



**UNIVERSITÀ
DI PADOVA**

UNIVERSITÀ DEGLI STUDI DI PADOVA

Dipartimento di Ingegneria Industriale

Corso di Laurea Magistrale in Electrical Engineering

Extraction of equivalent circuit models of superconducting quantum devices

Relatore:

Ing. Riccardo Torchio

University of Padova

Laureando:

Michael Garberino

Matricola 2159031

ACADEMIC YEAR 2025/2026

GRADUATION DATE: 02/10/2026

Reproducibility

Fundamental repositories for the development of this work are the following:



[quantum-device-consortium/qdw26-workshop-materials/tree/main](https://github.com/quantum-device-consortium/qdw26-workshop-materials/tree/main)



[qiskit-community/qiskit-metal](https://github.com/qiskit-community/qiskit-metal)

Summary

Scaling superconducting processors beyond a few qubits turns device-level parameters: frequencies, couplings, crosstalk and lifetimes into a design bottleneck, since they depend on the whole chip’s electromagnetic environment rather than on isolated junctions. This work builds a 7-port circuit model of a five-qubit processor: an Ansys HFSS scattering simulation of a Qiskit Metal layout is reduced by vector fitting to a SPICE macromodel, used both for nonlinear time-domain simulation and, independently, for analytic black-box quantisation, so the two cross-check each other. They agree on five well-separated qubit modes (anharmonicities between -206 and -196 MHz), with the classical Kerr shift of the time-domain model of the same order as the analytic anharmonicity; three further cavity-like modes are identified. Crosstalk is dominated by capacitive coupling to adjacent qubits when driven individually, and is markedly worse through the shared feed line. The main limitations are the macromodel’s residual non-passivity, which bounds the Purcell-limited T_1 only from below, and the lack of a dedicated control-line port, which restricts crosstalk to the coupling already present in the shared environment.

Sommario

La scalabilità dei processori superconduttori oltre pochi qubit rende la previsione accurata di parametri come frequenze, accoppiamenti, crosstalk e tempi di vita un collo di bottiglia progettuale, poiché tali grandezze dipendono dall'intero ambiente elettromagnetico e non dalla singola giunzione isolata. Questo lavoro costruisce un modello circuitale a 7 porte di un processore a cinque qubit: una simulazione a parametri di scattering in Ansys HFSS di un layout Qiskit Metal viene ridotta tramite vector fitting a un macromodello SPICE, utilizzato sia per una simulazione non lineare nel dominio del tempo, sia, indipendentemente, per una quantizzazione black-box analitica, cosicché le due descrizioni si verificano a vicenda. I due approcci concordano su cinque modi di qubit ben separati (anarmonicità tra -206 e -196 MHz), con un Kerr shift classico del modello nel dominio del tempo dello stesso ordine dell'anarmonicità analitica; vengono inoltre identificati tre ulteriori modi cavity-like. Il crosstalk è dominato dall'accoppiamento capacitivo tra qubit fisicamente adiacenti quando pilotati singolarmente, ed è nettamente peggiore attraverso la feed line condivisa. I limiti principali sono la non-passività residua del macromodello fittato, che vincola solo dal basso la stima del T_1 Purcell-limited, e l'assenza di una porta dedicata alle linee di controllo, che limita l'analisi del crosstalk all'accoppiamento già presente nell'ambiente condiviso.

Declaration of usage of generative AI and AI-assisted technologies

Generative AI and AI-assisted technologies (a large language model used as a coding and writing assistant) were used during the preparation of this thesis for the following purposes:

- Help in the execution of parts of the project. Particularly code generation and debugging
- Help in the interpretation of intermediate results based on the literature found
- Reviewing scientific and technical literature and highlighting relevant information

All AI-assisted output was checked against primary sources, independent calculations and simulation results. The author reviewed and edited the thesis as needed and takes full responsibility for its content.

Contents

Summary	iii
Sommario	v
Declaration of usage of generative AI and AI-assisted technologies	vii
Introduction	1
1 Superconducting qubit theory	3
1.1 Quantum harmonic oscillator	3
1.2 Josephson Junction	4
1.3 Superconducting qubits	6
1.3.1 Transmon	6
1.3.2 Xmon	7
1.4 Circuit QED	7
2 Quantization methods	9
2.1 Lumped oscillator model	9
2.2 Energy participation ratio	10
2.3 Black-box quantization	10
2.3.1 Vector fitting	12
2.3.2 Multiport generalization: Solgun, Abraham and DiVincenzo	13
3 From layout to equivalent circuit	15
3.1 Layout	15
3.2 Ansys HFSS simulation	16
3.3 Vector fitting and SPICE circuit	19
3.3.1 Data filtering	19
3.3.2 Curve weighting	19
3.3.3 Fit execution and order of the model	20
3.3.4 Diagnostics	21
3.3.5 Final validation	23
3.3.6 SPICE subcircuit	24
4 Black-box quantization: analytical implementation	27

4.1	Preliminary single-port frequency estimate	27
4.2	Multiport mode search	30
4.3	Qubit parameters calculation	32
4.3.1	Participation ratios and effective impedance	32
4.3.2	Self-Kerr and cross-Kerr coefficients	34
4.4	Quality factor and Purcell limit	35
4.5	Cavity-like modes	37
5	Nonlinear model validation	41
5.1	JJ LTspice model	41
5.2	Cheap transient validation	43
5.3	Fast validation against BBQ (FFT)	47
5.4	Frequency sweep: qubit mode and classical Kerr shift	48
5.5	Amplitude sweep: Kerr(amplitude) curve	51
6	Crosstalk	55
6.1	Methodology	55
6.2	Drive crosstalk	56
6.3	Feed line crosstalk	57
	Conclusions	61
	Nomenclature	63
	Bibliography	65

Introduction

Superconducting circuits based on the Josephson junction are today one of the leading platforms for quantum computation. Scaling them from a handful of qubits to registers large enough for error correction turns the accurate prediction of *device-level* parameters like: transition frequencies, anharmonicities, qubit–qubit couplings, control and readout crosstalk, and the lifetimes set by the electromagnetic environment into a design bottleneck. These quantities emerge from the distributed electromagnetic behaviour of the whole chip, so a model that treats the qubits in isolation, or that captures only the near field of each junction, is not sufficient once several qubits share a substrate, a ground plane and a readout line.

Two families of methods are commonly used to go from an electromagnetic simulation to a quantised Hamiltonian: energy-participation-ratio analysis of the eigenmodes [1] and black-box quantisation of the impedance or scattering response seen at the junction ports [2]. Both were originally formulated for one or few modes. This thesis takes the complementary route of building an explicit, multiport *circuit* model: the passive chip is characterised once as a many-port network, condensed into a compact rational macromodel, and reused as a lumped subcircuit into which the Josephson nonlinearity is reinserted. The nonlinear network can then be integrated in the time domain and, in parallel, quantised analytically, so that the two descriptions cross-check each other.

The objectives of the work are: (i) to define and implement a reproducible pipeline from a Qiskit Metal layout of a five-qubit processor to a SPICE-level nonlinear circuit model, passing through a full-wave scattering simulation and a vector-fitting reduction; (ii) to extract the mode frequencies, anharmonicities and cross-Kerr couplings of the fitted network through black-box quantisation, and to compare them with the nonlinear time-domain response of the same network; (iii) to quantify the linear microwave crosstalk between individually driven qubits and from the shared feed line to the qubits; and (iv) to estimate the Purcell-limited relaxation time and to identify where the passivity of the fitted macromodel limits that estimate.

The thesis is organised as follows. Chapter 1 reviews the quantum-harmonic-oscillator and Josephson-junction background needed to describe a transmon as an anharmonic oscillator. Chapter 2 introduces the black-box quantisation formalism used throughout this work to turn a linear electromagnetic model into qubit Hamiltonian parameters, together with the vector-fitting technique used to obtain a compact rational macromodel from scattering data. Chapter 3 de-

scribes the pipeline from the Qiskit Metal layout of the five-qubit chip, through the Ansys HFSS scattering simulation, to the vector-fitted SPICE-compatible network, including its passivity diagnostics. Chapter 4 applies black-box quantisation to the fitted network, extracting mode frequencies, self- and cross-Kerr couplings and a lower bound on the Purcell-limited relaxation time. Chapter 5 reinserts the Josephson nonlinearity into the fitted network as a SPICE subcircuit and validates the resulting time-domain model against the results of Chapter 4, through transient, FFT, frequency-sweep and amplitude-sweep simulations that recover the classical analogue of the Kerr shift. Chapter 6 quantifies the linear microwave crosstalk of the extracted model, both between individually driven qubits and from the shared feed line, relating the dominant leakage channels to the couplings and electromagnetic modes characterised in the previous chapters.

Chapter 1

Superconducting qubit theory

A classical bit is always found in one of two definite states, 0 or 1. A qubit, by contrast, is a two-level quantum system that can exist in a coherent superposition of its basis states $|0\rangle$ and $|1\rangle$, $|\psi\rangle = c_0 |0\rangle + c_1 |1\rangle$, with c_0, c_1 complex probability amplitudes satisfying $|c_0|^2 + |c_1|^2 = 1$ [3]. Measuring the qubit collapses this superposition to a definite outcome, 0 with probability $|c_0|^2$ and 1 with probability $|c_1|^2$: this probabilistic outcome, together with the interference between the complex amplitudes c_0 and c_1 , which has no classical counterpart, is what makes a qubit fundamentally different from a bit. Realizing this behaviour in hardware only requires a well isolated two-level system, with a controllable and non-degenerate transition between its two states. The rest of this chapter builds such a system out of superconducting circuit elements, starting from the quantum harmonic oscillator.

1.1 Quantum harmonic oscillator

The building block for the qubit theory is the harmonic oscillator. Typically presented as the mechanical oscillator, in this context the LC oscillator representation is more suited as it is more directly connected to the quantities used, and so its classical hamiltonian is shown [4]:

$$H = \frac{\Phi^2}{2L} + \frac{Q^2}{2C} \quad (1.1)$$

This formula works in the classical sense, but in order to move forward with a quantum description charge Q must be promoted to the charge operator $\hat{Q}$, and the same for the linked flux $\Phi \rightarrow \hat{\Phi}$ which must follow the commutation relationship:

$$[\hat{\Phi}, \hat{Q}] = \hat{\Phi}\hat{Q} - \hat{Q}\hat{\Phi} = i\hbar \quad (1.2)$$

which means that the two operators don't commute. The final form can thus be obtained as

$$\hat{H} = \frac{1}{2}C\omega_0^2\hat{\Phi}^2 + \frac{\hat{Q}^2}{2C} \quad (1.3)$$

where $\omega_0 = 1/\sqrt{LC}$ is the natural resonance frequency. The key difference here is that, for a quantum system there is no such thing as a continuous spectrum but rather discrete energy levels, and thus the reason to pass through operators, as well as that operators can describe the physical properties of a state. An alternative, and useful form, can be obtained by introducing the annihilation $\hat{a}$ and creation $\hat{a}^\dagger$ operators so that the hamiltonian takes the form:

$$\hat{H} = \hbar\omega_0 \left(\hat{a}^\dagger \hat{a} + \frac{1}{2} \right) \quad (1.4)$$

this is the form typically shown in literature [3], as it is more directly connected to the energy states, from which a key point is derived, the energy eigenstates spacing. The n -th energy eigenstate is

$$E_n = \left(n + \frac{1}{2} \right) \hbar\omega_0 \quad (1.5)$$

clearly the energy levels are discrete and evenly spaced, and this is the reason why, a linear system like an harmonic oscillator, can not work as a qubit [5]. So we have that for the energy eigenstate $|n\rangle$ the energy eigenvalue of $\hat{H}$ is E_n ¹. It is key for physical system to be used as a working qubit that, it can work between two energy states (usually $|0\rangle$ and $|1\rangle$) which have to be isolated from the rest. So it is fundamental that one can selectively drive transitions between the chosen states, without accidentally exciting surrounding states and thus the introduction of a Josephson Junction.

1.2 Josephson Junction

Josephson Junction (JJ), which is composed by two superconducting electrodes separated by an insulator interposed between the two, is the key element for engineered quantum systems [6]. It allows, behaving like a non linear inductor, to obtain unevenly spaced energy levels thus isolating the two working ones from the rest. The difference between first and second transition energies is defined as the anharmonicity:

$$\alpha = \omega_{12} - \omega_{01} = \frac{E_{12} - E_{01}}{\hbar} \text{ [rad/s]} \quad (1.6)$$

here we have that $E_{01} = E_1 - E_0$ and $E_{12} = E_2 - E_1$.

Physically, the working principle of a JJ can be complex, and explained by Bardeen-Cooper-Schrieffer's theory of superconductivity, which states that at low enough temperatures pair of electrons, with opposite spin, start to form becoming the new charge carriers called Cooper pairs.

¹These states are called Fock states and the following equation holds

$$\hat{H} |n\rangle = E_n |n\rangle$$

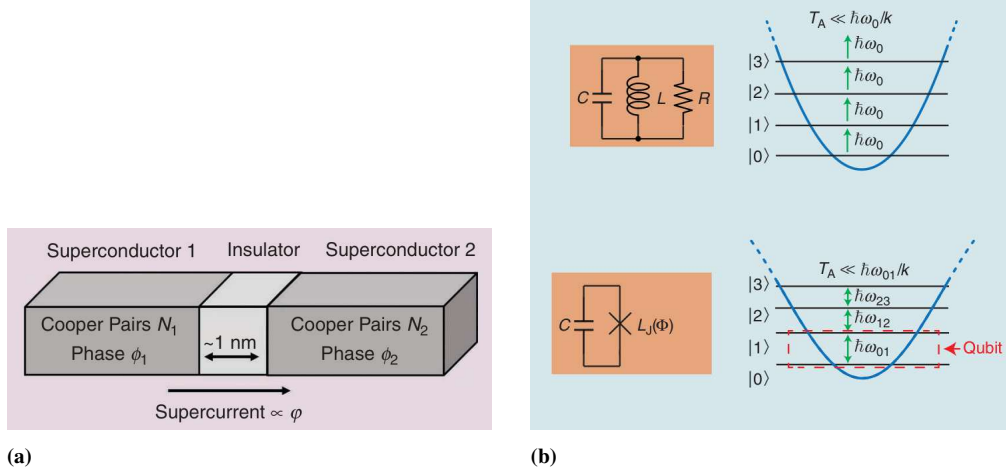


Figure 1.1. (a) Simple scheme of a JJ, typically built as a Al/AlOx/Al [6]. (b) Comparison of fully linear RLC circuit (top) and its evenly spaced energy levels and a JJ shunted by a capacitor (bottom) depicting the effect of non linearity in the energy spectrum [5].

Since, the Pauli exclusion principle does not apply to Cooper pairs, these all fall to the common ground state. This condensate is modeled by its wave function $\Psi(\mathbf{r}, t)$, but what is relevant for this work is that, when this Cooper pairs approach a JJ, the insulation acts as a barrier to their flow. When the potential barrier is "thin", the wave functions at the two conducting ends can overlap, resulting in a Cooper pair being able to tunnel across the insulator, thus resulting in the non linear behavior of a JJ, whose V - I characteristic is described by the Josephson's relation [4], [6]

$$\begin{cases} I = I_c \sin \varphi \\ \frac{\partial \varphi}{\partial t} = \frac{2e}{\hbar} V \end{cases} \quad (1.7)$$

where $I_c = 2eE_j/\hbar$ is the critical current with $E_j = L_j I_c^2$ being the Josephson energy and L_j the Josephson inductance [5]. I_c is the maximum current that can tunnel through the junction before Cooper pairs start breaking due to dissipation. Fundamental quantity is also φ^2 , which represents the phase difference between the wave functions at the two conductive elements of the JJ. From the above equation one obtains the non linear inductance and the inductive energy

²Strictly related to the magnetic flux Φ as

$$\Phi = \frac{\varphi \hbar}{2e} \rightarrow \frac{\partial \Phi}{\partial t} = V$$

this will be relevant later on for the numerical implementation of the JJ. Typically $\hbar/2e = \phi_0$ is referred to as the reduced flux quantum and so $\Phi_0 = 2\pi\phi_0$ can be referred to as a flux quantum

needed in the hamiltonian as

$$L(\varphi) = \frac{L_j}{\cos \varphi} \quad (1.8)$$

$$L_j = \frac{\hbar}{2eI_c} \quad (1.9)$$

$$H_L = \int V(t)I(t)dt = -E_j \cos \varphi \quad (1.10)$$

the cosine term in the inductive energy is the reason of the shape of the energy well in Figure 1.1b. In the physical construction of a JJ it is unavoidable to obtain also a parasitic capacitance C_j , considering also the shunt capacitance C_s we obtain the total hamiltonian

$$\hat{H} = 4E_c(\hat{n} - n_g)^2 - E_j \cos \hat{\varphi} \quad (1.11)$$

where n is the number of Cooper pairs that tunneled through the junction and n_g the initial offset (either forced or from the environment). $E_c = e^2/(2(C_j + C_s))$ is the charging energy of a single electron and controlling the ratio E_j/E_c modifies the spectrum of the hamiltonian, in the case of the used design Xmon qubits have been used which present a ratio $E_j/E_c \gg 1$ [5]

1.3 Superconducting qubits

Traditionally, the first architecture to make use of a JJ, was a Cooper pair box (CPB). It is built as a Josephson junction connecting a superconducting "island" (with no connection to other circuitry) to a "reservoir", making this design extremely sensitive to the number of Cooper pairs that have tunneled. This sensitivity to charges can be exploited to control the operating point of a CPB, for example by modifying n_g , via a voltage source capacitively coupled to the "island", in eq. 1.11. This qubit works in the range $E_j/E_c < 1$ and so choosing well n_g is fundamental for a good working point, but this high dependence on charge makes it susceptible to decoherence due to external charge noise [5].

1.3.1 Transmon

This qubit can be simply seen as a CPB shunted by a large capacitance relative to the stray JJ C_j . Being E_c inversely proportional to the total capacitance the new regime is $E_j/E_c \gg 1$. This modification reduces charge sensitivity following an exponential law, improving coherence times, but it makes the levels more harmonic as the anharmonicity reduces as a weak power law, still large enough to work properly. Thanks to the higher resilience to external noise this architecture is preferred to a CPB [5]

1.3.2 Xmon

This is a variant of the Transmon, in which the shunt capacitance is cross shaped thus giving better interconnectivity with external circuitry. Not only that, but instead of one JJ, it has a loop of two, called superconducting quantum interference device (SQUID). This loop allows to link a DC magnetic flux thanks to which the qubit angular frequency ω_{01} is tunable [7]

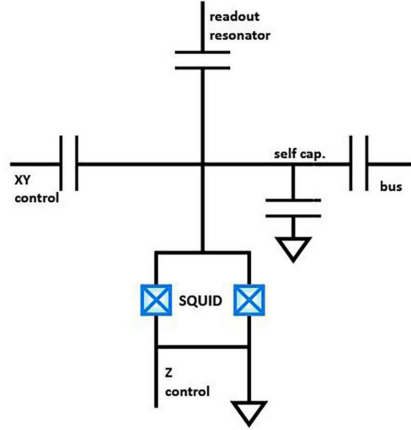


Figure 1.2. Schematic representation of an Xmon qubit [8]

without going into the derivation details, the hamiltonian, with two equal JJ, takes the shape [5]:

$$E_{j,\Phi_{ext}} = 2E_j \cos\left(\frac{\pi\Phi_{ext}}{\Phi_0}\right) \rightarrow H_L = -E_{j,\Phi_{ext}} \cos \varphi \quad (1.12)$$

1.4 Circuit QED

As of now the qubit has been presented as an isolated element and studied like so, but to make a practical chip it must be connected to many other elements. Firstly, shown in Figure 1.2, we need control lines which allow to modify the state of the qubit. Then we need a bus and readout resonators to be able to measure the state it is really in. Then there will be the connection to nearby qubits, either wanted or not, for example to form a register. It is clear that studying a qubit (or many qubits) immersed in a circuit is fundamental as for them to be used in practice that is the environment they will live in [3]. An example can be the measure of the state of the qubit, which is usually done with a dispersive readout, taking advantage of the fact that the resonance frequency of the readout resonator coupled to the qubit shifts depending on the qubit state. So by sending a microwave pulse at the readout input, with a carrier frequency tuned near the resonator frequency, and reading the signal at the output the state can be inferred [3]. The Jaynes-Cummings Hamiltonian provides a description of the interaction between a two-level

system, like the ones described, and a resonant cavity as [3]:

$$\hat{H} = \underbrace{\frac{\hbar\omega_q}{2}\hat{\sigma}_z}_{\text{Qubit hamiltonian}} + \underbrace{\hbar\omega_r\left(\hat{a}^\dagger\hat{a} + \frac{1}{2}\right)}_{\text{Harmonic oscillator}} + \underbrace{\hbar g(\hat{a}\hat{\sigma}^+ + \hat{a}^\dagger\hat{\sigma}^-)}_{\text{Interaction hamiltonian}} \quad (1.13)$$

what is important here is that, to be in dispersive regime, $\Delta = |\omega_q - \omega_r| \gg g$. This description is an improvement as it takes into account the interaction of cavity and qubit, but is far from being enough. In the above described situation there are many other interactions like: qubit-qubit, qubit-feedline, qubit-resonator and many others which is the reason why, quantization methods like the Black-box quantization [2] and/or Energy participation ratio [1] exist.

