathbb{I}}{2^n}$, with probability d. This process can be represented with a quantum map as follows:

$$\mathcal{E}_{dc}(\rho) = d\frac{\mathbb{I}}{2^n} + (1-d)\rho \tag{1.34}$$

1.3.1   Rabi experiments

Chapter 2

Hardware and software description

2.1 Software

2.1.1 Qibo

2.1.2 Qibocal

Di seguito descrivo con maggiore dettaglio gli esperimenti per la calibrazione dei qubit che ho usato principalemente durante il lavoro di tesi.

Rabi experiment

Ramsey experiment

Chevron

2.2 Hardware

Chapter 3

RB fidelity optimization

Tutti i risultati che sono presentati nel seguito sono stati ottenuti utilizzando il software di `Qibolab`per l'interazione con gli strumenti del laboratorio e `Qibocal`per il controllo delle operazioni sui qubit. L'hardware è un chip di QunatumWare. Durante il lavoro condotto per questo progetto di tesi entrambe le libererie, sia Qibocal che Qibolab undergo update and release, for this reason the first part of this work was realized using `Qibocal`v0.1 and `Qibolab`v0.1 while the second part of the work, dato che puntava anche allo sviluppo di routine ch epotessero essere utili per la calibrazione dei qubit è stato realizzato direttamente con `Qibocal`v0.2 e `Qibolab`v0.2.

3.1 Randomized Benchmarking

A strong limitation to the realization of quantum computing technologies is the loss of coherence that happens as a consequence of the application of many sequential quantum gates to to the quibts. A possible approach to characterize gate error is the quantum process tomography which allows the experimenter to establish the behaviour of a quantum gates; the main drawback of this approach is that process tomography can be very time consumig since its time complexity scales exponentially with the number of qubits involved [11] and the result is affected by state preparation and measurements (SPAM) errors.

To overcome these limitations, randomized benchmarking (RB) was introduced and is currently widely used to quantify the avarage error rate for a set of quantum gates.

The main idea is that the error obtained from the combined action of random unitary gates drawn from a uniform distribution with respect to the Haar measure [12] and applied in sequence to the qubit will avarage out to behave like a depolarizing channel [13]. This last consideration simplifies the characterization of noise because it removes dependence on specific error structures and allows fidelity to be extracted through a simple exponential decay.

It was later shown that it is possible to simplify this procedure even more, by restricting the unitaries to gates in the Clifford group [1] and by not requiring that the sequence is strictly self-inverting [14].

The fundamental principle of RB is the application of sequences of randomly selected quantum gates from the Clifford group $\mathcal{C}$ followed by an inversion gate which, in absence of noise, return the system to its initial state. For real systems, where noise is present, the observed survival probability provides an estimate of the avarage gate fidelity.

The standard RB protocols consist of the following steps:

1. Initialize the system in ground state $|0\rangle$

2. For each sequence-length m build a sequence of m randomly drawn Clifford gates $C_1, C_2, ..., C_m$

3. Determine the inverse gate $C_{m+1} = (C_m \circ ... \circ C_1)^{-1}$

4. Measure $C_{m+1} \circ C_m \circ ... \circ C_1 |0\rangle$

The process must be repeated for multiple sequence of the same length and with varying length.

In ideal systems without noise we should have

$$C_{m+1} \circ C_m \circ ... \circ C_1 |0\rangle = (C_m \circ ... \circ C_1)^{-1} \circ (C_m \circ ... \circ C_1) |0\rangle = |0\rangle \qquad (3.1)$$

---

[1]unitary rotations mapping the group of Puali operators in itself

In real systems, where noise is present, eq. 3.1 does not hold; instead randomization with Clifford gates behave as a depolarizing channel 1.34 with depolarization probability d. The survival probability of the initial state $|0\rangle$ for different sequence lengths follows the exponential decay model

$$F(m) = Ap^m + B, \tag{3.2}$$

where $1 - p$ is the rate of depolarization and A and B capture the state preparation and measurement error but not the rate of decay p. Note that the exponential form arises naturally due to the assumption that each gate introduces independent noise.

The parameter p is directly related to the depolarization probability d through the avarage gate fidelity F which, for a depolarizing channel, is given by

$$F = 1 - \frac{d}{2^n - 1}. \tag{3.3}$$

For the details of the calculations to obtain eq. 3.3 see Appendix A.

