
**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


**1 Superconducting qubits** **1**
- 1.1 Introduction . . . 1
- 1.2 Transmon qubits . . . 2
  - 1.2.1 Density matrix . . . 2
- 1.3 Quantum operations . . . 2
- 1.4 Transmon qubit . . . 3
  - 1.4.1 Rabi experiments . . . 3
  - 1.4.2 cQED systems . . . 3

**2 Qibo** **5**

**3 RB fidelity optimization** **7**
- 3.1 Randomized Benchmarking . . . 7
- 3.2 Scipy optimization methods . . . 8
  - 3.2.1 Algorithm description . . . 8
  - 3.2.2 Results . . . 9
- 3.3 CMA-ES . . . 9
  - 3.3.1 Algorithm description . . . 9
  - 3.3.2 Results . . . 10
- 3.4 Optuna . . . 10
  - 3.4.1 Algorithm description . . . 10
  - 3.4.2 Results . . . 10
- 3.5 RB optimization conclusions . . . 10

**4 Pulses analysis and tuning** **11**
- 4.1 RX90 calibration . . . 11
- 4.2 Flux pulse correction . . . 11
  - 4.2.1 Notes on signal analysis . . . 11
  - 4.2.2 Cryoscope . . . 11
  - 4.2.3 Corrected pulse . . . 12
  - 4.2.4 Filter determination . . . 12

**5 Conclusions** **13**

# 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 $\alpha$ and $\beta$.

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 $\theta$, with $0 \leq \theta \leq \pi$ and the azimuthal angle $\varphi$, 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

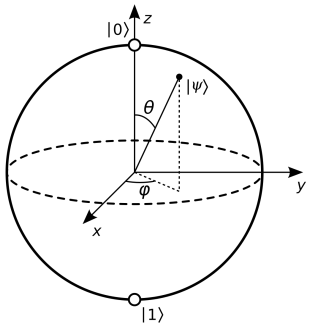


Figure 1.1: Example of qubit state representation on the Bloch sphere

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

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

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

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

where $\omega_q = (E_1 - E_0)/\hbar$ is the qubit transition frequency, and we have used the relation $\hat{\sigma}^+ \hat{\sigma}^- = |1\rangle \langle 1|$. 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:

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

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 [5], [6]:

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.
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

### 1.2.1 Density matrix

## 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 $\hat{\rho}'$:

$$\mathcal{E}(\rho) = \rho'. \tag{1.5}$$

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 $\rho$, is replaced by $\frac{\mathbb{I}}{2^n}$, with probability $d$. This process can be represented with a quantum map as follows:

$$\mathcal{E}_{dc}(\rho) = d\frac{\mathbb{I}}{2^n} + (1-d)\rho \tag{1.6}$$

## 1.4 Transmon qubit

### 1.4.1 Rabi experiments

### 1.4.2 cQED systems

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[7]

$$f_Q(\Phi_Q) \approx \frac{1}{h}\left(\sqrt{8E_J E_C \cos\left(\pi\frac{\Phi_Q}{\Phi_0}\right)} - E_C\right) \tag{1.7}$$

where $E_C$ is the charging energy, $E_J$ is the sum of the Josephson energies of the individual Josephson junctions, $\Phi_0$ is the flux quantum, and $h$ is the Planck's constant.

# Chapter 2

# Qibo

# 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 [8] 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 [9] and applied in sequence to the qubit will avarage out to behave like a depolarizing channel [10]. 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 [11].

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

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

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

In realsystems, where noise is present, eq. **??** does not hold; instead randomization with Clifford gates behave as a depolarizing channel 1.6 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` [12] evitando metodi gradient-based considerato il landscape potenzialmente complicato della funzione RB. 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 [13].

The Nelder-Mead optimization method, originally introduced by Nelder and Mead in 1965 [14], 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 $\alpha, \gamma, \rho$ and $\sigma$ were left to default of `scipy`: $\alpha = 1, \gamma = 2, \rho = 0.5, \sigma = 0.5$.

### 3.2.2 Results

## 3.3 CMA-ES

### 3.3.1 Algorithm description

Covariance Matrix Adaptation Evolution Strategy (`CMA-ES` [15]), 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 [16]), 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 $\mu_t$ denote the mean of the distribution at iteration $t$, and $\Sigma_t$ the covariance matrix. Then, a new population of $\lambda$ candidate solutions $\mathbf{x}_i^(t+1) \sim \mu_t + \sigma_t \mathcal{N}(0, \Sigma_t)$, where $\sigma_t$ 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 $\sigma_0$, population size $\lambda$

2. Generate $\lambda$ 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: $f(\mathbf{x}_0) \leq ... \leq f(\mathbf{x}_\lambda)$.

5. Update the mean vector $\mu$ 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].

[2] $\Sigma_0 = \mathbb{I}$ for isotropic search
[3] For details on the update process of the evolution paths see [15].

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 $c_\mu$ are learning rates and $\mathbf{y}_i$ represents the deviation of the $i$-th cnadidate solution from the mean **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 $c_\sigma$ is the learning rate for step-size adaptation, $d_\sigma$ 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 [17].

Tree-Structured Parzen Estimator (TPE) is a Sequential Model-Based Optimization (SMBO) approach [18]. 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. [19] 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 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`

### 3.4.2 Results

## 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 [20], 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 [20], 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 1.4.2, 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 [20] 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 $\tau$ 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 $\tau$ 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

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

where in the second step we separated the contributions from flux response up to $\tau$ 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 $\phi_\tau$.

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 $\phi_\tau$ 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[21][22], for this reason it will be neglected.

Then we can obtain the reconstructed flux pulse $\Phi_R(t)$ inverting eq. 1.7.

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

[6]R. Manenti and M. Motta, *Quantum Information Science* (Oxford University Press, Aug. 2023) Chap. 5, pp. 173–176.

[7]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", Phys. Rev. A **76**, 042319 (2007).

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

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

[10]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).

[11]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).

[12]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).

[13]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).

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

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

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

[17]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).

[18]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.

[19]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).

[20]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).

[21]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).

[22]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