Chapter 2

Quantization methods

As already mentioned, in a real world application, there are many interactions not considered in the simpler theoretical frameworks presented. For this reason, quantization methods have been developed in order to study complex circuits containing qubits. Regardless of the specific method, the common goal is to reduce the electromagnetic description of the surrounding linear media, together with the non-linear JJ, to a quantized Hamiltonian, from which the excitation frequency (f_{01}), the anharmonicity (α) and the cross-Kerr coupling (χ_{mn})¹ can be extracted. Methods differ in how the reduction is performed, giving priority either to the accuracy or to modularity and computational cost [8]:

- **Quasi static methods:** Numerical techniques that rely on a fundamental hypothesis; field propagation of electromagnetic fields can be neglected. This is usually true when physical dimensions are much smaller than the wavelength of applied fields. These are usually more lightweight and faster methods that lose in accuracy, an example is the Lumped oscillator model (LOM)
- **Full-wave methods:** Numerical methods which solve full Maxwell problems without simplifications. Much heavier and slower allow to reach better accuracy. In this category can be found the Black-box quantization (BBQ) and Energy participation ratio (EPR) methods

2.1 Lumped oscillator model

With this method the physical system is divided into subsystems — regions of the physical layout small enough, or conveniently isolated, to be simulated on their own — that can be simulated independently. Each one of the obtained subsystems is analyzed with an electrostatic simulation to obtain the Maxwell capacitance matrix, and similarly the inductance matrix is obtained. This subdivision must be done carefully, to properly encapsulate interactions between subsystems, and it is sometimes convenient to group several of them into a single *cell* when strongly coupled.

¹The cross-Kerr denotes the frequency shift of one mode due to the excitation of another; between a qubit and a readout cavity it is the dispersive shift

At the end the distributed EM environment is transformed into a network and used to obtain the hamiltonian as

$$\hat{H} = \underbrace{\hat{H}_0 + \sum_{n=1}^K \hat{H}_n}_{\text{Free hamiltonians}} + \underbrace{\sum_{n=0}^{K-1} \sum_{m=n+1}^K \left(\frac{\hat{Q}_n \hat{Q}_m}{C_{nm}^{eff}} + \frac{\hat{\Phi}_n \hat{\Phi}_m}{L_{nm}^{eff}} \right)}_{\text{Interaction hamiltonian}} \quad (2.1)$$

The ability to study independently the subsystems, requiring only simple static simulations, makes this method very flexible and computationally light, at the cost of missing distributed effects and higher-order modes [8].

2.2 Energy participation ratio

Starting from an eigenmode full-wave simulation, the objective of the EPR method is to obtain the participation ratios

$$p_{mj} = \frac{\text{energy of mode } m \text{ stored in element } j}{\text{total energy of mode } m} \quad (2.2)$$

a number between zero and one that represents the amount of energy of mode m that is stored in element j [1]. To obtain so, the JJ is linearized as an inductor, and the whole system is discretized taking particular attention to the junctions, and studied through a conventional eigensolver, the linear component of the hamiltonian is obtained $\hat{H}_{lin}$. The non linearity is restored afterwards by splitting the total Hamiltonian as $\hat{H} = \hat{H}_{lin} + \hat{H}_{nl}$ and calculating, with the found p_{mj} , the non linear part $\hat{H}_{nl}$. EPR method is more accurate than LOM and, doesn't have the problem that BBQ has to do a frequency sweep risking losing information if such sweep is not dense enough but is not suited to obtain an equivalent circuit as it directly interacts with energy fractions and not impedances/scattering response.

2.3 Black-box quantization

BBQ is the method used to develop the work of this thesis as it is perfectly suited to obtain an equivalent circuit. The point being that, with BBQ, one gets a Full-wave method directly working with impedances from which, via a vector fitting algorithm a network can naturally be obtained. The core idea of this quantization method is to treat the whole linear EM environment, including the linear components of the JJ (L_j and C_j), together making the quantity of interest $Z(\omega)$ (or equivalently $Y(\omega)$) seen at the junction's port [2]. This impedance, by virtue of Foster's theorem can be synthesized by the equivalent circuit of parallel LCR oscillators in series as:

$$Z(\omega) = \sum_{n=1}^M \left(j\omega C_n + \frac{1}{j\omega L_n} + \frac{1}{R_n} \right)^{-1} \quad (2.3)$$

with M being the number of modes. The introduction of R_n is needed to represent losses as, a purely LC oscillator is lossless. For each mode the resonance frequency is $\omega_n = 1/\sqrt{L_n C_n}$, by having quantized each mode as an independent harmonic oscillator. Now, re inserting the non linearity of a JJ as a perturbation gives, to the leading quartic order² ($E_{nl} = -\phi_0^2 \varphi^4 / (24 L_j)$), the anharmonicity (self-Kerr) and cross-Kerr coefficients as:

$$\alpha_n = -\frac{C_j L_n}{L_j C_n} E_c \quad (2.4)$$

$$\chi_{mn} = -2\sqrt{\chi_{nn}\chi_{mm}} \quad (2.5)$$

in a compact way also the quality factor of mode n and the Purcell limit on the qubit lifetime are given:

$$Q_n = \frac{\omega_n \text{Im } Y'(\omega_n)}{2 \text{Re } Y(\omega_n)} \quad (2.6)$$

$$T_1 = \frac{Q_{qb}}{\omega_{qb}} \quad (2.7)$$

$$Z_n^{eff} = \frac{2}{\omega_n \text{Im } Y'(\omega_n)} \quad (2.8)$$

where Q_{qb} and ω_{qb} in T_1 are simply Q_n and ω_n evaluated for the mode identified as the qubit, and $Y'(\omega)$ stands for the derivative of $Y(\omega)$ with respect to the frequency. This treatment considers the presence of only one junction, but if N are present the procedure generalizes naturally. The impedance becomes an $N \times N$ matrix $\mathbf{Z}(\omega)$ where one port is associated to each junction, and one of them (here $k = 1$) is chosen as the reference. The frequencies don't depend on this choice even though the participation ratios do. The new formulas obtained are the following [2]

$$\xi_{sp} = \frac{Z_{s1}(\omega_p)}{Z_{11}(\omega_p)} \sqrt{Z_{1p}^{eff}} \quad (2.9)$$

$$\beta_{qq'pp'} = \sum_{s=1}^N \frac{e^2}{24 L_j^{(s)}} \xi_{sq} \xi_{sq'} \xi_{sp} \xi_{sp'} \quad (2.10)$$

$$\alpha_p = -12 \beta_{pppp} \quad (2.11)$$

$$\chi_{qp} = -24 \beta_{qqpp} \quad (2.12)$$

$$\Delta_p = 6 \beta_{pppp} - 12 \sum_q \beta_{qqpp} \quad (2.13)$$

with ξ_{sp} representing the participation ratio of junction s in mode p . Physically, Z_p^{eff} is the

²The non linearity introduced by $\cos \varphi$ is typically treated as a Taylor expansion of the cosine up to the 4th order

characteristic impedance of the equivalent Foster oscillator representing mode p : it is what fixes the amplitude of the zero-point flux fluctuation once the mode is canonically quantized. ξ_{sp} , in turn, measures the fraction of that flux fluctuation that is actually seen at junction s — it is the geometric weight that later determines how much each junction’s nonlinearity contributes to the mode’s self- and cross-Kerr coefficients. The same perturbative treatment also produces a correction Δ_p to the bare mode frequency itself, a Lamb shift that sets the physical $0 \rightarrow 1$ transition frequency of mode p to $\omega_p/2\pi + \Delta_p/h$, rather than the bare linear resonance $\omega_p/2\pi$ found from the linearized circuit alone.

2.3.1 Vector fitting

Vector fitting (VF) is an essential part of the BBQ workflow. For this work, Ansys HFSS, has been used to study the EM environment of the JJs, which gives as an output a table of complex data. Thus the raw output is discrete points, in this case complex valued scattering responses at varying frequencies, that can not realistically be used as they are. The vector fitting algorithm transforms this raw data into a rational function (poles/residues), causal that a transient simulator like LTspice can use. The rational model takes the form

$$Z(s) = \sum_i \frac{R_i}{s - s_i} + d + es \quad (2.14)$$

with s_i representing the poles of the function (either real or complex conjugate), R_i the residues, d a constant term and e a proportional one [9]. The VF doesn’t directly fit the poles, but rather starts from a set of initial poles (randomly placed, linearly distributed or other distributions) and iteratively solves a least-squares linearized problem through the auxiliary function $\sigma(s)$. At each iteration a pole flipping operation is done, so any pole with a positive real part is flipped to make it negative, thus ensuring stability [9], [10].

- **Stability:** poles with negative real part ensure a stable response
- **Passivity:** has a stronger requirement, the network can’t create energy. For a scattering model $\bar{\sigma}[S(j\omega)] \leq 1 \forall \omega$ (maximum gain equal to 1). A fit may be perfectly stable but non passive as this condition is never imposed

When passivity is violated, it can be restored a posteriori: the fitted residues are perturbed by the smallest amount that brings $\bar{\sigma}[S(j\omega)]$ back under 1 across the whole band, a step usually formulated as a constrained optimization problem over a dense frequency grid [11].

Once poles and residues are known, the rational model can be turned into an equivalent circuit, each real pole gives an RL or RC branch, and each complex-conjugate pair gives an RLC branch, in the spirit of the Foster decomposition used for $Z(\omega)$ in the black-box quantization derivation. For the multiport network considered in this work, this step is performed by the `write_spice_subcircuit_s` routine of `skrf`, which realizes the fitted model as a SPICE netlist of RC/RLC branches and controlled sources.

2.3.2 Multiport generalization: Solgun, Abraham and DiVincenzo

Two follow-up papers [12] and [13] push the original idea of [2] toward an exact circuit representation instead of the perturbative approach of the original work. The first difference is that the linear parts of the JJ are not inserted in the linear EM environment. With this choice, the fitted parameters obtained are independent of the choice of L_j and so if a change is made to the junction the whole model does not need to be simulated again, as the linear circuit technically did not change. The second important difference can be described as follows; in [2], the whole rational, positive-real, $Z(s)$ obtained from the classical simulation is fitted pole-by-pole, for example with the Gustavsen algorithm [9], [10] implemented in python package `skrf` [14] used in this work. Instead, [12] proposes to synthesize it as a single, physically exact circuit, Brune's circuit. This new approach moves from a problem of finding roots of a polynomial whose degree grows with the number of ports and poles, already a numerically ill operation, made worse by the very high Q characteristic of superconducting circuits, to a time domain representation in state space. It is worth noting that the vector fitting algorithm used in this work already relies on a state-space representation $\{A, B, C, D, E\}$ internally, to avoid this very root-finding problem during the *fitting* step. [12], [13] instead apply the state-space idea to the *synthesis* step, deriving an exact circuit topology from an already fitted, positive-real impedance, this step is not performed in this work. This also avoids another problem, the fact that a fit doesn't guarantee the obtained $Y(\omega)$ to be positive-real, a condition that if not met breaks the formula 2.6. In this work the followed approach is the one presented in [2] and many of the problems presented arose and will be discussed in the following.

Chapter 3

From layout to equivalent circuit

3.1 Layout

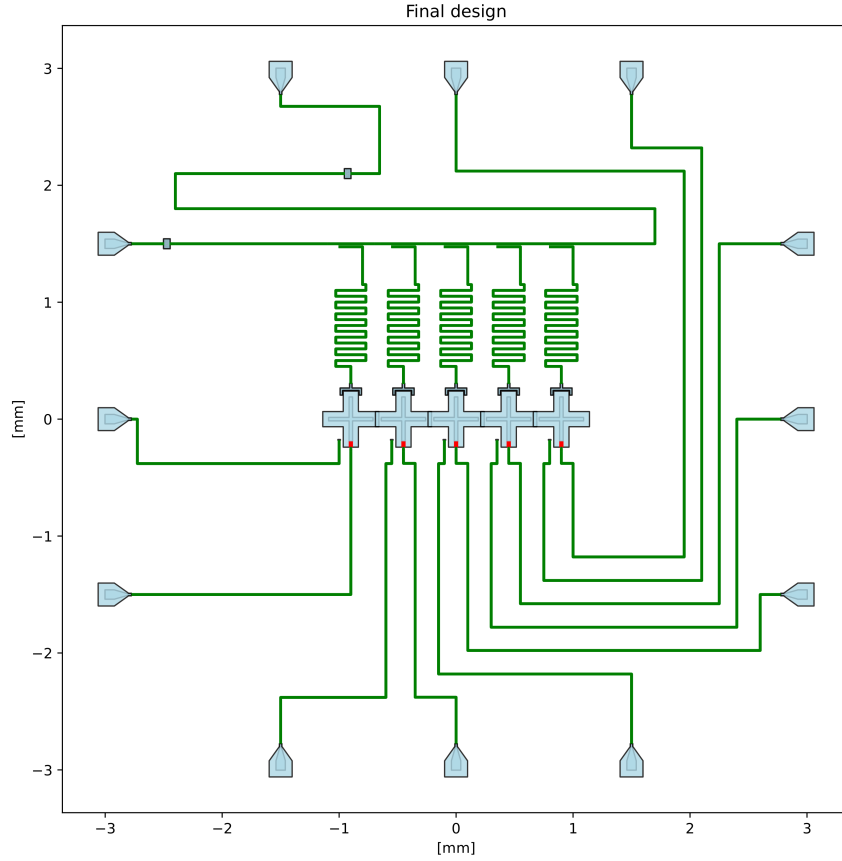


Figure 3.1. Schematic of the simulated design composed of: 5 flux tunable Xmons disposed in a linear array, directly coupled with nearest-neighbour capacitance. Single feed line shared by all qubits, each with its resonator, and individual XY (microwave) and Z (flux) control lines [15]. In the 7-ports model 0,1 = feedline and 2-6 = Q_0 - Q_4 . In red is highlighted the position of the JJs

The starting point to extract a multiport model, is having a working physical design, for this work obtained using Qiskit Metal [16]. The used architecture is inspired by, with some simplifications as it was part of a workshop [15], the 5-Qubit processor introduced by Barends [7], [17].

Each qubit is a `TransmonCrossFL` component, exactly the Xmon architecture of Figure 1.2, placed with a pitch of 0.45 mm in a 6 mm × 6 mm substrate. All 5 Xmons couple to the shared feed line through a `CoupledLineTee`, each to its own resonator as a meandered coplanar waveguide (`RouteMeander`). The five resonators vary in length, in steps of 40 μm spanning from 4.41 mm to 4.57 mm so that they resonate at different frequencies. Here a different choice from the original notebook must be highlighted; in the `RouteMeander` for the definition of the resonators, the original notebook made the choice of a fillet of 20 μm . This created a problem when uploading the geometry in Ansys HFSS, which due to the double turn close to the feedline, would result in a crash. For this reason this value has been reduced to 13 μm , with which the model could successfully be uploaded. In a high frequency environment like the one in study, making sharper corners isn't ideal for practical design, and therefore this choice is expected to have a negative impact in the performance of the chip. The feed line is terminated via `CapNInterdigital` at each end, lastly twelve `LaunchpadWirebond` pads, distributed equally along the four edges, provide external connections for the feed line and all the XY/Z control routes. As of now, all five junctions have been created with the same $L_J = 10 \text{ nH}$, and this is the value the JJs would keep in the original design, here instead, inductances will span from 6 nH to 14 nH with a step of 2 nH rendering them more spectrally distinct. This change is not already applied here, as the junctions are not rendered in the design of Ansys HFSS, instead they are substituted by lumped ports (referenced at 50 Ω); and, as already discussed in §2.3.2, the choice of not embedding the linear component of the JJ inductance in the EM linear environment, allows starting from a single HFSS simulation and single call of vector fitting algorithm to vary L_J afterwards. Therefore, the electromagnetic simulation captures only the geometry and materials, and everything junction related will be added afterwards. Lastly, the number of ports chosen to be accessible is 7, where five are needed for the qubits, and two others as input and output of the feed line. As a consequence, the control routes XY/Z will not be accessible in the circuit model obtained, and thus crosstalk mechanisms occurring specifically at a coupler can not be tested. If a more complex model is needed, for example to study the effects of that kind of crosstalk, one must treat the `LaunchpadWirebond` pads of the control routes as ports, in the same way it is done for the pads connected to the feed line and the junctions.

3.2 Ansys HFSS simulation

Qiskit contains different analysis approaches [8], the most suited for the objective of this work is the `em1 = ScatteringImpedanceSim`, which connects to Ansys HFSS to do a driven-modal simulation. As a result, this outputs exactly the quantities wanted: the broad band scattering response seen from the given ports. Instead of directly launching the simulation, the process is separated to firstly render the circuit and only then the real simulation starts, this is done as through `em1.render` one gets more control over the mesh directly from Python, and

additionally this method allows to calculate the mesh a single time, keeping it the same for all simulations. This has two advantages, the first one is that the simulation time reduces, as at each step Ansys doesn't calculate again the mesh but keeps the one that already converged. Secondly, this avoids mesh noise in different sweeps, as an always changing mesh introduces different errors each time, whereas with the same one the noise floor is consistent.

```

1 # ...
2 em1.setup.freq_ghz = 15 # Mesh frequency
3 em1.setup.max_delta_s = 0.01
4 em1.setup.max_passes = 60
5 em1.setup.min_passes = 3
6 # ...

```

Listing 3.1. First mesh setup.

In the listing 3.1 there are some key parameters for the mesh calculation. The mesh frequency being fixed at 15 GHz is motivated by the fact that no simulation is done above that frequency, and no qubit is expected to exceed 7 GHz with the parameters given. Thus building the mesh around that frequency is reasonable for the scope of this work. The maximum delta, the s parameter matrix is allowed to have between passes has to be set low, as high Q resonances are expected, thus also a larger number of maximum passes is given to the mesh to give the solver breathing room for the mesh convergence.

```

1 # ...
2 "MinimumConvergedPasses:=", 2,
3 "PercentRefinement:=", 15, # Default 30
4 "BasisOrder:=", 2,
5 # ...

```

Listing 3.2. Additional mesh setup.

A very important parameter is the `PercentRefinement`, which represents the percentage of mesh that can be broken down and made finer at each pass. Reducing it from 30 to 15 made it so that the shape of triangles built was much closer to an equilateral triangle, instead leaving it too high would lead to very elongated shapes. Lastly, for some key components a finer mesh has been imposed, by fixing a maximum element size, those being: the Xmon crosses with $7\ \mu\text{m}$, the resonators at $50\ \mu\text{m}$ and the ground plane fixing $250\ \mu\text{m}$. A last implementation detail is the choice of separating the simulation in two sweeps, the first one from 0.9 GHz to 15 GHz separated in 30 blocks of 20001 points each (Ansys allows for a max of 25000 per block) resulting in a step of about 23 kHz. This is done as this region contains many resonances at very high Q factor that with a coarse grid may be lost. The second sweep instead goes from 1 MHz to 0.9 GHz separated in 8 blocks of 1001 points each. At lower frequencies the behavior is smooth and doesn't need many points, this is useful to give a reasonable low frequency behavior to the

fit once it is done. Giving the same grid spacing to the whole sweep would have risked losing resonances at high frequency while wasting a lot of points in flat regions. In this work, the Ansys HFSS simulator has been set as *interpolating*, a different choice not explored could have been a discrete simulation.