Now we can derive the avarage error per Clifford gate $\epsilon_{Clifford}$

$$\epsilon_{Clifford} = 1 - F, \tag{3.4}$$

where F is the avarage gate fidelity. Sobstituting in 3.4 the formula for the avarage gate fidelity 3.3 we obtain

$$\epsilon_{Clifford} = \frac{d}{2^n - 1} = \frac{1 - p}{1 - 2^{-n}}, \tag{3.5}$$

which shows how the avarage error per Clifford gate is directly connected to the exponential decay rate.

QUA Randomized Benchmarking

For the results we present in the following the technique used slightly differs from the one described in section 3.1

3.2   Scipy optimization methods

3.2.1   Algorithm description

I primi metodi che abbiamo provato per l'ottimizzazione dei parametri sono quelli standard implementati nella libreria `Scipy` [15].

The first gradient-free optimization method to be tested was Nelder-Mead since in letteratura era già stato riportato il suo utilizzo per obiettivi simili [16].

The Nelder-Mead optimization method, originally introduced by Nelder and Mead in 1965 [17], is a widely used numerical optimization technique for unconstrained problems in multidimensional spaces. This derivative-free method is operates using simplex, which is a polytope of $n + 1$ vertices in a n-dimensional space. The algorithm iteratively updates the simplex by replacing its worst-performing vertex with a new candidate point, thereby guiding the search towards an optimal solution. If the goal is to minimize a given function $f(\mathbf{x})$ where $\mathbf{x} \in \mathbb{R}^n$ the algorithms proceeds with the following steps:

1. If not otherwise initialized, $n + 1$ points are sampled for building the initial symplex

2. **Order** the test points according to their values at vertices: $f(\mathbf{x}_1) \leq f(\mathbf{x}_2) \leq \cdots \leq f(\mathbf{x}_{n+1})$ and check whether the algorithm should terminate.

3. **Calculate** $\mathbf{x}_0$, the centroid of all points except $\mathbf{x}_{n+1}$.

4. **Reflection:** Compute the reflected point $\mathbf{x}_r = \mathbf{x}_0 + \alpha(\mathbf{x}_0 - \mathbf{x}_{n+1})$ with $\alpha > 0$. If $\mathbf{x}_r$ satisfies $f(\mathbf{x}_1) \leq f(\mathbf{x}_r) < f(\mathbf{x}_n)$, then a new simplex is obtained by replacing the worst-performing point $\mathbf{x}_{n+1}$ with $\mathbf{x}_r$ and then go to step 1.

5. **Expansion:** If $\mathbf{x_r}$ is the current best point, meaning that $f(\mathbf{x}_r) < f(\mathbf{x}_1)$, then the expanded point is computed: $\mathbf{x}_e = \mathbf{x}_0 + \gamma(\mathbf{x}_r - \mathbf{x}_0)$ with $\gamma > 1$. If $\mathbf{x}_e$ satisfies $f(\mathbf{x}_e) < f(\mathbf{x}_r)$, then a new simplex is obtained by replacing $\mathbf{x}_{n+1}$ with the expanded point $\mathbf{x}_e$ and then go to step 1.
   If instead $f(\mathbf{x}_e) \geq f(\mathbf{x}_r)$, the new simplex is obtained by replacing $\mathbf{x}_{n+1}$ with $\mathbf{x}_r$, and then go to step 1.

6. `Contraction`: In this case is certain that $f(\mathbf{x}_r) \geq f(\mathbf{x}_n)$ then:

   - If $f(\mathbf{x}_r) < f(\mathbf{x}_{n+1})$: compute the contracted point $\mathbf{x}_c = \mathbf{x}_0 + \rho(\mathbf{x}_r - \mathbf{x}_0)$ with $0 < \rho \leq 0.5$. If $\mathbf{x}_c$ satisfies $f(\mathbf{x}_c) < f(\mathbf{x}_r)$, then a new simplex is obtained by replacing $\mathbf{x}_{n+1}$ with $\mathbf{x}_c$ and go to step 1.
   Else go to step 6.

   - If $f(\mathbf{x}_r) \geq f(\mathbf{x}_{n+1})$: compute the contracted point $\mathbf{x}_c = \mathbf{x}_0 + \rho(\mathbf{x}_{n+1} - \mathbf{x}_0)$ with $0 < \rho \leq 0.5$. If $\mathbf{x}_c$ satisfies $f(\mathbf{x}_c) < f(\mathbf{x}_{n+1})$, the a new simplex is constructed with $\mathbf{x}_c$ and go to step 1.
   Else go to step 6.