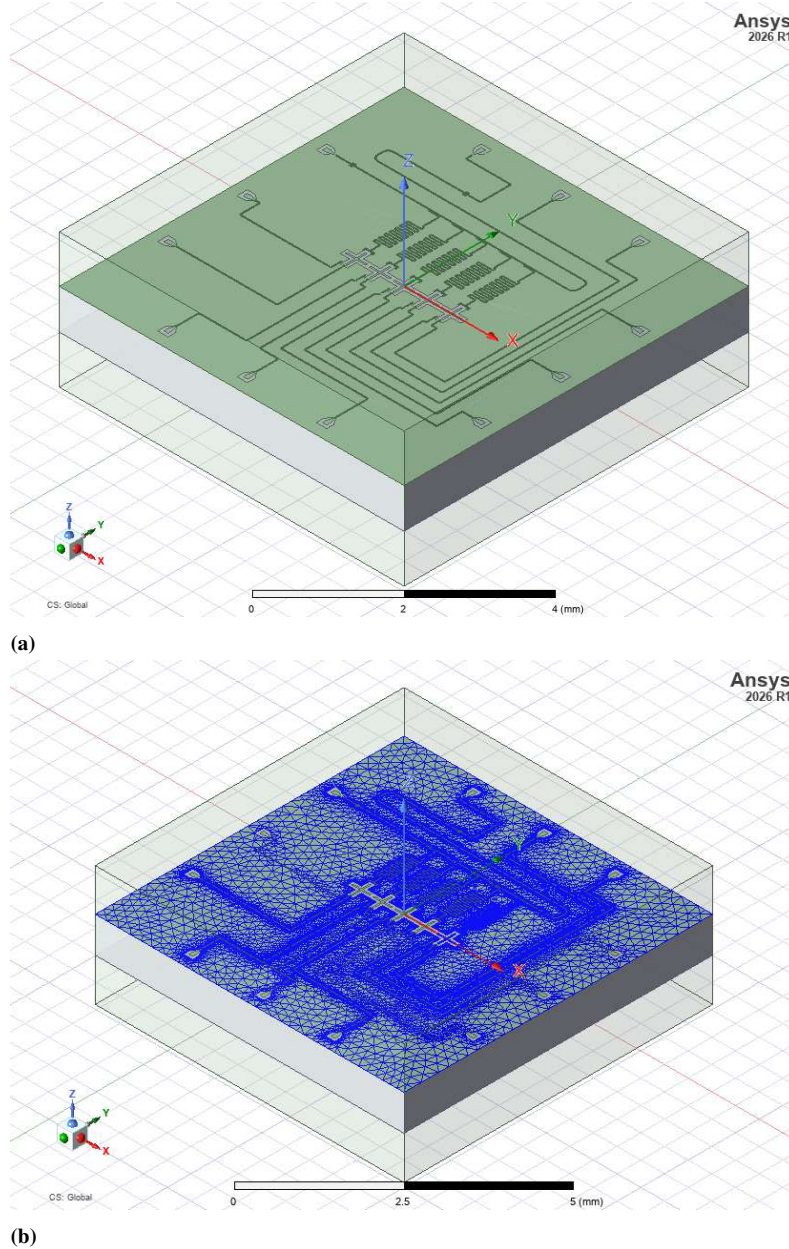


Figure 3.2. (a) Final design rendered in Ansys HFSS to be studied. (b) Mesh of the main ground plane resulting from the imposed parameters, some control lines can be seen to have a worse mesh but this has been accepted to reduce the simulation time, as they will not be accessible as ports.

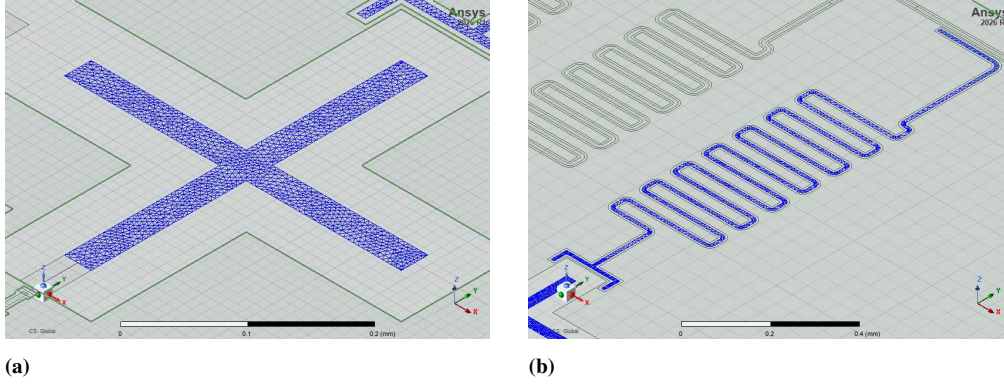


Figure 3.3. (a) Close up of the finer mesh imposed at the Xmon cross. (b) Close up to one of the resonators mesh, all of which have the same mesh parameters imposed.

3.3 Vector fitting and SPICE circuit

3.3.1 Data filtering

From the Ansys simulation, a total of 38 files .s7p with overall 608038 frequency points are obtained. Results represent the 7×7 scattering matrix $\mathbf{S}$ for each of the calculated frequencies, overall having 49 complex valued s_{ij} parameters for each one of the 608038 points extracted. Giving this huge amount of points to the vector fitting algorithm would make it unreasonably slow, as the least-squares problem becomes massive, just because many areas with a simpler curve maintain the same point density as difficult ones. To solve this problem the idea is to filter the obtained data, keeping many points in spiky regions hard to fit, and relaxing the density in flat regions thus reducing a lot the total number of points passed to the VF algorithm. To do so, a Ramer-Douglas-Peucker filter has been used, in its vertical distance variant as the usual point-line distance, precisely near steep resonance peaks, degenerates due to Δx_{Hz} becoming smaller and smaller while Δy_{dB} is big. This translates in the filter scaling as Δx_{Hz} instead of Δy_{dB} , thus stopping to add points too early where they are needed the most. Then, in order to keep a minimum point density everywhere, a maximum gap of 5 MHz has been imposed, so that for any two points $|x_{p1} - x_{p2}| \leq 5 \text{ MHz}$. In addition, another buffer grid has been added, this only in the region 0.9 GHz – 6 GHz, this time with a gap of 1.2 MHz, to help the fit reach a better match of the lower frequency behavior. This filtering process is applied to S_{ij} both in magnitude and phase, choosing a $\Delta s_{\text{dB}} \leq 0.15 \text{ dB}$ and $\Delta s_{\text{deg}} \leq 0.12 \text{ deg}$, the final output consists of 34038 points (about 5.6% of the total).

3.3.2 Curve weighting

The vector fitting algorithm [9], [10] implemented in [14], uses a single scalar weight for each s_{ij} curve, applied uniformly along all the frequency band. Originally, the weight was set equal to the norm of the curve $|s_{ij}|$, but here it has been changed to $1/|s_{ij}|$ directly in the source code

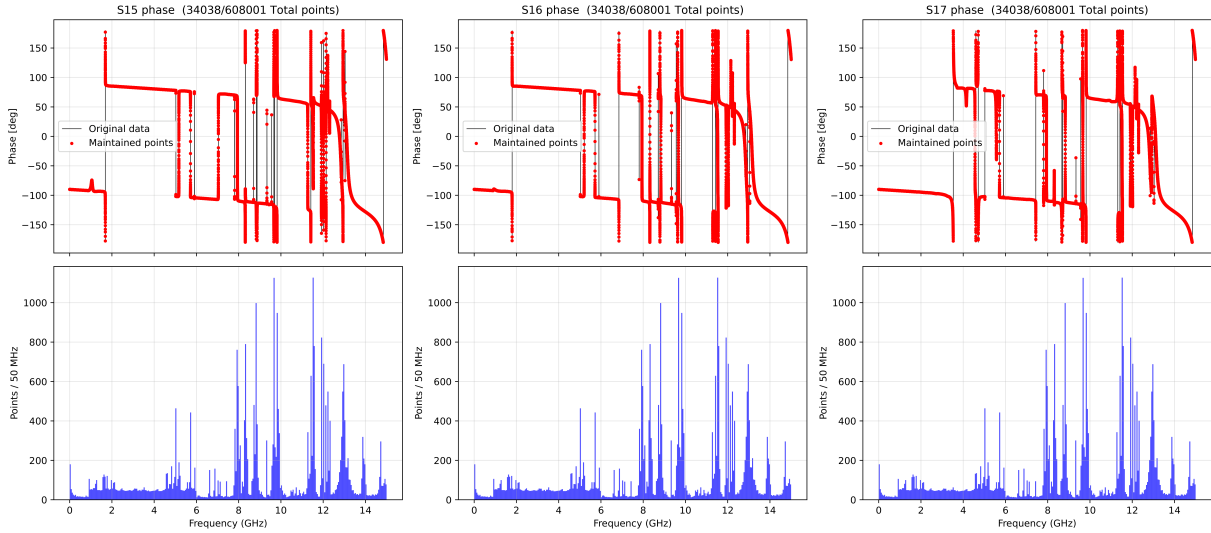


Figure 3.4. Effect of the filter on the data points. The top row shows the kept points in red superposed over the original data and the second row depicts the points density over the frequency span

as that option is not available otherwise. This choice is made as the curves on the diagonal s_{ii} have the highest absolute value, but are the simplest to fit; instead off diagonal curves s_{ij} and particularly curves like s_{16} reach much lower values with more complex shapes. Making this variation led to a fit that could better match those off-diagonal parameters, while keeping a low error on those simpler ones. This is also due to the fact that the weighting in this algorithm seems to be used only for the pole relocation process after which residues of each curve are re-fitted afterwards with no weight. As a result the curve with the highest RMS error will be s_{55} with $e_{RMS,55} \approx 6.7 \times 10^{-6}$ but since this curve always has a magnitude $|S| \geq 0.1$ it still remains one of the best fitted.

3.3.3 Fit execution and order of the model

The final choice has been to use `vf.vector_fit`, which maintains a fixed number of poles and only relocates them, instead of the alternative `vf.auto_fit` which, starting from a low number of poles, modifies it following the error [11]. This is a result of some testing in which the first one reached better results while also taking less time, and also gave a direct control over the model order.

```

1 # ...
2 vf.vector_fit(
3     n_poles_real=1,
4     n_poles_cmplx=35,
5     parameter_type='S',
6     init_pole_spacing='lin'
7 )

```

Table 3.1. Convergence of the vector-fitting RMS error with the model order (one real pole, varying number of complex-conjugate pole pairs n_{cmplx}).

n_{cmplx}	Order	RMS error	Relative gain	Gain per pole
30	61	$\sim 2.10 \times 10^{-5}$	—	—
35	71	$\sim 1.83 \times 10^{-5}$	+12.9% (from 61)	$\sim 1.3\%$
40	81	$\sim 1.69 \times 10^{-5}$	+7.7% (from 71)	$\sim 0.77\%$
50	101	$\sim 1.65 \times 10^{-5}$	+2.5% (from 81)	$\sim 0.13\%$

8 # ...

Listing 3.3. Vector fitting execution

Many different model orders with a varying combination of real and complex poles were tested, finding a first sweet spot at `n_poles_real=2` and `n_poles_cmplx=35` which took about 35 min to converge with a global RMS¹ error of $e_{RMS} \approx 1.8 \times 10^{-5}$. This was thought of as a good sweet spot as it gave a fast fit, with a small enough global error, which resulted in fast LTspice simulations with a coarse time step. But as soon as a realistically small time step was implemented, to obtain meaningful data from transient simulations, all of them started crashing. The culprit being the real poles, as also mentioned in the skrf manual, an even number of real poles may be found behaving similarly to complex conjugate poles, making the transient simulations unstable. None of the operations tested at a SPICE level were able to fix the problem, thus the choice of changing the poles to `n_poles_real=1`, which led to an even faster fit taking only 20 min and an $e_{RMS} \approx 1.83 \times 10^{-5}$, but more importantly simulations could converge with a time step of 1 ps with no need for a global shunted capacitance.²

Table 3.1 depicts the effects of changing order in the RMS error, as it can be seen after the chosen model order of 71, the gain of increasing n_{complex} starts to drop, not only that but SPICE simulations take significantly more time, leading to the choice of sticking with 71. If longer simulation times can be accepted one may decide to go further, for example to 81 – 86 but values higher seem to start following noise thus may also require a finer mesh upstream. Lastly, the initial pole spacing didn't seem, in the tests done, to make any difference to the final fit result and thus has been kept as linear.

3.3.4 Diagnostics

A cell has been added to check poles positioning as well as their Q factor and two interesting aspects arise. Regarding the positioning, three poles fall out of the simulated data, the real pole

¹Here the global RMS is calculated as:

$$e_{RMS} = \sqrt{\sum e_{RMS,ij}^2}$$

²Solution proposed by the LTwiki for the errors of the kind "time step too small"

as expected, and two poles at 35.742 GHz and 26.450 GHz, both with a very low Q factor of 0.89 and 5.93 respectively. As Ansys never simulated over 15 GHz this can not be a directly measured feature. But regardless of the starting pole placement, or even order of the model, these stayed almost unchanged, suggesting they are a useful feature of the fit resulting from the pole relocation process. The second detail that emerged is 6 poles with a Q factor in the order of 2×10^6 , four of these poles are at a frequency higher than 7.8 GHz which is outside the found qubit zone of 4.37 GHz – 6.78 GHz (will be discussed later) and are therefore not associated with the qubits. For the poles at 5.026 GHz and 5.903 GHz a further investigation must be done afterwards, checking their anharmonicity on how to correctly interpret their behavior. A second diagnostic quantifies the stiffness of the fitted pole set, defined as the ratio τ_{max}/τ_{min} , originally thought to be the reason for the failing transient simulations as well as the long time those took before changing order from 72 to 71. Also this feature stayed relatively constant with the model order, keeping a value of $\sim 1 \times 10^8$, which is 8 decades of stiffness. This was later found to not be the real culprit, since as discussed in §3.3.3, the real problem was the two real poles. Finally, the passivity of the fit was checked, since as already mentioned this quality is not guaranteed differently from the stability.

```

1  # ...
2  passivo = vf.is_passive(parameter_type='s')
3  print(f"Model is passive: {passivo}")
4
5  if not passivo:
6      bande = vf.passivity_test(parameter_type='s')
7      print("Passivity violation bands (Hz):")
8      for f_start, f_stop in bande:
9          f_stop_str = f"{f_stop/1e9:.4f} GHz" if np.isfinite(f_stop)
          else "infinite"
10         print(f"    {f_start/1e9:.4f} GHz - {f_stop_str}")
11 else:
12     print("No violation: model is passive.")
13 # ...

```

Listing 3.4. Lightweight passivity check, additionally giving information on which frequency bands the model violates the passivity condition

The output of this cell shows that the model is non passive, in most of the frequency span of the fit, but as of now no information on the severity is given.

```

1  # ...
2  A, B, C, D, E = vf._get_ABCDE()
3  freqs_check = rete_filtrata.f # check on the points given to the fit
4  _, sigma, _ = np.linalg.svd(vf._get_s_from_ABCDE(freqs_check, A, B,
5  C, D, E))

```

```

6 sigma_max_per_punto = sigma.max(axis=1)
7 idx_peggiore = np.argmax(sigma_max_per_punto)
8 # ...

```

Listing 3.5. Severity of the passivity violation using singular value decomposition

Here a clear point emerges, also shown in figure 3.5, the worst violation is $\bar{\sigma}_{max} = 1.0000853$ at 10.5185 GHz (about $+7.4 \times 10^{-4}$ dB), while for most of the band the overshoot is a full order of magnitude less than the found peak. Putting together the fact that the violation is distributed in the whole band with its magnitude, this seems to be the result of a numerical truncation introduced by the least-squares fit rather than a meaningful gain. `skrf` provides a routine, `passivity_enforce` [11] to correct the fit by perturbing the residues by the smallest amount needed to bring every singular value back under one. The routine mentioned has two problems, the first being how memory intensive it can become if a meaningful number of samples is given as input to force passivity. But, more importantly, even the reference method proposing this synthesis explicitly acknowledges [12] that passivity enforcement does not always converge and even if it does, the fitted model may still violate passivity requirements afterwards. This is exactly what happened, this exact routine was run in some previous fit always resulting in a model that still presented a non passive behavior, thus the choice to keep the model as it is, which has the consequence of making equation 2.6 unreliable.

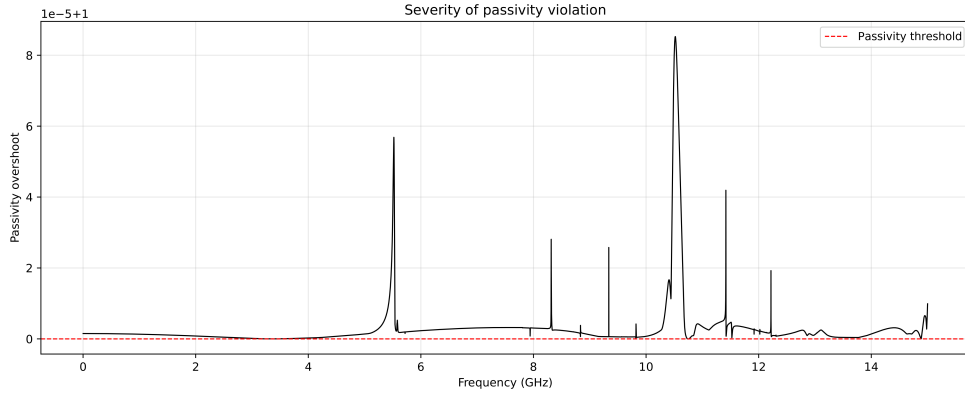


Figure 3.5. Plot output of the passivity severity test, as it can be observed, only a couple of spikes reach values up to 1.000085 while in most of the band it stays much lower and closer to a numerical noise floor.

3.3.5 Final validation

Having discussed the choices that led to the chosen fit a final validation may be done. The worst-performing region is a cluster of curves around 10.4–10.8 GHz, with an absolute RMS of about 1×10^{-5} , still below -90 dB in relative terms. Regarding the weak feed-line-to-qubit coupling curves, which matter most for the rest of this work, all fall in a range between 9.2×10^{-8} and 5.1×10^{-7} . Particular attention must be given to curves S_{17} and S_{27} , reported in figures 3.6 and

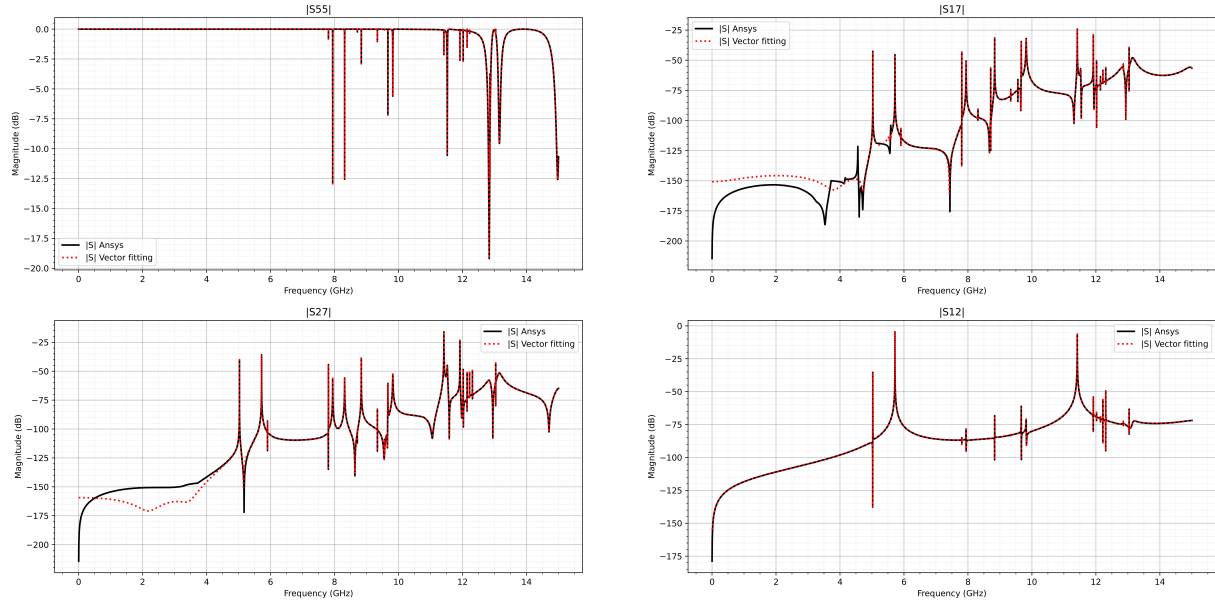


Figure 3.6. Selected curves comparing Ansys magnitude with the obtained fit. Most of the curves follow perfectly the original data, here S_{17} and S_{27} are reported as being the worst at matching the lower frequency spectrum, which most probably is due to the values being so low to reach numerical noise floor. Additionally, S_{55} is shown as it was the diagonal term with the worst RMS error to give a visual proof of the fact that its fit is clearly acceptable, taking into considerations its value range. Lastly S_{12} is the mutual scattering response between input and output of the feedline.

3.7, as even though RMS error is small, also is the value of the curves, and as it is clear from the images at lower frequencies the fit cannot match the original Ansys curves. Clearly, considering that their lower bound falls within -150 dB to -200 dB, and their upper limit is around -25 dB this results in a difference of up to ~ -175 dB, which is close to 9 orders of magnitude of difference. Making these considerations the mismatch shown is deemed acceptable, but it is a point in which improvements could be made.

3.3.6 SPICE subcircuit

Once the final fit has been accepted what is left to do is write it in a form that LTspice can accept and read, this can be done through `vf.write_spice_subcircuit_s` [14]. It is important to note that the option of creating reference pins for each port has been left to *False*, the reason being that this way the component created is internally referenced to the main ground used in LTspice, which is exactly what would have been done anyway, therefore there was no reason to give access to the reference pin of every port. If there is the need to reference some ports to a different value, then here one must set `vf.write_spice_subcircuit_s(..., create_reference_pins = True)`. The equivalent circuit obtained this way is what the rest of this thesis builds on, as all the following chapters either use this to directly implement BBQ formulas analytically or through transient simulations to cross check the results.

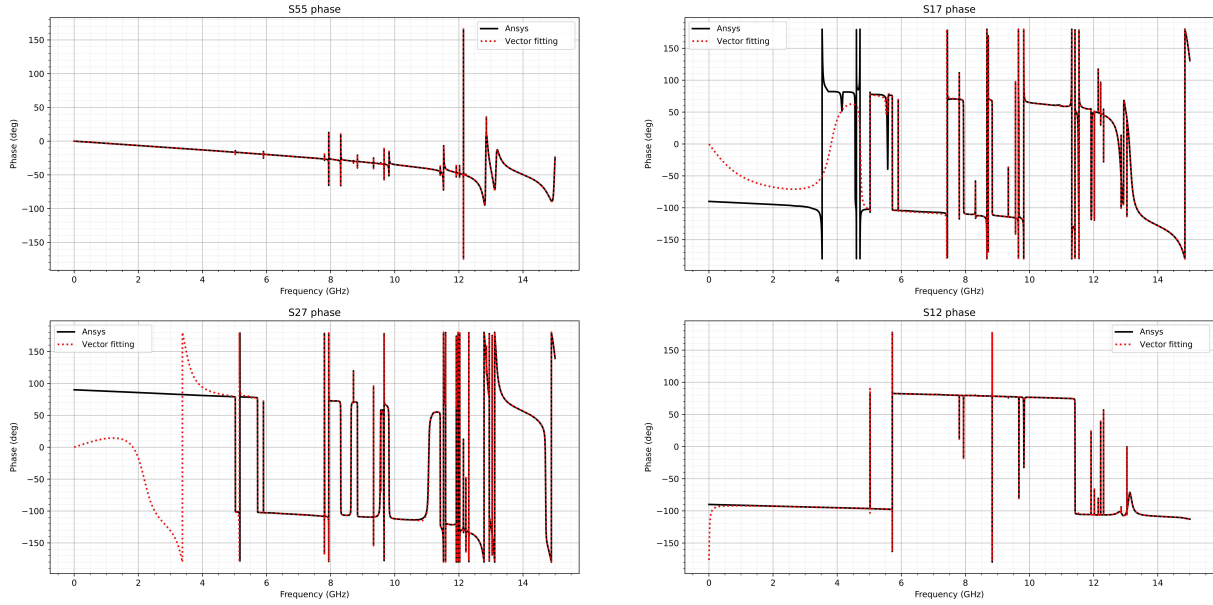


Figure 3.7. Selected curves comparing Ansys phase with the obtained fit. Most of the curves follow perfectly the original data, here S_{17} and S_{27} are again the worst curves, but at such low values this can be expected. Using again, S_{55} phase as a representative of diagonal terms fit, it is clear that also its phase has an almost perfect match. Lastly, S_{12} represents how, also for the mutual coupling between input and output of the feed line, at the lowest frequencies the fit loses track of the phase as it too reached pretty low values in magnitude.

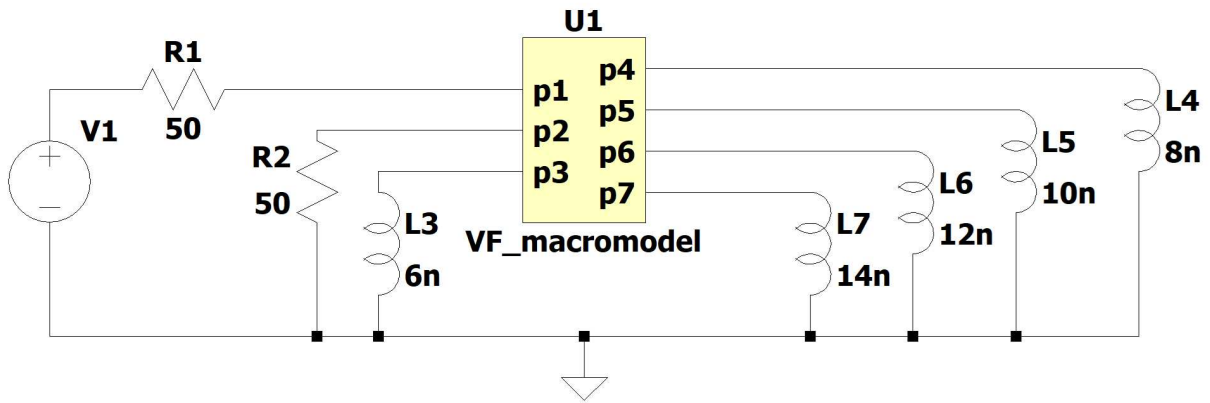


Figure 3.8. Example of a linear netlist, in yellow the 7-port subcircuit built from the vector fitting algorithm, connected to linear inductors at the qubit ports, $50\ \Omega$ resistors both at the input and output of the feed line, with the input also having a series connected voltage source. It is important to note that all components are referenced to the same main ground to which also the VF macromodel is connected internally.

Chapter 4

Black-box quantization: analytical implementation

The previous chapter ended by producing a fit (order 71) and a 7-port LTspice netlist constructed from it, both containing only the information of the electromagnetic environment, these two files will be used throughout all the following chapters. Here the formalism presented in §2.3 is applied, and not the exact-synthesis alternative discussed in §2.3.2. The logical steps followed are:

1. Reduction of the fitted network to 5-ports, by closing the feed line terminations to $50\ \Omega$ loads
2. Locating the normal modes of the system
3. For each mode found, calculating the participation ratio and effective impedance
4. Compute the resulting self- and cross-Kerr coefficients
5. Estimate the quality factor of each mode and the corresponding Purcell limit

Lastly, the spectral features found above to be within the qubit range (5.026 GHz and 5.903 GHz) are addressed, confirming they are not junction related but modes of the bare EM environment.

4.1 Preliminary single-port frequency estimate

Before applying the multiport BBQ treatment, an approximated pass to locate each qubit's frequency is done using the self-admittance of each qubit port, i.e. with all the other ports, feed line included, short-circuited. For each qubit k , the self-admittance of its own port, $Y_{env,k}(\omega) = Y_{kk}(\omega)$, is read directly out of the fitted $\mathbf{Y}$ -matrix and loaded with that qubit's own linear inductance, $Y_{tot,k}(\omega) = Y_{env,k}(\omega) + 1/(j\omega L_j^{(k)})$. This process is done sequentially, loading only one junction at a time, the other junction ports being short-circuited, thus not capturing the dressing of having all JJs at once.

```
1 def trova_risonanza_qubit(freq_hz, Y_tot, banda=(4e9, 7e9)):  
2     mask = (freq_hz >= banda[0]) & (freq_hz <= banda[1])
```

Table 4.1. Results of the first, approximated, qubit frequency estimate.

Qubit	L_j (nH)	Resonance (GHz)	Recurring triplet found (GHz)
Q_0	6	~ 6.7815	5.0262, 5.7187, 5.9030
Q_1	8	~ 5.8725	5.0262, 5.7187, 5.9031
Q_2	10	~ 5.2630	5.0262, 5.7188, 5.9031
Q_3	12	~ 4.8198	5.0263, 5.7188, 5.9031
Q_4	14	~ 4.4767	5.0263, 5.7188, 5.9031

```

3     f_banda = freq_hz[mask]
4     Y_banda = Y_tot[mask]
5
6     idx_picchi, _ = find_peaks(1 / np.abs(Y_banda))
7     idx_zero = np.where(np.diff(np.sign(Y_banda.imag)) != 0)[0]
8
9     return f_banda[idx_picchi], f_banda[idx_zero]
```

Listing 4.1. Function to find peaks of Z and zeros of Y .

In the listing 4.1 is shown the function that allows to search for resonance frequencies, by checking both zeros of the reactance and peaks of the impedance, the first being the standard Foster criterion for a low-loss resonance, expected to coincide closely with the impedance-magnitude peak whenever losses are weak. This method is clearly incomplete, as the real circuit in which the qubits will be working has $50\ \Omega$ terminations at the feed line ends, thus modifying the matrix $\mathbf{Y}$ from which the environment is extracted. Secondly, here only one junction is uploaded at any time, this is also not what will really happen in the complete circuit, again changing the EM environment.

Figure 4.1 and table 4.1 show the found results of this process, and an interesting observation can be made. If these findings will be compared to the ones presented in the following sections, one may notice that the found frequencies here are a very good match of the real ones. This is a result of two facts:

- The feed line is weakly coupled to the qubits, making it so that changes made to its connections don't strongly affect the qubit behavior
- Once the participation ratios are calculated in the following, it will become clear that qubits don't participate strongly in each other's modes

The second point just mentioned, can fall short really easily by simply making the L_j of each qubit closer to each other, thus this method should be treated more as a guideline than a real estimate. But having this initial values, will be helpful in the real BBQ implementation to help in the choice of reference ports for each found mode.

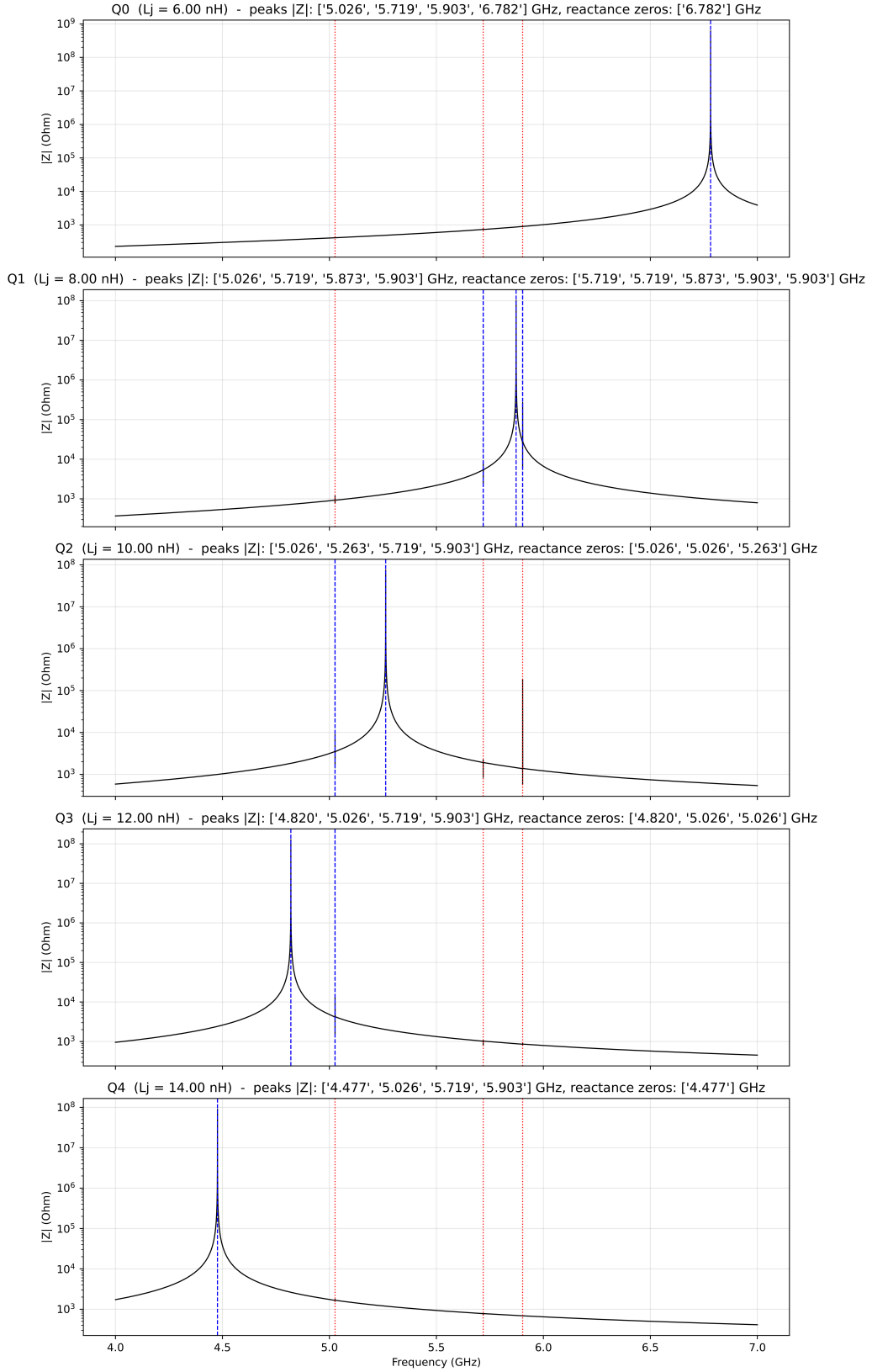


Figure 4.1. Output of the function `trova_risonanza_qubit`, showing the magnitude of the port impedance of each qubit, highlighting in blue the found reactance zeros and in red the impedance peaks.

4.2 Multiport mode search

Having done a first approximate estimate of the qubits' frequencies, it is now time to move forward with a more precise evaluation. For this reason, the fitted network obtained is now closed with $50\ \Omega$ resistors at ports 0 and 1, the ones associated with the feed line.

```

1 def rete_resistore(freq, R, z0=50):
2
3     Z = np.full(freq.npoints, R, dtype=complex)
4     gamma = (Z - z0) / (Z + z0)
5     return rf.Network(frequency=freq, s=gamma.reshape(-1, 1, 1), z0=
6                        z0)
7
8 carico_readout_out = rete_resistore(rete_fit.frequency, 50)
9 rete_JJ = rf.network.connect(rete_fit, 1, carico_readout_out, 0)
10
11 carico_readout_in = rete_resistore(rete_JJ.frequency, 50)
12 rete_JJ = rf.network.connect(rete_JJ, 0, carico_readout_in, 0)

```

Listing 4.2. Starting from the vector-fitted network obtained before, `rete_fit`, this code transforms it into a 5-port network closing ports 0 and 1 on $50\ \Omega$ resistors.

Listing 4.2 shows the code applied to the fitted object, `rete_fit`, to close its first two ports as above mentioned, resulting in a 5-port network, where each port is associated with a qubit. These terminations will be the main dissipation mechanism of the system, as mentioned also by Solgun, Abraham and DiVincenzo while working in a similar test case in [12]. The next logical step is to insert, at each of the remaining ports, its inductance as a shunted component, building the loaded admittance matrix

$$\mathbf{Y}_{tot}(\omega) = \mathbf{Y}_{bare}(\omega) + \begin{bmatrix} \frac{1}{j\omega L_j^{(0)}} & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{j\omega L_j^{(1)}} & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{j\omega L_j^{(2)}} & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{j\omega L_j^{(3)}} & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{j\omega L_j^{(4)}} \end{bmatrix} \quad (4.1)$$

where $\mathbf{Y}_{bare}$ is the 5-port admittance matrix of the network obtained via listing 4.2. Normal modes of the loaded system are located at the zeros of $\det \mathbf{Y}_{tot}$

$$\det \mathbf{Y}_{tot}(\omega) = 0 \quad (4.2)$$

Table 4.2. Eight candidate zeros coming from equation 4.2. Only those with a prominence greater than 3 decades will be kept.

Mode	f_{mode} (GHz)	Prominence (decades)	Kept or not
0	4.4739	~ 4.18	Kept
1	4.8218	~ 4.33	Kept
2	5.2633	~ 5.73	Kept
3	5.8723	~ 4.36	Kept
4	6.7828	~ 4.86	Kept
-	5.9031	~ 1.55	Discarded
-	5.7187	~ 0.03	Discarded
-	5.0262	~ 0.58	Discarded

which, since $\mathbf{Z}_{tot} = \mathbf{Y}_{tot}^{-1}$, coincide with the peaks of

$$|\mathbf{Z}_{00}(\omega)| \quad (4.3)$$

the impedance seen from port 0 (Q_0), for every mode with a non-negligible participation at that port. By applying both methods in the range 4-7 GHz, the same eight candidate zeros appear. These modes are then passed to a filter, which, considering their prominence, discards the ones that don't have one of at least 3 decades. The results are reported in table 4.2.

```

1  soglia_prominenza_decadi = 3.0
2
3  log_det_banda = np.log10(np.abs(det_banda))
4  idx_candidati, proprieta = find_peaks(-log_det_banda, prominence=0)
5  prominenze = proprieta['prominences']
6
7  tenuti = prominenze >= soglia_prominenza_decadi
8  idx_tenuti = idx_candidati[tenuti]
9  idx_scartati = idx_candidati[~tenuti]
10
11 f_modi_filtrati = f_banda[idx_tenuti]
12 f_modi_scartati = f_banda[idx_scartati]
13 # ...

```

Listing 4.3. The prominence filter used to discard non qubit-like modes found from equation 4.2. Notice that the `find_peaks` function has been imported from the `scipy.signal` library

The filter was applied with a minimum prominence of 3 decades, but one can observe that all the kept modes stay above 4; the remaining three will be discussed later in this chapter where they will be shown to be cavity-like modes of the bare EM environment.

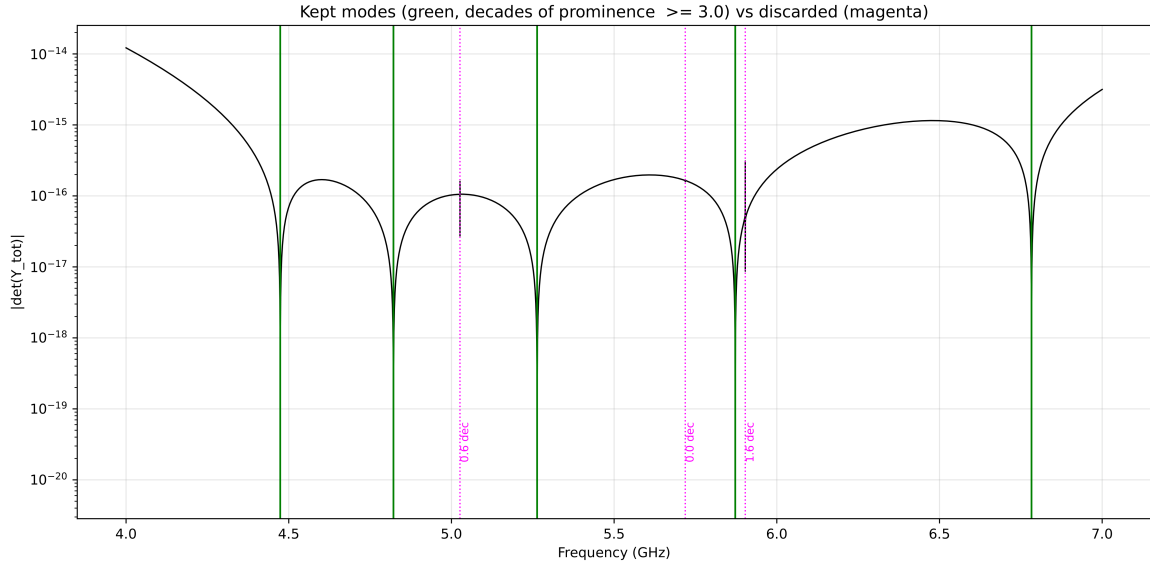


Figure 4.2. Found modes via equation 4.2, highlighting in green the kept, qubit-like modes and in magenta the discarded modes.