7. `Shrinkage`: Replace all points except the best, $\mathbf{x}_1$, with $\mathbf{x}_i = \sigma(\mathbf{x}_i - \mathbf{x}_1), 0 < \sigma \leq 0.5$

The algorithm terminates when the standard deviation of the function values of the current simplex fall below a user-initialized tolerance. When the cycle stops the point of the simplex associated to the lower function value is returned as proposed optimum

The values of the parameters α, γ, ρ and σ were left to default of `scipy`: $\alpha = 1, \gamma = 2, \rho = 0.5, \sigma = 0.5$.

Per valutare eventuali miglioramenti nella performance abbiamo provato ad utilizzare un algoritmo che fosse gradient-based. Nello specifico ho provato ad utilizzare l'algoritmo di Sequential Least Squares Programming (SLSQP) nella versione implementata all'interno della libreria scipy.

3.2.2 Results

Di seguito riporto i risultai che abbiamo ottenuto utilizzando gli algoritmi descritti in precedenza per la minimizzazione del RB. Per prima cosa ho studiato come l'utilizzo di algoritmi di ottimizzazione che agiscono su ampiezza e durata ... sono in grado di migliorare la assignment fidelity partendo da una calibrazione non ottimalel de qubit

3.3 CMA-ES

3.3.1 Algorithm description

Covariance Matrix Adaptation Evolution Strategy (`CMA-ES` [18]), is a population-based evolutionary algorithm designed for optimizing complex, non-convex, and high-dimensional functions.
It belongs to the broader class of Evolution Strategies (ES), a subset of Evolutionary Algorithms (EAs)(see [19]), and is particularly effective for black-box optimization where gradient information is unavailable.

Evolution Strategies (ES) are a class of optimization methods that employ self-adaptive mechanisms to explore the search space efficiently. Unlike classical optimization techniques that rely on gradient descent, ES leverage stochastic sampling to navigate rugged and multimodal landscapes. In this context, CMA-ES is an adaptive stochastic search method that iteratively refines a probability distribution over the search space. Unlike traditional Genetic Algorithms (GAs), which rely on crossover and mutation operators, CMA-ES employs a multivariate normal distribution to generate candidate solutions. The method adaptively updates the distribution's mean and covariance matrix based on the fitness of sampled points.

The fundamental idea behind CMA-ES is the use of a multivariate Gaussian distribution to model promising search directions. Let μ_t denote the mean of the distribution at iteration t, and Σ_t the covariance matrix. Then, a new population of λ candidate solutions $\mathbf{x}_i^{(t+1)} \sim \mu_t + \sigma_t \mathcal{N}(0, \Sigma_t)$, where σ_t is a step size controlling the exploration.

The CMA-ES algorithm follows the following steps:

1. If not otherwise specified, the initial parameters are set: mean vector $\mu_\mathbf{0}$, covariance matrix $\Sigma_0{}^2$, step size σ_0, population size λ

2. Generate λ new candidate solutions $\mathbf{x}_i$ according to a multivariate normal distribution.

3. Evaluate the objective function $f(\mathbf{x}_i)$ for each candidate solution.

4. Sort the new candidate solutions based on fitness: $f(\mathbf{x}_0) \leq ... \leq f(\mathbf{x}_\lambda)$.

$^2\Sigma_0 = \mathbb{I}$ for isotropic search

5. Update the mean vector μ with the $m = \lfloor \lambda/2 \rfloor$ top performing solutions:

$$\mu \leftarrow \sum_{i=0}^{m} \mathbf{w}_i \mathbf{x}_1, \tag{3.6}$$

   where $\mathbf{w}_i$ are internally defined weights.

6. Update the isotropic and anisotropic evolution path $\mathbf{p}_\sigma$, $\mathbf{p}_c$ [3].

7. Update the covariance matrix:

$$C \leftarrow (1 - c_1 - c_\mu)C + c_1 \mathbf{p}_c \mathbf{p}_c^T + c_\mu \sum_{i=1}^{\mu} w_i \mathbf{y}_i \mathbf{y}_i^T, \tag{3.7}$$

   where c_1 and c_μ are learning rates and $\mathbf{y}_i$ represents the deviation of the i-th cnadidate solution from the mean $\mathbf{mu}$.