4.3 Qubit parameters calculation

4.3.1 Participation ratios and effective impedance

Having found the five physical modes, the quantization procedure proposed in §2.3 requires, for each of them, to calculate the effective impedance Z_p^{eff} and the participation ratio ξ_{sp} . Both will be extracted from the same loaded impedance matrix $\mathbf{Z}_{tot} = \mathbf{Y}_{tot}^{-1}$ evaluated at ω_p , by applying the formulas 2.8 and 2.9 respectively. Regarding exactly those equations, the original paper [2] explicitly states that while it is required to fix a reference port, mode frequencies ω_p do not depend on that choice, while the individual participation ratios do, thus a single reference port $k = 1$ was fixed for every mode. In the following, instead, for each mode a reference port has been chosen independently, selecting for each the one whose associated qubit's frequency, already estimated in Table 4.1, is the closest to ω_p . This is not a physical deviation from the formalism: the Hamiltonian coefficients ultimately built from ξ_{sp} (the $\beta_{qq'pp'}$ shown in equation 2.10) are, by construction, independent of which port is used as reference, since they are computed from the same underlying physical matrix $\mathbf{Z}(\omega)$ regardless of how it is normalized in the intermediate step (it being the ξ_{sp} coefficients). Motivating this choice is a purely numerical issue. It has been seen that keeping the same reference would result in dividing, for at least some of the modes, by a quantity $Z_{kk}(\omega_p)$ in which that specific mode has almost no participation. At that regime, the signal is comparable to the numerical floor noise of the fitted network (the same non-passivity floor already characterized in §3.3.4), which resulted in meaningless results like 'nan' both for Z_p^{eff} and ξ_{sp} in every mode except the one closest to that port's own qubit frequency. From here comes the choice of selecting ad hoc, thanks to the preliminary estimation, the reference port for each mode, thus completely avoiding this ill-conditioning.

Table 4.3. Modes and port matching results, highlighting the distance from quantities shown in Table 4.1.

Mode	f_{mode} (GHz)	Reference port (Qubit)	Distance from preliminary estimate (MHz)
0	4.4739	Q_4	2.867
1	4.8218	Q_3	2.021
2	5.2633	Q_2	0.352
3	5.8723	Q_1	0.211
4	6.7828	Q_0	1.316

Table 4.4. Resulting effective impedance and participation ratios, with the dominant qubit for each mode highlighted in bold.

Mode	f_{mode} (GHz)	Z_p^{eff} (Ω)	Q_0	Q_1	Q_2	Q_3	Q_4
0	4.4739	385.54	0.000	0.000	0.000	0.008	0.992
1	4.8218	351.85	0.000	0.000	0.002	0.988	0.010
2	5.2633	320.24	0.000	0.001	0.997	0.002	0.000
3	5.8723	284.62	0.001	0.998	0.001	0.000	0.000
4	6.7828	247.78	0.999	0.001	0.000	0.000	0.000

```

1 for p in range(M):
2     idx = np.argmin(np.abs(omega - omega_p[p]))
3
4     dist_min, k = np.inf, 0
5     for kk in range(5):
6         candidat_i = np.concatenate([freq_singole[kk]['zeri'],
7         freq_singole[kk]['picchi']])
8         if len(candidat_i) == 0:
9             continue
10        d = np.min(np.abs(candidat_i - f_modi_ord[p]))
11        if d < dist_min:
12            dist_min, k = d, kk
13        k_rif_mod0[p] = k
14        dist_match[p] = dist_min
15 # ...

```

Listing 4.4. Assignment of a reference port to each mode, by nearest match to the preliminary single-port resonances of Table 4.1.

The matching turns out to be unambiguous, with every mode being found to be within a few MHz from the preliminary estimate. This may not always be the case, but if a mode is found at a comparable distance from two different port estimations, it will probably mean that both junctions have a meaningful participation there, in which case either choice remains well-conditioned — though for a strongly hybridized mode neither preliminary estimate may track ω_p as closely as in the well-separated case seen here. Finally both the effective impedance and the participation ratios can be computed as in equations 2.8 and 2.9 resulting in the values shown in table 4.4. From the data obtained, it is evident that each mode is concentrated mostly in a single junction

(minimum single concentration 98.8%), resulting in a set of five essentially independent qubits. This sharp separation is what makes the preliminary estimation, done in §4.1, so close to the more precise estimation here carried out. This same near-diagonal structure also anticipates a feature of the cross-Kerr matrix computed next (Table 4.6): every off-diagonal χ_{qp} turns out to be smaller than the self-Kerr values of Table 4.5 by about 30 times for the closest pair of qubits, down to almost five orders of magnitude for the most distant ones.

4.3.2 Self-Kerr and cross-Kerr coefficients

Having calculated Z_p^{eff} and ξ_{sp} for every mode, it is now time to reintroduce the non linearity of the system as a perturbation to the linear Hamiltonian, using a quartic approximation of the cosine term as mentioned in §2.3. The leading correction produces the self-Kerr α_p (anharmonicity) and cross-Kerr χ_{qp} through equations 2.11 and 2.12. Both of which, to be calculated, make use of the term $\beta_{qq'pp'}$ derived from equation 2.10¹, from which also the Lamb shift Δ_p is derived as in equation 2.13.

```

1 import scipy.constants as spc
2 h_planck = spc.h
3
4 def beta_qqpp(xi, Lj_list, q, p):
5     return sum((spc.e ** 2 / (24 * Lj_list[s])) * xi[s, q] ** 2 * xi
6               [s, p] ** 2 for s in range(len(Lj_list)))
7
8 alpha_J = np.array([-12 * beta_qqpp(xi, Lj, p, p) for p in range(M)
9                    ]) # Joule
10
11 chi_J = np.array([[-24 * beta_qqpp(xi, Lj, q, p) if q != p else
12                   0.0] for p in range(M)] for q in range(M)) # Joule
13
14 Delta_J = np.array([6 * beta_qqpp(xi, Lj, p, p) - 12 * sum(beta_qqpp(
15   xi, Lj, q, p) for q in range(M)) for p in range(M)]) # Joule
16
17 f01 = omega_p / (2 * np.pi) + Delta_J / h_planck # From J to Hz,
18   Lamb shift
19
20 anarmonicita = alpha_J / h_planck # From J to Hz, anharmonicity (
21   self-Kerr)
22
23 chi = chi_J / h_planck # From J to Hz, cross-Kerr
24 # ...

```

Listing 4.5. Code used to calculate anharmonicity, cross-Kerr and Lamb shift in (Hz).

The obtained results are summarized in tables 4.5 and 4.6. All five self-Kerr values fall in the -196 to -206 MHz range, which is aligned with ranges found in literature [2]. As ex-

¹It is relevant to notice that the term $\beta_{qq'pp'}$, as presented, has units of energy [J], while the parameters α_p , χ_{qp} and Δ_p are usually given in [Hz]. To stay consistent with the literature, here $\beta_{qq'pp'}/h$ [Hz] is used instead of $\beta_{qq'pp'}$ [J]

Table 4.5. Resulting transition frequency f_{01} and anharmonicity for each qubit, compared to the initially calculated mode frequency.

Qubit	Mode	f_{mode} (GHz)	$f_{01} = f_{mode} + \Delta_p/h$ (GHz)	α_p (MHz)
Q_4	0	4.4739	4.3674	-205.67
Q_3	1	4.8218	4.7172	-199.86
Q_2	2	5.2633	5.1626	-198.65
Q_1	3	5.8723	5.7734	-196.15
Q_0	4	6.7828	6.6833	-198.21

Table 4.6. Calculated cross-Kerr between each mode, the diagonal is not reported, the self-Kerr α_p being given in Table 4.5. Additionally the lower triangular part is not reported being the matrix perfectly symmetrical.

χ_{qp}	Mode 0	Mode 1	Mode 2	Mode 3	Mode 4
Mode 0	–	-7.2149	-0.0386	-0.0167	-0.0027
Mode 1		–	-1.9461	-0.1809	-0.0277
Mode 2			–	-0.6301	-0.0861
Mode 3				–	-0.8096
Mode 4					–

pected, the two closest qubits, Q_3 and Q_4 , whose associated modes differ by about 350 MHz, have the strongest cross-Kerr entry at -7.2149 MHz, consistent with a capacitively mediated interaction that grows with proximity in frequency. Lastly, a consistency check can be made, as mentioned in [2], the Cauchy–Schwarz inequality applied to $\beta_{qq'pp'}$ implies $|\chi_{qp}| \leq 2\sqrt{\alpha_q\alpha_p}$. Checking this against the largest cross-Kerr entry just mentioned results in $7.2149 \text{ MHz} \leq 2 \times \sqrt{205.67 \times 199.86} \approx 405.5 \text{ MHz}$, comfortably satisfied, as expected from a weakly coupled system like this one.

4.4 Quality factor and Purcell limit

Once all modes' frequencies, effective impedances and Kerr coefficients are established, this leaves only two physical quantities to be extracted: the radiative quality factor Q_p and the resulting Purcell-limited T_1 . Those, in §2.3, are calculated via equations 2.6 and 2.7, making use of $\text{Re}Y(\omega_p)$ which represents the residual conductance of the loaded port at resonance, i.e. the radiative loss channel through which energy leaks from the qubit toward the $50\ \Omega$ readout terminations connected to the feed line. For a well isolated qubit, this quantity is expected to be particularly small, but still positive. What happens here instead is that, since the fit did not force the passivity condition, as mentioned in §3.3.4, this residual conductance takes a *negative* sign, since, for such a well isolated qubit, the true radiative $\text{Re}Y(\omega_p)$ is smaller, in absolute value, than the parasitic noise floor due to the network not being passive. This failure mode is exactly what is presented in §2.3.2, proposing the alternative path of Brune's circuit to avoid it. In order to still have a quantitative estimate of the quality factor and time T_1 , here the formula 2.6 is only used if $\text{Re}Y(\omega_p)$ clears the noise floor by a safety factor ($K_{\text{floor}} = 3$). If this condition is not met,

then a positive lower bound is set following two distinct logic paths:

1. **Passivity floor:** The fit introduces a spurious conductance up to ε_g ; any value lower than this (in modulus) cannot be distinguished from the error introduced by the fit itself (the same non-passivity already quantified, via a different diagnostic, in figure 3.5). Therefore, $\text{Re}Y(\omega_p)$ is substituted with the floor value $K_{\text{floor}} \varepsilon_g$, giving in output a worst case estimate of Q_p .
2. **Grid resolution:** Valid also for a perfectly passive fit with $\varepsilon_g = 0$, for any resonance the quality factor can be expressed as $Q_p = \omega_p / \Delta\omega$. Thus, to be able to see that it has a finite width (typically expressed as the Full Width at Half Maximum, FWHM), at least a couple of points are needed inside its FWHM range. Fixing the minimum solvable width to be $\sim 2d\omega$ (the local grid pitch) results in $Q_p \geq \omega_p / (2d\omega)$, since a resonance sharper than $2d\omega$ cannot be properly captured by the used grid.

Each of these gives an independent lower bound on Q_p — Q_{lb}^{pass} from the passivity floor, $Q_{lb}^{\text{grid}} = \omega_p / (2d\omega)$ from the grid resolution — obtained via two different formulas, not by comparing the two floors directly (which, being a conductance and a frequency, are not even in the same units). Finally, taking the maximum of these two resulting lower bounds on Q_p gives the tightest lower bound on Q_p , and consequently T_1 , that can still be defended. This whole idea is implemented via the following listing 4.6.

```

1  _fJJ  = omega / (2 * np.pi)
2  _mb   = (_fJJ >= banda_ricerca[0]) & (_fJJ <= banda_ricerca[1])
3  _Hn   = 0.5 * (Y_bare[_mb] + np.conj(np.transpose(Y_bare[_mb], (0,
    2, 1))))
4  eps_g = float(max(0.0, -np.linalg.eigvalsh(_Hn)[: , 0].min()))
5  _Yscale = float(np.median(np.abs(Y_bare[_mb][:, np.arange(5), np.
    arange(5)])))
6
7  K_FLOOR = 3.0    # Safety factor
8  ReY_floor = K_FLOOR * eps_g
9
10 # ...
11
12 for p in range(M):
13     k = int(k_rif_mod0[p])
14     idx = int(np.argmin(np.abs(omega - omega_p[p])))
15     Y_k = 1.0 / Z_tot[:, k, k]    # self port k admittance (eq. 12
    BBQ paper)
16     dY_k = np.gradient(Y_k, omega)
17     ReY_p[p] = float(np.real(Y_k[idx]))
18     ImYd_p[p] = float(np.imag(dY_k[idx]))    # = 2/(omega_p*Zeff), well
    conditioned
19
20     d_omega = float(omega[idx + 1] - omega[idx])    # local step of
    grid
21     Q_lb_grid = omega_p[p] / (2.0 * d_omega)

```

Table 4.7. Quality factor and T_1 found. All of them, due to $\text{Re}Y(\omega_p)$ falling below the noise threshold, are a lower-bound estimate obtained via the described procedure of listing 4.6.

Mode	f_{01} (GHz)	$\text{Re}Y(\omega_p)$ (S)	Status	Q_p	T_1 (μs)
0	4.3674	-3.51×10^{-11}	below threshold	$\geq 9.5 \times 10^4$	≥ 3.4
1	4.7172	-7.45×10^{-9}	below threshold	$\geq 1.0 \times 10^5$	≥ 3.4
2	5.1626	-1.81×10^{-9}	below threshold	$\geq 1.1 \times 10^5$	≥ 3.4
3	5.7734	-5.74×10^{-10}	below threshold	$\geq 1.2 \times 10^5$	≥ 3.4
4	6.6833	-1.28×10^{-9}	below threshold	$\geq 1.4 \times 10^5$	≥ 3.4

```

22     if eps_g > 0:
23         Q_lb_pass = (omega_p[p] / 2.0) * ImYd_p[p] / ReY_floor
24         Q_p_lb[p] = max(Q_lb_pass, Q_lb_grid) # Keep most stringent
           test
25     else:
26         Q_p_lb[p] = Q_lb_grid
27     T1_lb[p] = Q_p_lb[p] / omega_p[p]
28
29     if ReY_p[p] > ReY_floor:          # Re Y(omega_p) above noise
30         risolto[p] = True
31         Q_p[p] = (omega_p[p] / 2.0) * ImYd_p[p] / ReY_p[p]
32         T1_purcell[p] = Q_p[p] / omega_p[p]
33
34 # ...

```

Listing 4.6. Code used to calculate the noise floors due to passivity and frequency grid pitch, taking the maximum of the two resulting lower bounds on Q_p to apply equation 2.6 only when $\text{Re}Y(\omega_p)$ is greater than the floor.

The found passivity floor is $3\varepsilon_g \approx 2.22 \times 10^{-7}$ S, whereas the local grid pitch is ~ 23.5 kHz, resulting in all modes falling below the threshold as reported in table 4.7. Possible paths to avoid this clear limitation may be:

- Maintaining this method, but forcing the passivity and using a finer frequency sampling
- Moving to the method proposed by Solgun, Abraham and DiVincenzo [12], [13], which is still not immune to the frequency grid pitch issue

4.5 Cavity-like modes

Recalling table 4.2 as well as table 4.1, it is time to explain why the frequencies 5.0262 GHz, 5.7187 GHz² and 5.9031 GHz are not considered to be qubit modes.

²The candidate 5.7187 GHz was not talked about much in the previous sections as it had a much lower Q factor than the other two, $Q \approx 44233$, data coming from pole inspection of the fit, but since the evidence hereafter described applies also to this candidate it makes sense to talk about it.

Table 4.8. Effective impedance and self-Kerr of the discarded modes.

Candidate (GHz)	$Z^{eff}(\Omega)$	Anharmonicity (MHz)	Largest χ with a qubit mode (MHz)
5.9031	0.42	≈ 0	≈ -0.57
5.7187	15.01	≈ -0.55	≈ -20.68
5.0262	-10.62	nan	nan

Evidence 1 In the single-port scan, carried out in §4.1, each qubit port is loaded with a different linear inductance L_j . This triplet, as shown in table 4.1, keeps occurring, almost unchanged, in every one of the five scans. If these modes were really dependent on the junction’s inductance, their frequency should have been affected by the varying L_j (spanning from 6 nH to 14 nH); this clearly doesn’t happen. These frequencies may already be considered a feature of $\mathbf{Y}_{bare}$ alone, present before any junction is even introduced.

Evidence 2 Applying the same procedure of §4.3.2 to calculate the Kerr coefficients results in table 4.8. For 5.9031 GHz and 5.7187 GHz, Z^{eff} goes from three to one order of magnitude smaller than the 250–390 Ω found for the five qubit-like modes (Table 4.4). Additionally, the anharmonicity is basically zero for 5.9031 GHz and three orders of magnitude lower than ~ -200 MHz (Table 4.5) for 5.7187 GHz; particularly, an anharmonicity almost null is exactly the behavior expected from a cavity-like mode [2]. Despite this small self-Kerr, the largest cross-Kerr entry of 5.7187 GHz with a qubit mode, $\chi \approx -20.68$ MHz (Table 4.8), comes close to saturating the Cauchy-Schwarz bound $|\chi_{qp}| \approx 20.68 \leq 2\sqrt{\alpha_q \alpha_p} = 2 \times \sqrt{0.55 \times 200} \approx 21.0$ already used in §4.3.2: with $\alpha \approx -0.55$ MHz for this mode and ≈ -200 MHz for the qubit mode it pairs with, the bound is less than 2% above the value found. Despite its own tiny anharmonicity, this mode channels nearly all of its available nonlinear coupling into interaction with that single nearby qubit. Regarding 5.0262 GHz, Z^{eff} comes out negative, making ξ_{sp} and the anharmonicity undefined; this phenomenon is still informative and addressed in the next evidence.

Evidence 3 Independently cross-checking the poles of the fit at two different orders, 71 (the one used throughout this work) and 101 (from a previous test), gives quality factors agreeing from 5 up to 6 significant digits for all three candidates. Noticing that independent fits agree to such a degree in both the position and shape of these poles suggests that these are genuine features of the HFSS data. For candidate 5.0262 GHz, which has the second highest $Q \approx 3163188$, the resonance linewidth is $\Delta f = f_{candidate}/Q \approx 1.6$ kHz, already much lower than the frequency grid pitch of ~ 23.5 kHz. This makes 5.0262 GHz, as well as 5.9031 GHz (with $Q \approx 5119679$), impossible to realistically resolve, particularly for a derivative with respect to frequency like the one used in $\text{Im}Y'(\omega_p)$ for the calculation of Z^{eff} . The reason why the candidate 5.0262 GHz gives a negative effective impedance, resulting in undefined results in table 4.8, and 5.9031 GHz does not, is simply that the single point representing it falls in a grid point in which the derivative

happens to be better conditioned. Regarding 5.7187 GHz, its much lower $Q \approx 44233$ gives $\Delta f \approx 130$ kHz, making it the only candidate resolvable with the given grid, which is reflected in the results of table 4.8.

All three pieces of evidence are mutually consistent and, taken together, leave little doubt that all candidates are modes of the bare electromagnetic environment rather than junction resonances. In this sense they are called cavity-like, with their precise physical origin being harder to pin down.

Chapter 5

Nonlinear model validation

The linear part of the circuit, studied in Chapter 3, is directly exported as a `.sp` file, LTspice's native subcircuit format. LTspice does not have a native implementation of the JJ, but custom elements can be constructed allowing for it to be built, whereas specialized simulators like JoSIM [18] already implement it through an RCSJ model (Resistively and Capacitively shunted junction). JoSIM's core simulation engine is actively maintained, but the Python layers built around it for automated, parametrized simulation are not. Both `pyjosim` [19], meant for scripting JoSIM from Python, and the higher-level `josim-tools` package [20], meant for parameter sweeps and margin analysis, have not been updated since 2020 and 2021 respectively. This work instead extensively relies on Python automation to programmatically generate netlists, launch parallel transient runs, and parse results directly into array form, through the `PyLTspice` package. Additionally, JoSIM's independent sources are limited to a fixed set of waveforms (PULSE, SIN, EXP, PWL, PWS, an external waveform file, DC, NOISE), and cannot use behavioral sources like LTspice. Therefore the custom envelopes used in this work, like a Gaussian pulse or other shaped drives for the cross-talk estimation, cannot be implemented as a symbolic expression, but would need to be precomputed pointwise in Python and imported as explicit PWL. This introduces a further problem: a piecewise-linear approximation of an otherwise smooth envelope has a discontinuous first derivative at each breakpoint, and this discontinuity produces a spectral tail that decays only as a power law, rather than the much faster decay of the original smooth waveform. This is exactly the kind of spectral leakage this work tries to avoid, for example by band-limiting the drive turn-on with $1 - e^{-t/\tau}$ instead of a sharp step, since an abrupt transition injects broadband energy that can excite every mode of the circuit. JoSIM is still a valid alternative, and should be considered regarding this kind of study, but in this work LTspice will be used.

5.1 JJ LTspice model

The first step, due to the choice of using LTspice, is to build a component that behaves like a JJ, which is done through a behavioral source. The initial try was to directly implement the system

of equations 1.7, which has been discovered to be an unsuccessful approach.

```
1 B_JJ top bottom I={ (Phi0/Lj) *sin(idt(V(top,bottom)) /Phi0) }
```

Listing 5.1. First implementation of a JJ model in LTspice using a behavioral source directly implementing Josephson’s relations.

The culprit here is the integral `idt()`, as it introduced a numerical damping artifact. Paired linear and nonlinear simulations were run together at a low amplitude drive, so that the phase φ would stay low enough to make $\sin \varphi \approx \varphi$, making the nonlinearity negligible, thus the circuits should match closely. Instead, the two systematically differed, with the nonlinear having a response 36 to 257 times smaller than the linear one, depending on the qubit. The most likely cause is a solver-level detail: LTspice resolves the circular dependency of a self-referential B-source through auxiliary internal elements, introducing spurious damping. Pushing this same model to high drive amplitude produces a different failure mode, the transient solver aborts with a “time step too small” error at the instant the junction’s phase reaches exactly $\varphi = \pi/2$, with $I_{JJ}/I_c = 1.00000$. The point at which the differential inductance $L_j/\cos \varphi$ diverges and then turns negative, which is genuine physical switching of the junction and not a numerical artifact. This crash behavior at high drive is a known feature of this class of solvers, as also JoSIM’s documentation of its trapezoidal-integration engine notes that “when a rapid change in y occurs such as a 2π Josephson phase switch, the derivative (voltage) tends to produce spikes” [18]. Other alternatives tested were:

- Cubic residual on a native inductor, $I = -(\Phi_0/L_j) * \text{pow}(\text{idt}(V) / \Phi_0, 3) / 6$: $\text{pow}(\text{idt}(V), 3)$ is unbounded, and the integrator’s natural drift made it diverge, resulting in early crashes
- Flux-defined inductor $\text{Flux} = \Phi_0 * \text{asin}(x * L_j / \Phi_0)$, avoiding `idt()` but still unsuccessful due to an intrinsic reactance/phase offset of LTspice’s `Flux=` construct itself

The successful model, which is used in the rest of the work, has been found to be the following

```
1 .subckt JJ_Xmon_Lcorr top bottom Lj=10nH Cj=0fF R_shunt=100G
2 .param Phi0 = 3.2910597847545335e-16
3 C_j      top bottom {Cj}
4 R_shunt  top bottom {R_shunt}
5 L_ref    top bottom {Lj}
6 B_n1     top bottom I={ (Phi0/Lj) *sin(I(L_ref)*Lj/Phi0) - I(L_ref) }
7 .ends
```

Listing 5.2. Working JJ implementation in LTspice.

The model’s dominant behavior is given by the linear inductance L_{ref} , LTspice’s own exact model, and a single B-source, with no integration involved, adds only the bounded non linear residual $I_c(\sin \varphi - \varphi)$, obtaining $I = I_{L,ref} + (I_c \sin(I_{L,ref} L_j / \Phi_0) - I_{L,ref})$, with the phase

directly linked to the linear inductance as $\varphi = I_{L,ref} L_j / \phi_0$. This implementation exploits the linear inductor to calculate the phase, directly through its natural equation $\Phi = Li$, using the link between magnetic flux and phase (it being the reduced flux quantum ϕ_0). Also, this model has limitations: due to L_{ref} having no current limit, φ can grow past $\pi/2$ without the model diverging. This is not a physical behavior, since, beyond that threshold, a real junction would switch, thus making the model valid only up to $\varphi \lesssim 1$, rendering it necessary to monitor the phase (flux) during computations.

5.2 Cheap transient validation

Before delving in any spectroscopy or Kerr measurement, having a first cheap validation metric is useful, just to have an idea if the setup is responding in a reasonable way. The whole system will be excited, from the feed line, via a broad band pulse to check if the resulting response, at least qualitatively, is consistent with the modes found in §4.

```

1 gaussian_gen_lib = "Gaussian_gen.lib"
2 gaussian_sbckt = "V_gaussian"
3
4 A = "20.0m"      # Amplitude (Volt)
5 t0 = "30ns"      # Central instant
6 sigma = "4ns"    # Standard deviation
7 fc = "5.2G"      # Carrier frequency
8
9 with open(gaussian_gen_lib, 'w') as f:
10     f.write(f"* Gaussian generator\n")
11     f.write(f".subckt {gaussian_sbckt} top bottom A={A} t0={t0}
12         sigma={sigma} fc={fc}\n\n")
13     f.write("B_gauss top bottom V={A*exp(-pow(time - t0, 2)/(2*pow(
14         sigma, 2)))*sin(2*pi*fc*time)}\n\n")
15     f.write(f".ends {gaussian_sbckt}\n")

```

Listing 5.3. Sinusoidal generator modulated by a gaussian envelope, used to drive the first cheap validation step.

A Gaussian-modulated tone is the right choice in this case, unlike in the dedicated spectroscopy sections that will follow, since with $\sigma = 4$ ns its bandwidth is on the order of $1/(2\pi\sigma) \sim 40$ MHz around f_c , broad enough to briefly excite several nearby qubit modes at once from a single transient run, which for the same reason renders this signal not suited for precise, single-mode frequency or a small Kerr shift analysis.

```

1 # ...
2 f.write(f".include {modello_sp}\n\n")      # VF linear network
3 f.write(f".include {gaussian_gen_lib}\n\n") # Gaussian gen
4 f.write(f".include {nome_lib_jj_lc}\n\n")  # JJ model

```

```

5      f.write(f".options method=gear itl4=400\n\n") # More stable
      solver for this type of circuit
6      f.write(f"X1 p1 p2 p3 p4 p5 p6 p7 {nome_subckt}\n\n")
7      for j in range(5):
8          k=j+3
9
10         f.write(f"X_JJ{k} p{k} 0 {nome_sbckt_lc} Lj={L_j[j]} Cj={C_j
            } R_shunt={R_sht}\n")
11
12     f.write(f"R_in p_add p1 {Z_ref}\n")
13     f.write(f"X_voltage p_add 0 {gaussian_sbckt}\n")
14     f.write(f"R_out p2 0 {Z_ref}\n")
15     f.write(f"\n.tran {0} {t_stop} {0} {t_step_max} startup
      convreport\n")
16 # ...

```

Listing 5.4. Nonlinear network used for the cheap validation step. X1 imports the model of the linear network constructed by the vectorfitting step whereas X_JJ is the JJ model.

Listing 5.4 shows the code to generate the nonlinear netlist here used. L_j sweeps the five values already mentioned in this work (6, 8, 10, 12, 14 nH), and is directly connected to the qubit ports in place of X_JJ in the fully linear netlist. Now, even though the implementation method may differ, like in JoSIM, here an RCSJ model is used. Therefore in addition to the inductive component both a capacitance C_j and resistor R_{sht} are shunted. $C_j = 1$ fF represents only the junction's self capacitance, since the Xmon's capacitance is already inside the multiport model, and gives the solver a displacement current path. $R_{sht} = 100$ k Ω is added for numerical stability. It limits the quality factor of each mode to $Q \approx R_{sht}/Z^{eff} \approx 260\text{--}400$, far below the radiative one of Chapter 4: the simulated resonances are therefore broadened, while their frequencies and the Kerr shifts are affected only at second order. The resulting two netlists will be identical with the sole exception of the inductive component of the junction, in one a linear inductance, in the other a JJ inductance. `.options method=gear` selects LTspice's implicit, A-stable integration method rather than the default trapezoidal one, more suited for a stiff circuit like the one under analysis. Figures 5.1 and 5.2 show the resulting time traces of a simulation run with a carrier $f_c = 5.2$ GHz and an amplitude of $A = 20$ mV. Q_2 shows a textbook delayed, ring-down response: its envelope peaks about 2.3 ns after the drive's peak, and continues oscillating for several nanoseconds after the drive itself has died away, the expected signature of a damped oscillator driven near resonance. Q_3 instead shows a more irregular envelope, with a distinct shoulder before the main peak, this distortion is not to be confused with a Q_3 feature, but is an early sign of crosstalk from the strongly driven Q_2 , an analysis left to the last chapter of this work dedicated to crosstalk. Lastly, driving in the range from 5.6 GHz to 5.9 GHz may produce a response that barely decays, with reason becoming clear in the last chapter, dedicated to crosstalk.

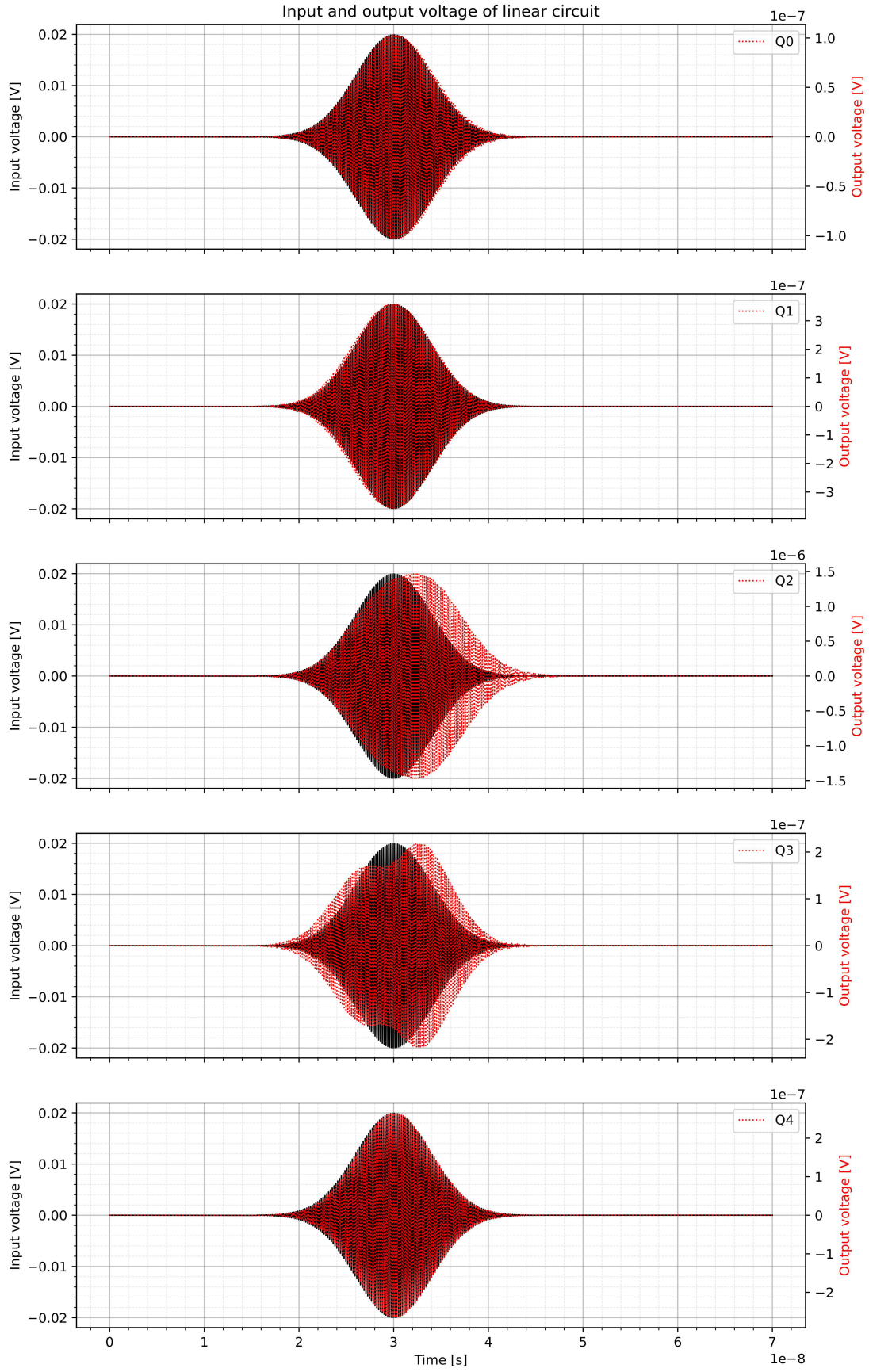


Figure 5.1. Linear network output of the first validation test run at $f_c = 5.2$ GHz. Q_2 responding correctly, while Q_3 shows cross-talk effects.

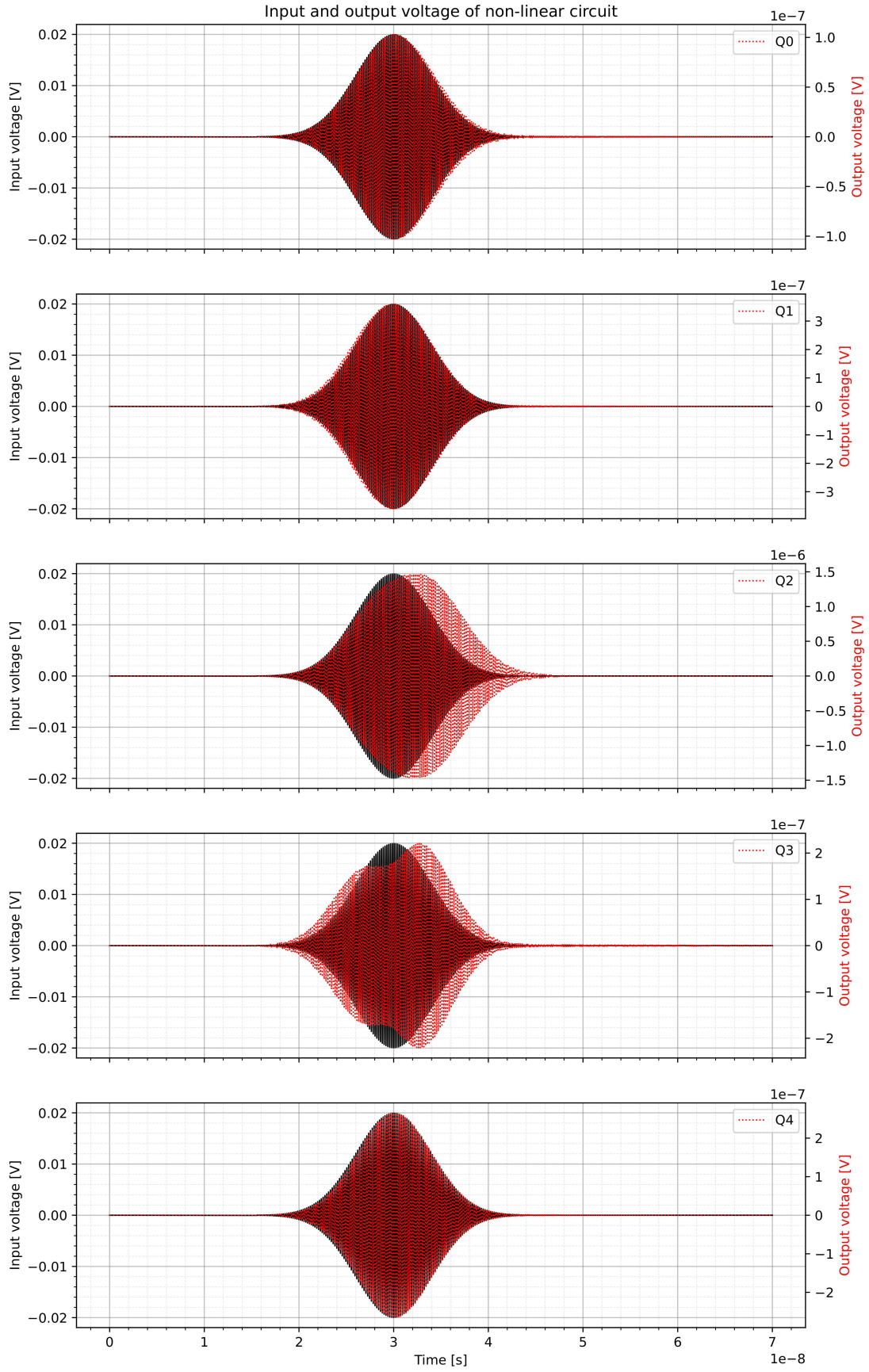


Figure 5.2. Nonlinear network output of the first validation test run at $f_c = 5.2$ GHz. Q_2 responding correctly, while Q_3 shows cross-talk effects.

Table 5.1. BBQ validation of the cheap nonlinear netlist run, comparing frequencies found in table 4.5, with the peaks found through the FFT algorithm.

Qubit	f_{mode} (GHz)	f_{01} (GHz)	FFT peak (GHz)	Δ vs f_{01} (MHz)
Q_4	4.4739	4.3674	4.6571	+289.74
Q_3	4.8218	4.7172	5.0143	+297.09
Q_2	5.2633	5.1626	5.2143	+51.69
Q_1	5.8723	5.7734	5.4857	-287.69
Q_0	6.7828	6.6833	6.6571	-26.16

5.3 Fast validation against BBQ (FFT)

Having done in the previous section a qualitative check, it is now natural to delve into a quantitative comparison against the analytical BBQ results of Chapter 4. This can be achieved by reusing the same data of nonlinear network run before, therefore requiring no additional simulation cost. For each qubit port p_k , the phase (flux) $\varphi(t) = \int V(t)dt/\phi_0$ is reconstructed and its FFT is compared against the frequencies f_{mode} and f_{01} of table 4.5.

```

1 def fft_fluxo(t, v, phi0, dt_uniforme=10e-12, finestra=True):
2
3     t_u = np.arange(t[0], t[-1], dt_uniforme) # Uniform time step
4     v_u = np.interp(t_u, t, v)
5     flux_u = cumulative_trapezoid(v_u, t_u, initial=0.0) / phi0
6     sig = flux_u * np.hanning(len(flux_u)) if finestra else flux_u
7     spettro = np.abs(np.fft.rfft(sig))
8     freq = np.fft.rfftfreq(len(sig), d=dt_uniforme)
9     return freq, spettro, t_u, flux_u

```

Listing 5.5. Function to calculate the phase (flux) of a junction and apply the FFT on it.