8. Update the step size using a cumulative path evolution mechanism

$$\sigma \leftarrow \sigma \cdot \exp\left(\frac{c_\sigma}{d_\sigma} \left(\|\mathbf{p}_\sigma\| - E\|\mathcal{N}(0, I)\| \right) \right), \tag{3.8}$$

   where c_σ is the learning rate for step-size adaptation, d_σ is a damping factor $\|\mathbf{p}_\sigma\|$ is the length of the evolution path and $E\|\mathcal{N}(0, I)\|$ is the expected length of a standard normally distributed random vector.

Nel seguito, a meno che non sia diversamente specificato, i parametri sono stati inizializzati ai valori di default della libereria `CMA-ES`

3.3.2   Results

3.4   Optuna

3.4.1   Algorithm description

In addition to the optimization methods mentioned earlier, the Tree-Structured Parzen Estimator (TPE) method was employed, using its implementation available in the `optuna` library [20].

Tree-Structured Parzen Estimator (TPE) is a Sequential Model-Based Optimization (SMBO) approach [21]. SMBO methods sequentially construct models to approximate the performance of optimization parameters based on historical measurements, and then subsequently choose new parameters values to test based on this model. [22] At the heart of SMBO is the idea of building a surrogate model, which is used to predict the objective function's values for unseen parameters configurations. The surrogate model is iteratively updated as new observations are made, and the optimization process balances exploration, which focuses on uncertain regions of the search space, and exploitation, which focuses on areas that are more likely to improve the objective based on past evaluations. This balance ensures that the optimization process makes efficient use of resources and avoids wasting time on suboptimal regions.

The TPE algorithm is a probabilistic model-based optimization method that uses non-parametric density estimation to guide the search. The TPE algorithm differs from traditional Bayesian optimization approaches, such as Gaussian Process-based methods, in its modeling strategy. Rather than directly approximating the objective function, TPE constructs two separate probabilistic models:

- $p(x|y < y^*)$, the likelihood of observing a parameter configuration x given that the objective function value y is below a chosen threshold y^*.

- $p(x|y \geq y^*)$, the likelihood of observing X for less promising function values.

These probability densities are estimated using non-parametric methods such as kernel density estimation (KDE). New candidate points are then generated by sampling from $p(x|y < y^*)$, favoring

---

[3]For details on the update process of the evolution paths see [18].

configurations that are expected to yield lower objective values. The threshold y^* is typically set as a quantile of observed values, ensuring a focus on the most promising regions of the search space.

The TPE method is the default optimization strategy in `Optuna`, an advantage in the optimization algorithm as implemented in optuna is the addition of an authomatic *pruning* mechanism that stops unpromising trials early, which can significantly speed up the optimization process by avoiding unnecessary computations. In our case, this is particularly relevant because the execution of the RB routine, which is performed at each call to the cost function, requires [insert approximate execution time]

As implemented in our code, the default pruner used is the median pruner 'optuna.pruners.MedianPruner'. This pruner works by evaluating the intermediate results of a trial and comparing them to the median of completed trials at the same step. If the current trial's performance is worse than the median, it is pruned to prevent wasting computational resources on unpromising configurations.

3.4.2   Results

D1 - steps

D1 - 1000 steps

D2 - 1000 steps

3.5   RB optimization conclusions

Chapter 4

Pulses analysis and tuning

Having concluded that closed-loop optimization would not significantly improve fidelity, we shifted our focus towards improvement and implementation of individual protocols to improve the accuracy of qubit operations.

In this chapter, I present the results of two additions to the `Qibocal` software. The first is the inclusion of an $RX90$ gate as a native gate, which can enhance the performance of protocols requiring qubit rotations of $\frac{\pi}{2}$. The second is the implementation of the cryoscope, a routine first described in [23], which is useful for correcting distortions in the magnetic flux pulse applied to the SQUID.

4.1 RX90 calibration

4.2 Flux pulse correction

4.2.1 Notes on signal analysis

4.2.2 Cryoscope

The experiment that we describe in this section was first introduced in [23], the goal is to determine predistortions that needs to be applied to a flux pulse signal so that the qubit receives the flux pulse as intended by the experimenter.

As explained in section 1.2, accurate dynamical control of qubit frequency is of key importance to realize single- and two-qubit gates. One of the on-chip control variable that is used on QunatumWare chip is the magnteic flux through a SQUID loop, the signal for magnetic flux control originates from an arbitrary waveform genarator (AWG) which operates at room temperature.

As the signal propagates through various electrical components along the control line leading to the quantum device it undergoes linear dynamical distortions. If not properly compensated, these distortions can degrade gate performance, impacting experiments fidelity and repeatability.

In [23] is proposed a technique to characterize flux-pulse distortions induced by components inside the dilution refrigerator by directly measuring the qubit state. In this protocol we send the qubit a pulse sequence where a flux pulse of varying duration τ is embedded between two $\frac{\pi}{2}$ pulses which are always separated by a fixed interval T_{sep}.

The first $\frac{\pi}{2}$ pulse rotates the qubit of $\frac{\pi}{2}$ around the y axis of the Bloch sphere changing its state from $|0\rangle$ to $\frac{|0\rangle + |1\rangle}{\sqrt{2}}$.

When a flux pulse $\Phi_{Q,\tau}(t)$ of duration τ is sent to the qubit[1] after the first $\frac{\pi}{2}$ pulse, the qubits evolve to the state $\frac{|0\rangle + e^{i\varphi_\tau}|1\rangle}{\sqrt{2}}$ with relative quantum phase

$$\frac{\varphi_\tau}{2\pi} = \int_0^{T_{sep}} \Delta f_Q(\Phi_{Q,\tau(t)}) \mathrm{d}t = \int_0^\tau \Delta f_Q(\Phi_{Q,\tau(t)}) \mathrm{d}t + \int_\tau^{T_{sep}} \Delta f_Q(\Phi_{Q,\tau(t)}) \mathrm{d}t \qquad (4.1)$$

where in the second step we separated the contributions from flux response up to τ and the turn-off transient.

---

[1]To send a $\Phi_{Q,\tau}(t)$ flux pulse we are actually sending a $V_{\mathrm{in},\tau}(t)$ voltage pulse through the electronics

The experiment is then completed with a $\frac{\pi}{2}$ rotation aroud the y- or x-axis of the Bloch sphere to measure respectively the $\langle X \rangle$ or $\langle Y \rangle$ components of the Bloch vector when applying the measurement gate MZ. From the measurement of $\langle X \rangle$ and $\langle Y \rangle$ we can extract the relative phase ϕ_τ.

Then we can estimate $\Phi_Q(t)$ in the interval $[\tau, \tau + \Delta\tau]$ as follows. From the measurement of $\phi_{\tau+\Delta\tau}$ and ϕ_τ we can compute $\overline{\Delta f_R}$:

$$\overline{\Delta f_R} = \frac{\phi_{\tau+\Delta\tau} - \phi_\tau}{2\pi\Delta\tau} \tag{4.2}$$

$$= \frac{1}{\Delta\tau} \left(\int_0^{\tau+\Delta\tau} \Delta f_Q(\Phi_{Q,\tau+\Delta\tau}(t))dt + \int_{\tau+\Delta\tau}^{T_{sep}} \Delta f_Q(\Phi_{Q,\tau+\Delta\tau}(t))dt \right) \tag{4.3}$$

$$- \frac{1}{\Delta\tau} \left(\int_0^{\tau} \Delta f_Q(\Phi_{Q,\tau}(t))dt - \int_{\tau}^{T_{sep}} \Delta f_Q(\Phi_{Q,\tau}(t))dt \right) \tag{4.4}$$

$$= \frac{1}{\Delta\tau} \left(\int_{\tau}^{\tau+\Delta\tau} \Delta f_Q(\Phi_{Q,\tau+\Delta\tau})dt + \int_{\tau+\Delta\tau}^{T_{sep}} \Delta f_Q(\Phi_{Q,\tau+\Delta\tau}(t))dt - \int_{\tau}^{T_{sep}} \Delta f_Q(\Phi_{Q,\tau}(t))dt \right) \tag{4.5}$$

$$= \frac{1}{\Delta\tau} \int_{\tau}^{\tau+\Delta\tau} \Delta f_Q(\Phi_{Q,\tau+\Delta\tau})dt + \varepsilon \tag{4.6}$$

with

$$\varepsilon = \frac{1}{\Delta\tau} \left(\int_{\tau+\Delta\tau}^{T_{sep}} \Delta f_Q(\Phi_{Q,\tau+\Delta\tau}(t))dt - \int_{\tau}^{T_{sep}} \Delta f_Q(\Phi_{Q,\tau}(t))dt \right)$$

The phase contribution from the turn-off transients is minimal due to the sharp return to the first-order flux-insensitive sweet spot of the nearly quadratic $\Delta f_Q(\Phi_Q)$; numerical simulations suggest that $|\varepsilon|/\Delta f_R$ remains within the range of approximately 10^{-2} to 10^{-3} for typical dynamical distortions in commonly used electronic components[24][25], for this reason it will be neglected.