Three implementation choices are worth noting:

- LTspice's internal time step is adaptive, but the FFT requires a uniform sample spacing, thus the need to interpolate the voltage over a uniform time grid
- The transient record is finite (70 ns), with most of the signal being null before and after the pulse. An abrupt rectangular truncation of a finite-duration record spreads spectral power into neighboring bins. Using a Hann window instead thins out to zero smoothly at both ends, reducing spectral leakage, at the cost of a slightly broadened main lobe.
- Peak search is fixed around $f_{01} \pm 300$ MHz to avoid bumping into low-frequency artifacts and/or harmonics of the drive itself

Q_2 , the closest to the drive's carrier frequency, validates within tens of MHz from the BBQ prediction, as shown in table 5.1. Additionally, Q_0 , is the only qubit not driven near resonance that gets validated by this metric, presumably because it is the furthest in frequency and physically, effectively reducing the crosstalk towards it. Differently Q_1 , Q_3 and Q_4 show deviations close to 300 MHz, suspiciously enough, this is the exact edge of the used window. This suggests

a crosstalk contamination coming from the spectral leakage at the edges, the qubit response being weaker than this contamination. Q_1 is also affected by the shared feed line, via a global mode found to be within 5.6 GHz and 5.9 GHz more clearly motivated in the crosstalk dedicated chapter. Combining all of this information, it becomes clear that, this is not a robust validation metric for any qubit except the driven¹ one, its main advantage is how cheap it is to evaluate as the non linear run takes ~ 20 s with the FFT routine adding ~ 2 s.

5.4 Frequency sweep: qubit mode and classical Kerr shift

This section moves from the single, broadband validation of the previous section, to a dedicated qubit per qubit measurement. The objective is to identify and locate the qubit's mode inside the LTspice circuit and compare it to the previously found f_{mode} and f_{01} , then, by comparing linear and non linear circuit responses, at the same drive, the classical signature of Kerr nonlinearity is estimated. Unlike section §5.3, where the drive is injected from the feed line input, here it is instead directly connected to the selected qubit node p_q . This is done to avoid bumping into the port-to-port transfer function $H_{p_1 \rightarrow p_q}(f_c)$, which monotonically grows with frequency covering the real qubit's resonance.

```
1 B_drv p_q 0 I = I_sw*sin(2*pi*fc*time)*(1 - exp(-time/tau_on))
```

Listing 5.6. Frequency sweep behavioral generator used as current drive for the qubit excitation.

- A parallel voltage source would directly fix $V(p_q)$, thus a current drive is preferred, as it allows the node voltage to develop freely in response to the network's own impedance
- A continuous, monochromatic tone with a soft turn-on $1 - e^{-t/\tau_{on}}$ is preferred over a Gaussian burst. The gaussian pulse, having $\sigma = 4$ ns gives a bandwidth of ~ 90 MHz (FWHM, i.e. $\sigma_f = 1/(2\pi\sigma) \approx 40$ MHz) which is comparable or higher than the linewidths being measured thus risking to hide small Kerr shifts entirely. A continuous wave (CW) tone is monochromatic by construction, same procedure used experimentally by Yamaji et al. [21], which together with the soft turn on avoids injecting broadband energy, that would excite every mode.

Another important implementation detail is that, steady-state amplitude of $\varphi(t) = \int V dt / \phi_0$ on p_q is measured, instead of $V(p_q)$, as $\varphi = I(L_{ref})/I_c$ is the true amplitude of the resonator, whereas using $V = \phi_0 d\varphi/dt$ carries an extra ω factor that tilts the curve.

```
1 def _ampiezze(raw, win, dt=5e-12):
2     t = np.abs(np.asarray(raw.get_trace(0).get_wave()))
```

¹It is important to not forget to check the phase values to be sure to be within the validity range of the JJ model, here this is done through a couple of lines of code added together with the FFT function defined in listing 5.5. For the presented run one obtains $\varphi_{max} \approx 0.136 \text{ rad} < 1$

Table 5.2. Fixed current amplitude sweep results.

Quantity	Q_0	Q_1	Q_2	Q_3	Q_4
f_{mode} (GHz)	6.7828	5.8723	5.2633	4.8218	4.4739
f_{01} (GHz)	6.6833	5.7734	5.1626	4.7172	4.3674
f_{lin} (GHz)	6.7396	5.8364	5.2316	4.7929	4.4466
f_{nl} (GHz)	6.7361	5.8302	5.2218	4.7790	4.4271
$f_{nl} - f_{lin}$ (MHz)	-3.5	-6.2	-9.8	-13.9	-19.5
φ_{max} (rad)	0.10	0.15	0.21	0.27	0.33
$f_{mode} - f_{lin}$ (MHz)	-43.2	-35.9	-31.7	-28.9	-27.3

```

3     v = np.asarray(raw.get_trace(f"V(p{q_sw})").get_wave())
4     tu = np.arange(t[0], t[-1], dt); vu = np.interp(tu, t, v)
5     m = (tu >= win[0]) & (tu <= win[1])
6     phi = cumulative_trapezoid(vu, tu, initial=0.0) / phi_0
7     seg = phi[m] - np.polyval(np.polyfit(tu[m], phi[m], 1), tu[m])
8     return 0.5 * (seg.max() - seg.min()), float(np.max(np.abs(phi)))

```

Listing 5.7. Function to find the steady state amplitude of φ , searching also for φ_{max} to check the model's validity.

Measurement window is deliberately set to discard the initial tens of ns as the early response is a transient to the drive's turn on. For each of the five qubits a fixed current amplitude drive is run around the expected resonance twice

1. **Linear:** A fully linear network, where the JJ is substituted by its linear inductance L_j , but leaving in parallel C_j and R_{sht} .
2. **Nonlinear:** Same netlist but using the JJ model of listing 5.2

Being the only difference of the two runs the inductance, either linear or not, the difference on the found frequencies $f_{nl} - f_{lin}$ is the classical Kerr shift alone.

Table 5.2 depicts the results of the described sweep applied to each qubit independently. If one tries to calculate the predicted variation from f_{mode} , due to the added C_j not present before, as the leading-order approximation $\Delta f_{pred} = -\frac{1}{2}\omega^2 C_j L_j f_{mode}$ the values obtained fall within -24.7 MHz (Q_4) to -36.9 MHz (Q_0), systematically underestimating the real value of $\sim 12\%$, presumably due to effects not accounted for at this order. φ_{max} clearly grows monotonically from Q_0 to Q_4 , this effect perfectly aligns with Z_p^{eff} , computed in table 4.4, it being $247.78 \rightarrow 385.54 \Omega$. Expected for a current-driven resonator, whose voltage, and therefore flux, amplitude scales with its impedance, thus cross-checking the results found via a frequency domain simulation, as in §4, and a time domain analysis. This same behavior is observed on the Kerr shift ($-3.5 \text{ MHz} \rightarrow -19.5 \text{ MHz}$) consistent with the Duffing-like softening [22] that grows with the drive's amplitude, and therefore scales for a fixed-amplitude current-driven simulation with the impedance. The last effect mentioned is expected to grow with the drive's amplitude, and since here none of the qubits approach a shift even close to the anharmonicity ($\sim -200 \text{ MHz}$) a dedicated amplitude sweep is carried out next.

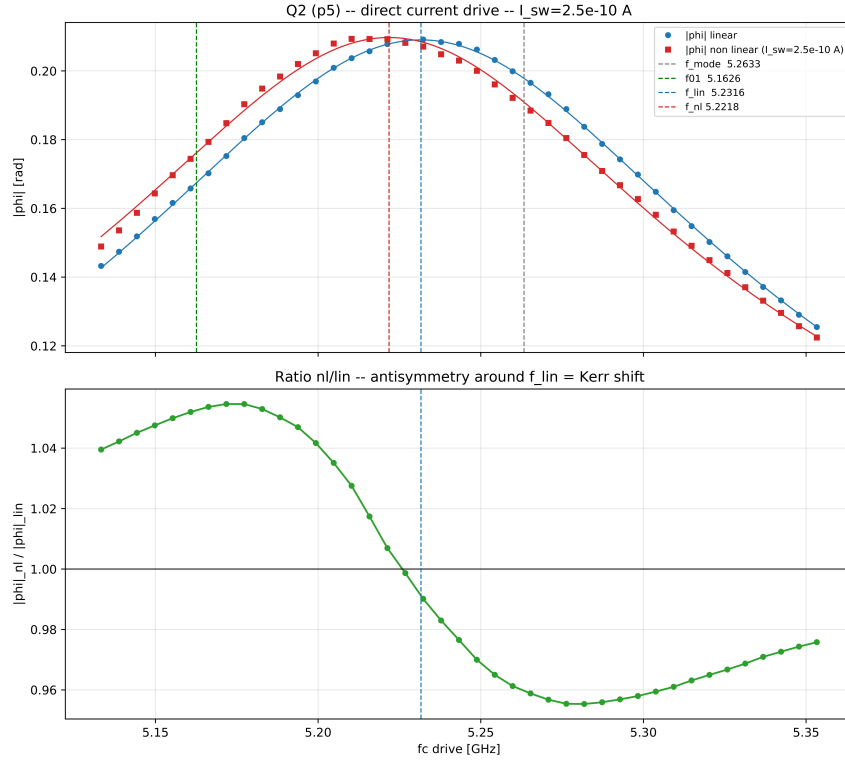


Figure 5.3. Steady-state flux amplitude $|\varphi|$ of Q_2 versus drive frequency, linear (blue) and nonlinear (red), each fitted to a Lorentzian (top). Ratio $|\varphi_{nl}|/|\varphi_{lin}|$, antisymmetric around f_{lin} , the qualitative signature of the classical Kerr shift (bottom).

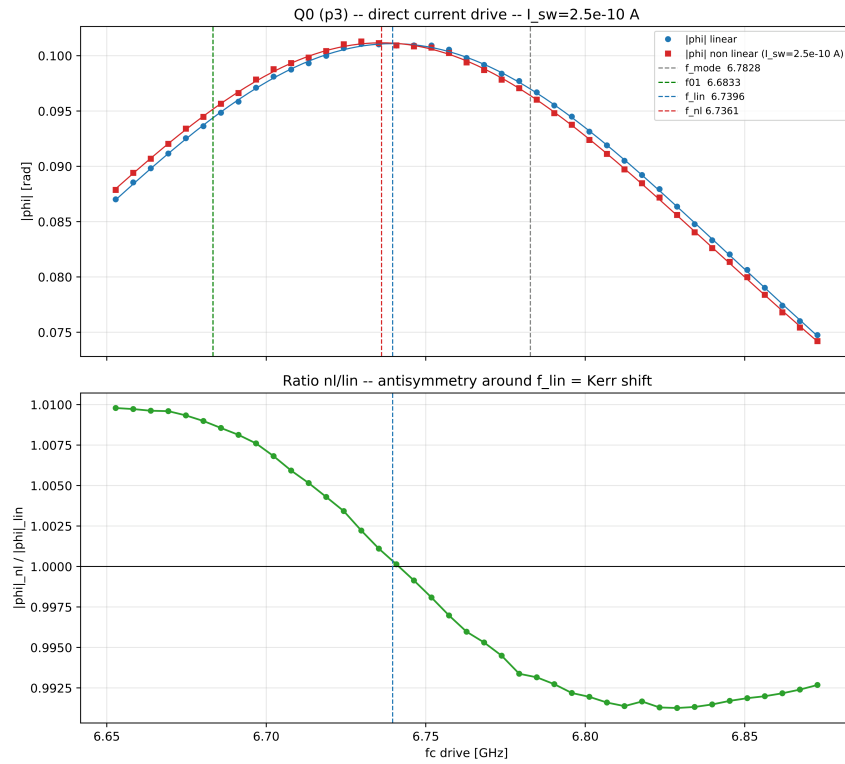


Figure 5.4. Steady-state flux amplitude $|\varphi|$ of Q_0 versus drive frequency, linear (blue) and nonlinear (red), each fitted to a Lorentzian (top). Ratio $|\varphi_{nl}|/|\varphi_{lin}|$, antisymmetric around f_{lin} , the qualitative signature of the classical Kerr shift (bottom).

5.5 Amplitude sweep: Kerr(amplitude) curve

This section builds up from the previous, where instead of having a single drive amplitude, I_{sw} is now a swept variable, with the goal being to trace the classical Kerr curve. The linear sweep is not repeated here, but taken from the previous, as f_{lin} is a property of the linear system and does not depend on the drive amplitude, this detail allows to reduce the computational cost of the simulation here carried. For each I_{sw} , here six values spanning $5 \times 10^{-11} \text{ A} \rightarrow 4 \times 10^{-10} \text{ A}$, a nonlinear frequency sweep with the same methodology of the previous is run.

```

1 def _phi_fond(raw, fc_hz, win, dt=5e-12):
2
3     t = np.abs(np.asarray(raw.get_trace(0).get_wave()))
4     v = np.asarray(raw.get_trace(f"V(p{q_sw})").get_wave())
5     tu = np.arange(t[0], t[-1], dt); vu = np.interp(tu, t, v)
6     phi = cumulative_trapezoid(vu, tu, initial=0.0) / phi_0
7     m = (tu >= win[0]) & (tu <= win[1]); ts = tu[m]; s = phi[m] -
    phi[m].mean()
8     I = 2 * np.mean(s * np.cos(2 * np.pi * fc_hz * ts))
9     Q = 2 * np.mean(s * np.sin(2 * np.pi * fc_hz * ts))
10    return float(np.hypot(I, Q))

```

Listing 5.8. Function used to obtain the fundamental projection of φ_{pk} on sin/cos at f_c , rejecting $2f_c, 3f_c \dots$ harmonics.

At larger drive amplitudes the junction's nonlinearity generates genuine harmonic content at $2f_c, 3f_c$, which the semi-peak-to-peak amplitude used before would include. For this reason the fundamental is isolated by projecting the steady state flux onto $\sin 2\pi f_c$ and $\cos 2\pi f_c$, avoiding to bias the amplitude dependence measured by inflating φ_{pk} . The junction current $I_c \sin \varphi \approx I_c(\varphi - \varphi^3/6)$ combined with the resonator's linear dynamic gives a mode equation (of the Duffing form)

$$\ddot{\varphi} + \omega_0^2 \varphi + \epsilon \varphi^3 = \text{drive}, \quad \epsilon = -\frac{\omega_0^2}{6} \quad (5.1)$$

where the standard perturbative amplitude-frequency relation for a Duffing oscillator [23], $\omega \approx \omega_0 \left(1 + \frac{3\epsilon A^2}{8\omega_0^2}\right)$ gives if combined to the above ϵ

$$\omega \approx \omega_0 \left(1 - \frac{A^2}{16}\right) \rightarrow \frac{\Delta\omega}{\omega_0} \approx -\frac{\varphi_{pk}^2}{16} \quad (5.2)$$

the classical analogue to the transmon's anharmonicity. An implementation detail worth noting is that, the frequency grid around each amplitude's resonance is deliberately asymmetric, $f_{lin} - 180 \text{ MHz}$ to $f_{lin} + 60 \text{ MHz}$, as the Kerr shift is known to move downward (softening).

Figures 5.5 and 5.6 show the obtained results² of this sweep for qubits Q_2 and Q_4 respectively,

²Data can be considered valid as the maximum found phase is $\varphi_{max} \approx 0.54 < 1 \text{ rad}$, found for Q_4 .

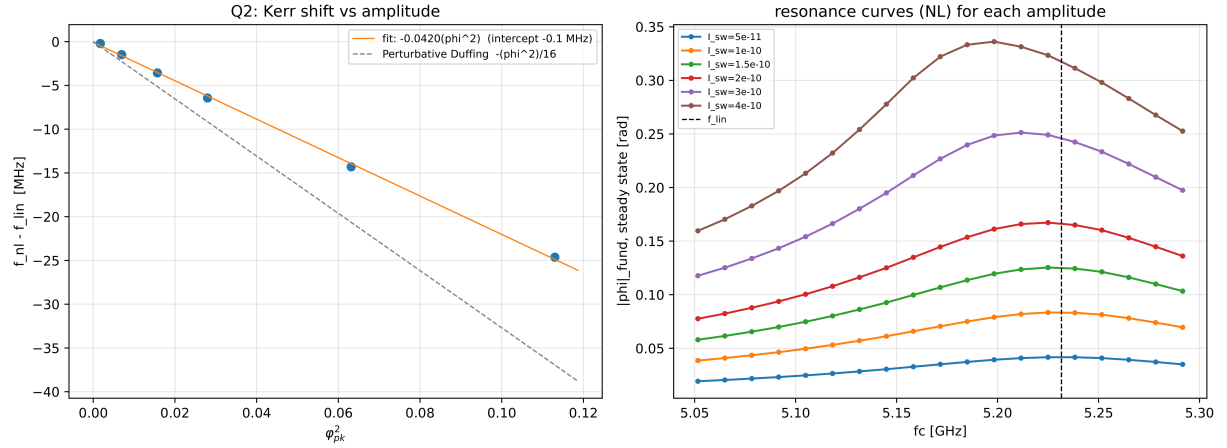


Figure 5.5. (Left) Compares the ideal curve with a slope of $-f_{lin}/16$ with the found slope for Q_2 . (Right) All the frequency sweeps, with varying I_{sw} are plotted to show how the peak moves backwards with amplitude.

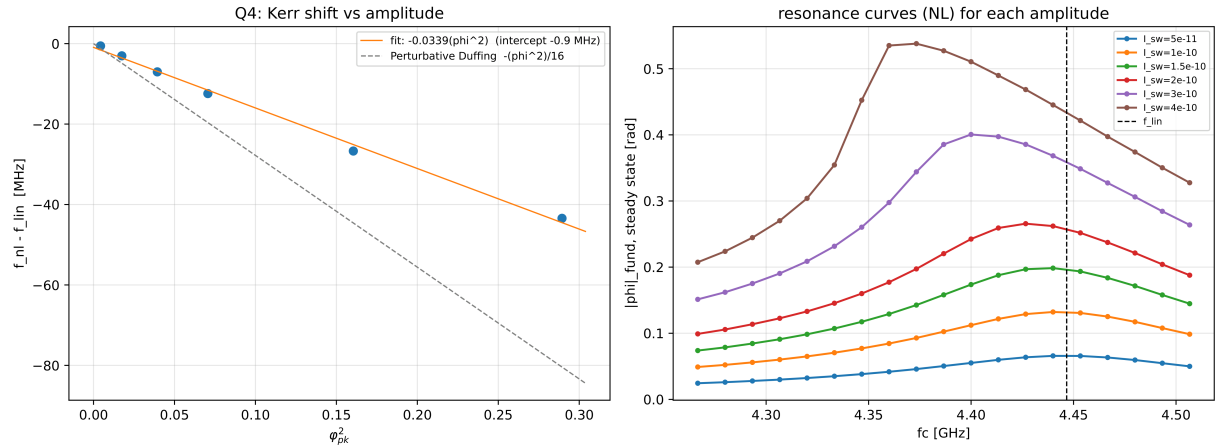


Figure 5.6. (Left) Compares the ideal curve with a slope of $-f_{lin}/16$ with the found slope for Q_4 . (Right) All the frequency sweeps, with varying I_{sw} are plotted to show how the peak moves backwards with amplitude.

chosen as representative examples. In both cases, the shift $\Delta f = f_{nl} - f_{lin}$ grows monotonically in the negative direction with I_{sw} , and by plotting it against ϕ_{pk}^2 confirms the quadratic scaling expected. The Kerr shift curves depicted in figures 5.5 and 5.6 are obtained from fitting $\Delta f / f_{lin}$, resulting in the slopes of $k_{fit} \approx -0.042$ for Q_2 and $k_{fit} \approx -0.034$ for Q_4 , but are plotted as $\Delta f = k_{fit} \phi_{pk}^2 f_{lin}$ to directly depict the frequency shift. In both cases, $|k_{fit}| < 1/16$, making Q_4 's slope $\sim 54\%$ of the ideal one, and Q_2 around $\sim 67\%$. The $-\phi_{pk}^2/16$ relation describes an ideal, isolated oscillator, whereas in the simulated circuit the mode is loaded by the surrounding network, damped by R_{shr} , and measured over a finite simulation window, none of which enters the theoretical relation; an exact match of the slope is therefore not expected. Q_4 , which reaches a higher phase ($\phi_{max,Q_4} \approx 0.54$ rad against $\phi_{max,Q_2} \approx 0.34$ rad), deviates more from the ideal slope. By comparing the resonance curves plotted in 5.5 and 5.6, it is evident how Q_4 is much deeper in the non linear regime, as its peak shifts much more left than Q_2 , confirmed also by table 5.3 where the max Δf of Q_4 is ~ 1.8 times higher. Lastly, extrapolating to $\phi_{pk} = 1$, from

Table 5.3. Nonlinear frequency shift due to the drive's current amplitude (qubits Q_2 and Q_4).