Then we can obtain the reconstructed flux pulse $\Phi_R(t)$ inverting eq. 1.32.

Pulse reconstruction

Corrections study

4.2.3   Corrected pulse

4.2.4   Filter determination

IIR

FIR

for description and notes on `CMA-ES` see section 3.2

Output filters in QM

Chapter 5

Conclusions

Bibliography

[1] R. P. Feynman, "Simulating physics with computers", International Journal of Theoretical Physics **21**, 467–488 (1982).

[2] K. L. Brown, W. J. Munro, and V. M. Kendon, *Using quantum computers for quantum simulation*, 2010.

[3] I. M. Georgescu, S. Ashhab, and F. Nori, "Quantum simulation", Reviews of Modern Physics **86**, 153–185 (2014).

[4] A. Montanaro, "Quantum algorithms: an overview", npj Quantum Information **2**, 15023 (2016).

[5] Z. Chen, "Metrology of quantum control and measurement in superconducting qubits", ProQuest ID: Chen_ucsb_0035D_13771, Merritt ID: ark:/13030/m50p5xxf, PhD thesis (UC Santa Barbara, 2018).

[6] D. P. DiVincenzo, "The physical implementation of quantum computation", Fortschritte der Physik **48**, 771–783 (2000).

[7] R. Manenti and M. Motta, *Quantum Information Science*, en, 1st ed. (Oxford University PressOxford, Aug. 2023).

[8] S. Olivares, *Lecture notes on quantum computing*, Nov. 2021.

[9] J. Koch, T. M. Yu, J. Gambetta, A. A. Houck, D. I. Schuster, J. Majer, A. Blais, M. H. Devoret, S. M. Girvin, and R. J. Schoelkopf, "Charge insensitive qubit design derived from the Cooper pair box", en, Physical Review A **76**, arXiv:cond-mat/0703002, 042319 (2007).

[10] D. Vion, A. Aassime, A. Cottet, P. Joyez, H. Pothier, C. Urbina, D. Esteve, and M. H. Devoret, "Manipulating the quantum state of an electrical circuit", Science **296**, 886–889 (2002).

[11] M. Mohseni, A. T. Rezakhani, and D. A. Lidar, "Quantum-process tomography: resource analysis of different strategies", Phys. Rev. A **77**, 032322 (2008).

[12] A. A. Mele, "Introduction to haar measure tools in quantum information", Quantum **8**, 1340 (2024).

[13] J. Emerson, R. Alicki, and K. Życzkowski, "Scalable noise estimation with random unitary operators", Journal of Optics B: Quantum and Semiclassical Optics **7**, S347 (2005).

[14] E. Knill, D. Leibfried, R. Reichle, J. Britton, R. B. Blakestad, J. D. Jost, C. Langer, R. Ozeri, S. Seidelin, and D. J. Wineland, "Randomized Benchmarking of Quantum Gates", en, Physical Review A **77**, arXiv:0707.0963 [quant-ph], 012307 (2008).

[15] P. Virtanen, R. Gommers, T. E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovski, P. Peterson, W. Weckesser, J. Bright, S. J. van der Walt, M. Brett, J. Wilson, K. J. Millman, N. Mayorov, A. R. J. Nelson, E. Jones, R. Kern, E. Larson, C. J. Carey, İ. Polat, Y. Feng, E. W. Moore, J. VanderPlas, D. Laxalde, J. Perktold, R. Cimrman, I. Henriksen, E. A. Quintero, C. R. Harris, A. M. Archibald, A. H. Ribeiro, F. Pedregosa, P. van Mulbregt, and SciPy 1.0 Contributors, "SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python", Nature Methods **17**, 261–272 (2020).

[16] J. Kelly, R. Barends, B. Campbell, Y. Chen, Z. Chen, B. Chiaro, A. Dunsworth, A. G. Fowler, I.-C. Hoi, E. Jeffrey, A. Megrant, J. Mutus, C. Neill, P. J. J. O'Malley, C. Quintana, P. Roushan, D. Sank, A. Vainsencher, J. Wenner, T. C. White, A. N. Cleland, and J. M. Martinis, "Optimal Quantum Control Using Randomized Benchmarking", en, Physical Review Letters **112**, 240504 (2014).

[17] J. A. Nelder and R. Mead, "A simplex method for function minimization", The Computer Journal **7**, 308–313 (1965).

[18] M. Nomura and M. Shibata, *Cmaes : a simple yet practical python library for cma-es*, 2024.

[19] A. N. Sloss and S. Gustafson, *2019 evolutionary algorithms review*, 2019.

[20]T. Akiba, S. Sano, T. Yanase, T. Ohta, and M. Koyama, "Optuna: a next-generation hyperparameter optimization framework", in Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery and data mining (2019).

[21]F. Hutter, H. H. Hoos, and K. Leyton-Brown, "Sequential model-based optimization for general algorithm configuration", in Learning and intelligent optimization, edited by C. A. C. Coello (2011), pp. 507–523.

[22]B. Shahriari, K. Swersky, Z. Wang, R. P. Adams, and N. de Freitas, "Taking the human out of the loop: a review of bayesian optimization", Proceedings of the IEEE **104**, 148–175 (2016).

[23]M. A. Rol, L. Ciorciaro, F. K. Malinowski, B. M. Tarasinski, R. E. Sagastizabal, C. C. Bultink, Y. Salathe, N. Haandbaek, J. Sedivy, and L. DiCarlo, "Time-domain characterization and correction of on-chip distortion of control pulses in a quantum processor", en, Applied Physics Letters **116**, arXiv:1907.04818 [quant-ph], 054001 (2020).

[24]R. Sagastizabal, X. Bonet-Monroig, M. Singh, M. A. Rol, C. C. Bultink, X. Fu, C. H. Price, V. P. Ostroukh, N. Muthusubramanian, A. Bruno, M. Beekman, N. Haider, T. E. O'Brien, and L. DiCarlo, "Experimental error mitigation via symmetry verification in a variational quantum eigensolver", Phys. Rev. A **100**, 010302 (2019).

[25]N. K. Langford, R. Sagastizabal, M. Kounalakis, C. Dickel, A. Bruno, F. Luthi, D. J. Thoen, A. Endo, and L. DiCarlo, "Experimentally simulating the dynamics of quantum light and matter at deep-strong coupling", Nature Communications **8**, 1715 (2017).

Acknowledgement

Appendix A

Starting with the definition of average gate fidelity:

$$F = \int d\psi \, \langle\psi| \, U^\dagger \mathcal{E}(|\psi\rangle \langle\psi|) U \, |\psi\rangle$$

The quantum maps that represent a depolarizing channel is

$$\mathcal{E}(\rho) = (1-d)U\rho U^\dagger + d\frac{I}{2^n} \tag{5.1}$$

When we substitute it in the avarage gate fidelity definition we get:

$$
\begin{aligned}
F &= \int d\psi \, \langle\psi| \, U^\dagger \left[(1-d)U \, |\psi\rangle \langle\psi| \, U^\dagger + d\frac{\mathbb{I}}{2^n} \right] U \, |\psi\rangle \\
&= \int d\psi \, \langle\psi| \, U^\dagger \left[(1-d)U \, |\psi\rangle \langle\psi| \, U^\dagger \right] U \, |\psi\rangle + \int d\psi \, \langle\psi| \, U^\dagger \left[d\frac{I}{2^n} \right] U \, |\psi\rangle \\
&= (1-d) \int d\psi \, \langle\psi| \, |\psi\rangle \langle\psi| \, |\psi\rangle + \frac{d}{2^n} \int d\psi \, \langle\psi| \, U^\dagger \mathbb{I} U \, |\psi\rangle \\
&= (1-d) + \frac{d}{2^n} \int d\psi \, \langle\psi| \, |\psi\rangle \\
&= (1-d) + \frac{d}{2^n} = 1 - d + \frac{d}{2^n} = 1 - d\left(1 - \frac{1}{2^n}\right) \\
&= 1 - d\frac{2^n - 1}{2^n}
\end{aligned}
$$

Appendix B