I_{sw} (A)	f_{nl} (GHz)	$[Q_2]$	Δf (MHz)	$[Q_2]$	f_{nl} (GHz)	$[Q_4]$	Δf (MHz)	$[Q_4]$
5×10^{-11}	5.2314		-0.21		4.4461		-0.57	
1×10^{-10}	5.2301		-1.48		4.4436		-3.03	
1.5×10^{-10}	5.2280		-3.57		4.4396		-7.02	
2×10^{-10}	5.2252		-6.44		4.4342		-12.41	
3×10^{-10}	5.2173		-14.30		4.4199		-26.70	
4×10^{-10}	5.2070		-24.63		4.4032		-43.44	

the linear fits obtained for the Kerr curve, the scale at which the classical shift is expected to become comparable to the transmon's anharmonicity, results in $\Delta f \approx -219.7$ MHz and $\Delta f \approx -151.1$ MHz for Q_2 and Q_4 respectively. Comparing those values with the anharmonicities found in table 4.5 leads to the conclusion that, for Q_2 the extrapolated value has an overshoot of $\sim 10\%$ whereas in Q_4 case it is an undershoot of $\sim 27\%$. This asymmetry is coherent with the fact that the extrapolation itself is only valid within the perturbative regime it was built for, so it is expected to increasingly deviate from α_p the more the underlying data already departs from the ideal $-1/16$ slope.

Chapter 6

Crosstalk

A multi-qubit device, like the one used in this work and presented in chapter 3, does not consist of independent or isolated channels, but every port shares the same EM environment, which *will* couple to all the qubits, both in wanted and unwanted ways. This chapter gives a hint to the estimation of the crosstalk between qubits and/or qubits and feed line, closing the observations done for the shoulder of Q_3 's response in §5.2 (Figure 5.2) as well as the slowly-decaying response obtained driving between 5.6 and 5.9 GHz.

6.1 Methodology

Crosstalk is characterized through the cross-drive ratio introduced by Song et al. [24], the S-parameter-style figure of merit already used by Wang et al. [25] to build a full microwave crosstalk matrix

$$X_{k,i} = 20 \log_{10} \left(\frac{|\varphi_k(f_i)|}{|\varphi_i(f_i)|} \right) \quad [\text{dB}] \quad (6.1)$$

where $\varphi_k = \int V(p_k) dt / \phi_0$ is the same steady-state flux amplitude already used throughout Chapter 5, and f_i is the frequency at which qubit i is stimulated. Unlike in the Kerr characterization, here the junction's non linearity plays no role, as in the small signal regime crosstalk is a linear leakage phenomenon fixed entirely by the shared electromagnetic topology, therefore JJs are substituted with their linear inductance L_j .

Two drive methodologies are used

- Current drive directly injected at each qubit's node p_k giving the drive crosstalk
- Voltage drive applied at the shared feedline giving the feed line crosstalk

The first methodology comes from a limitation of the extracted model already discussed in §3: only 7 ports were chosen to be accessible, so the dedicated XY/Z control routes are not accessible and the real measurement using those lines cannot be carried out. Thus the drive crosstalk (X_{addr} and X_{res}) still captures the genuine capacitive coupling between physically adjacent qubit islands, already present in the fitted admittance matrix regardless of which port injects the drive,

Table 6.1. Drive crosstalk X_{addr} (row = responding qubit, column = driven qubit at f_i). X_{res} , the worst-case value over the whole ± 450 MHz band, agrees with X_{addr} to within 1 dB for most entries, with the two adjacent-qubit pairs (Q_3, Q_2) and (Q_4, Q_3) departing further, by 2.4 dB and 1.2 dB respectively. Note that the matrix is not symmetric, despite the network being reciprocal, since each column is evaluated at a different frequency f_i (that qubit’s own mode), so entries (k, i) and (i, k) compare the same port pair at two different frequencies rather than the same one.

	Q_0	Q_1	Q_2	Q_3	Q_4
Q_0	0.0	-29.9	-46.8	-56.9	-71.5
Q_1	-30.1	0.0	-30.3	-41.2	-58.7
Q_2	-36.5	-32.1	0.0	-26.3	-45.0
Q_3	-40.6	-33.9	-26.2	0.0	-21.3
Q_4	-50.7	-43.4	-44.0	-20.1	0.0

what is missing is the additional contribution given by the control line’s routing, making the found values a floor. Reference levels against which the results are compared, from Song et al. [24]: -30 to -40 dB for package-mediated coupling, worsening to as little as $+2$ to -9 dB for crossover-induced leakage.

6.2 Drive crosstalk

One at a time, for each qubit i , the current $I_{xt} = 5 \times 10^{-11}$ A is injected directly at its node p_i , spanning ± 450 MHz around the tabulated f_{mode} of qubit i . During this process every qubit has connected its linear L_j , C_j and R_{sh} with the feed line closed on 50Ω terminations.

```

1 def _phi_k(raw, port, fc_hz, win, dt=5e-12):
2
3     t = np.abs(np.asarray(raw.get_trace(0).get_wave()))
4     v = np.asarray(raw.get_trace(f"V(p{port})").get_wave())
5     tu = np.arange(t[0], t[-1], dt); vu = np.interp(tu, t, v)
6     phi = cumulative_trapezoid(vu, tu, initial=0.0) / phi_0
7     m = (tu >= win[0]) & (tu <= win[1]); ts = tu[m]; s = phi[m] -
    phi[m].mean()
8     I = 2 * np.mean(s * np.cos(2 * np.pi * fc_hz * ts))
9     Q = 2 * np.mean(s * np.sin(2 * np.pi * fc_hz * ts))
10    return float(np.hypot(I, Q))

```

Listing 6.1. Amplitude of the φ fundamental at f_c on node p_k .

The peak of each qubit’s response, f_i matches the f_{lin} already tabulated in Table 5.2 to within a few MHz.

The worst off-diagonal pair, highlighted in bold in table 6.1, is the same couple that in §4.3.2 showed the highest cross-Kerr coupling (-7.2149 MHz), consistent with nearest-neighbor capacitive coupling combined with the smallest mode’s separation of ~ 350 MHz. Crosstalk generally decreases both with frequency and physical separation, almost exactly what is observed as

-21 dB is the coupling for the closest pair down to -71.5 dB for the two extremes of the chain, Q_0 and Q_4 . This trend, however, is not strictly set by frequency as Q_0 - Q_1 , at $\Delta f = 0.91$ GHz, has a stronger coupling (-30 dB) than Q_2 - Q_4 (-44.5 dB) which are further in physical distance but closer in frequency $\Delta f = 0.79$ GHz. This mentioned example, together with the 1-diagonal and -1 -diagonal values representing neighbouring qubits in the chip, leads to the conclusion that the dominant effect is physical proximity followed by frequency. If a comparison is done with the values mentioned in §6.1, all of the elements in the 1-diagonal and -1 -diagonal either exceed or are very close to the -30 dB limit, though all values are far from the -9 dB crossover-induced worst case, consistent with the fact that here only the proximity channel is captured but not the added contribution of control routes. This same channel also explains the shoulder observed on Q_3 's response while driving Q_2 in the cheap validation of §5.2 (Figure 5.2): direct capacitive coupling between the two physically adjacent islands, active regardless of the specific drive geometry used to excite it.

6.3 Feed line crosstalk

Differently from the previous section, here a voltage source, in series to a $50\ \Omega$ load, is connected to node p_1 , or in other words to the input port of the feedline, with the output closed to another $50\ \Omega$ load. The sweep spans 4.0-7.0 GHz with a drive's amplitude of $V_{fl} = 1$ mV, notice that being the circuit fully linear, scaling V_{fl} leaves X_{fl} unchanged. An additional difference from the previous is that f_i is not fitted from this sweep but taken from the tabulated f_{mode} and, for simplicity, corrected with the formula $\Delta f/f = -\frac{1}{2}\omega^2 C_j L_j$, already talked about in §5.4.

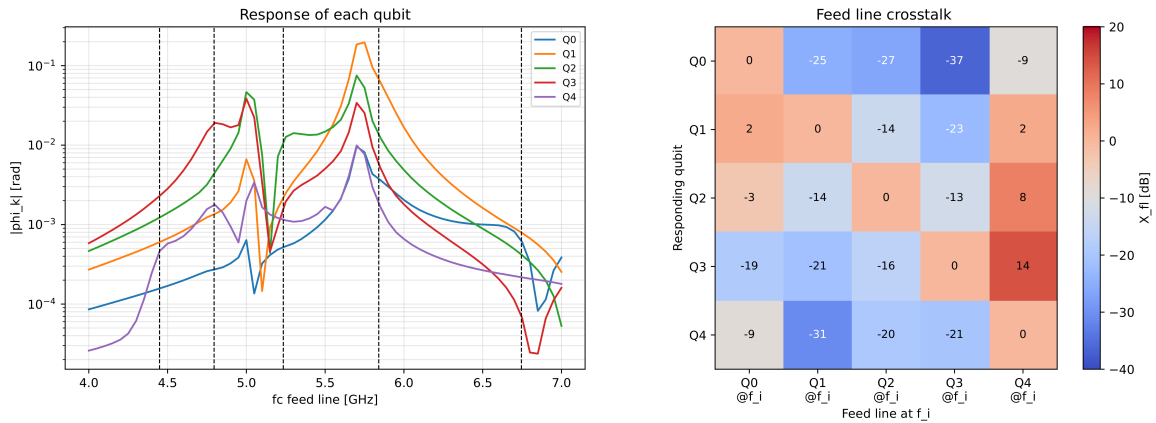


Figure 6.1. (Left) $|\varphi_k(f_c)|$ for every qubit vs. feed-line drive frequency, dashed lines marking each f_i . (Right) X_{fl} heatmap. As for X_{addr} (Table 6.1), the X_{fl} matrix is not symmetric, since each column is evaluated at a different f_i .

If the heat map of figure 6.1 is compared to values of table 6.1 it becomes clear that the feed line is dramatically less selective, several off-diagonal entries are positive, the worst being $+13.9$ dB. The reason for this behavior is evident in the qubit response (same figure 6.1) where $|\varphi_k(f_c)|$

shows a broad, dominant feature between 5.6 and 5.9 GHz, the same cavity-like mode identified in §4.5 (5.7187 GHz, $Q \approx 44233$). This behavior is relevant for:

- Observation cited in §5.2, where driving from the feed line with a gaussian pulse within 5.6 – 5.9 GHz leads to a barely decaying response. Figure 6.2 shows every qubit retaining an appreciable tail at $f_{drive} = 5.8$ GHz, particularly Q_1 , whose amplitude is still $\sim 80\%$ of its peak by the end of the simulated time.
- Citing §5.3, again for Q_1 , the FFT analysis flagged a feed-line contamination, which this dominant feature, and the fact that Q_1 is nearly degenerate with the bus at $f_{01} = 5.7734$ GHz, explains the nearly 300 MHz shift observed

A structural limitation must be stated: the extracted model has no separate coupler port, so X_{fl} is the *total* feed-line-to-qubit crosstalk, not decomposable into X_{cross} and X_{prox} as in Song et al. Eq. (8-9) [24]. The physical picture is still clear, the feed line is a shared bus and not a selective channel, thus for selective control XY/Z lines must be rendered accessible from the start, at the Ansys HFSS frequency sweep. Finally, Q_0 deserves a separate remark, as its maximum response is within the shared mode at 5.70 GHz, nowhere near its own $f_i = 6.7458$ GHz used for this sweep, and effectively showing no genuinely resolved resonance during this final test. For this reason, entries of column Q_0 should be taken with a grain of salt, as a "crosstalk relative to addressing Q_0 " is not really well defined for this qubit.

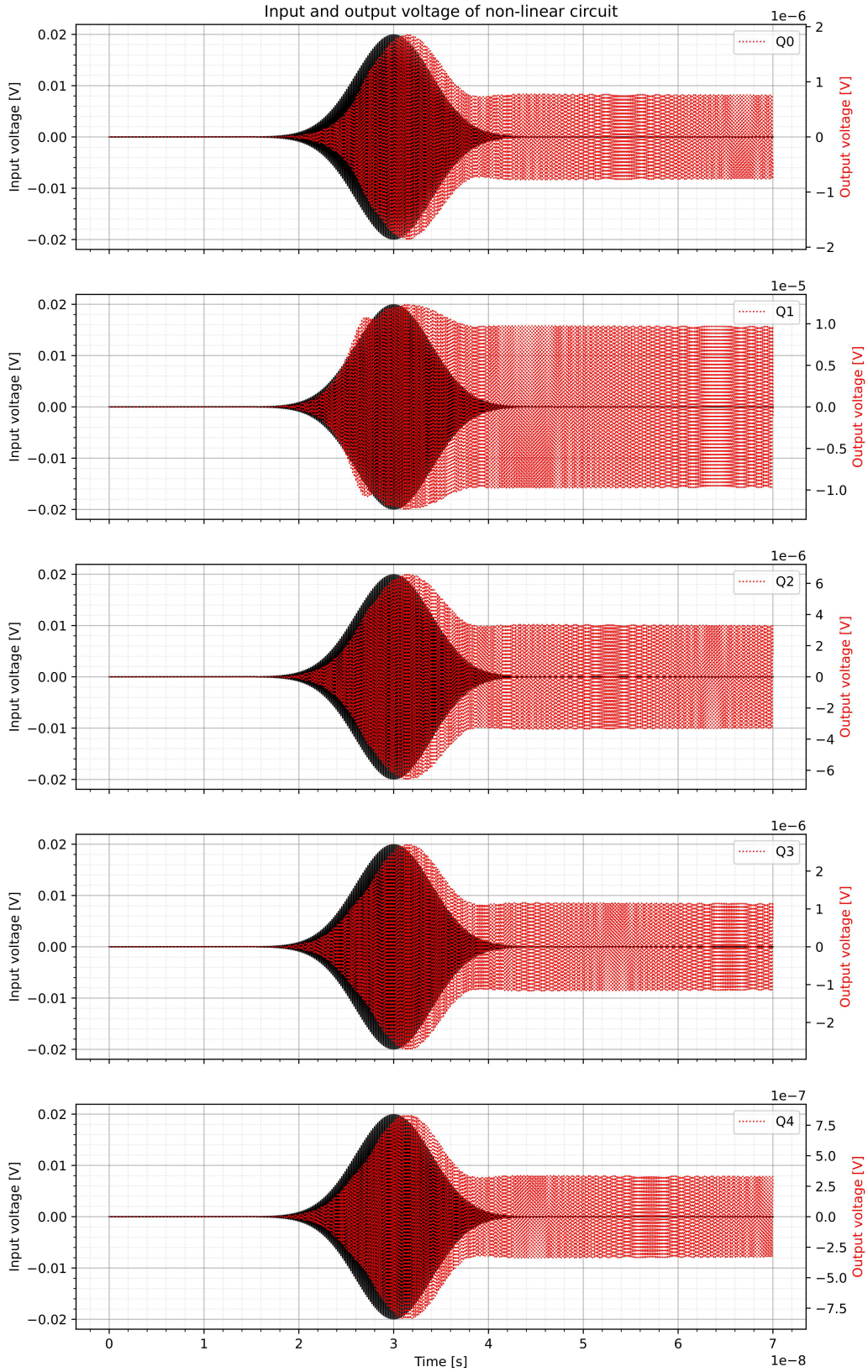


Figure 6.2. Cheap transient simulation, carried out as in §5.2, run at $f_{drive} = 5.8$ GHz, showing a slowly decaying tail for every qubit.

Conclusions

The objectives of this work have been addressed as follows. A reproducible pipeline from a Qiskit Metal layout of a five-qubit processor to a SPICE-level nonlinear circuit model was implemented, passing through a full-wave scattering simulation and a vector-fitting reduction. Mode frequencies, anharmonicities and cross-Kerr couplings were extracted from the fitted network through black-box quantisation, and compared with the nonlinear time-domain response of the same network. The linear microwave crosstalk between individually driven qubits and from the shared feed line was quantified, and the Purcell-limited relaxation time was estimated, taking into account the limitation posed by the non-passivity of the model.

The five-qubit chip was built through Qiskit Metal and characterised via a driven-modal Ansys HFSS scattering simulation, then, through a vector-fitting algorithm, reduced to a compact rational model exported as a SPICE subcircuit. Two parallel characterizations have been used: a non-linear time domain model, obtained by reinserting the JJ non linearity as a custom LTspice element, and an analytic black-box quantisation of the loaded admittance matrix. This same network, with the junctions linearised, has been used to estimate the microwave crosstalk, both due to a direct drive of the qubits at their nodes or via the feed line.

The two independent characterisations are consistent: five well-separated qubit modes were found ($4.47 \rightarrow 6.78$ GHz), with self-Kerr values between -206 and -196 MHz, while the nonlinear SPICE Kerr shift scales with the square of the phase as expected and, extrapolated to $\varphi = 1$, is of the same order as the black-box anharmonicity. Three additional cavity-like modes of the bare electromagnetic environment were identified and explained through independent evidence. A further, qualitative cross-check is given by the qubits' response amplitude under a fixed current drive, which grows from Q_0 to Q_4 following the same trend as the effective impedance obtained from black-box quantisation. Crosstalk between individually driven qubits is dominated by physical adjacency on the chip rather than by frequency separation alone, reaching -20 dB for the closest pair. Crosstalk from the shared feed line is markedly worse, up to $+13.9$ dB, driven by a high-quality-factor mode shared among several qubits, which also accounts for the slowly-decaying transient response and the FFT validation mismatch observed for the qubit closest to it.

The main technical difficulty was obtaining a numerically well-behaved nonlinear junction model in LTspice. A direct implementation through the integral of the junction voltage intro-

duced spurious numerical damping, and was replaced with a native-inductor-based model valid for $\varphi \lesssim 1$. The main limitation of the results is the residual non-passivity of the fitted macromodel, which, together with the finite frequency grid, turns the Purcell-limited relaxation time into a declared lower bound ($T_1 \geq 3.4 \mu\text{s}$) rather than an exact estimate. A second limitation is structural, the extracted network has no dedicated port for the qubits' XY/Z control lines, so the crosstalk analysis captures the proximity coupling already present in the shared electromagnetic environment, but not the additional contribution of a specific control-line routing.

Natural extensions of this work include a passivity-enforced re-fit of the macromodel, or the exact-synthesis alternative discussed in §2.3.2, to remove the lower-bound caveat on the Purcell-limited relaxation time. Additionally, the number of ports of the extracted network could be extended to include the control-line pads, to directly capture crossover-induced crosstalk. Lastly, validating the static ZZ interaction, already estimated here through the cross-Kerr coefficients, against the time-domain model.

Nomenclature

Latin symbols

C_j	Josephson-junction self-capacitance [F]
C_s	shunt capacitance of the qubit island [F]
E_c	charging energy, $E_c = e^2/2(C_j + C_s)$ [J]
E_j	Josephson energy, $E_j = L_j I_c^2$ [J]
f_{01}	qubit $0 \rightarrow 1$ transition frequency [Hz]
f_{mode}	bare frequency of a network mode [Hz]
I_c	Josephson critical current [A]
L_j	Josephson inductance, $L_j = \hbar/2eI_c$ [H]
L_{ref}	linear reference inductance of the RCSJ junction model [H]
Q_p	quality factor of mode p (black-box quantisation) [–]
R_{sh}	shunt resistance of the RCSJ junction model [Ω]
S_{ij}	scattering parameter, port j to port i [–]
T_1	energy-relaxation time [s]
$X_{k,i}$	microwave cross-drive ratio: drive on line i , response on qubit k [dB]
$Y(\omega)$	port admittance of the passive network [S]
$\mathbf{Y}_{\text{tot}}(\omega)$	loaded admittance matrix of the passive network [S]
Z_p^{eff}	effective impedance of mode p seen by its Josephson junction [Ω]
$\mathbf{Z}_{\text{tot}}(\omega)$	total impedance matrix of the passive network [Ω]

Greek symbols

α_p	transmon anharmonicity (self-Kerr) of mode p [Hz]
$\beta_{qq'pp'}$	quartic perturbation coefficient of the black-box quantisation [J]
Δ_p	Lamb shift of mode p [Hz]
ξ_{sp}	participation ratio of qubit s in mode p [–]
φ	superconducting phase difference across a Josephson junction [rad]
ϕ_0	reduced flux quantum, $\phi_0 = \hbar/2e$ [Wb]
χ_{qp}	cross-Kerr coupling between modes q and p [Hz]
ω_p	angular frequency of mode p [rad s^{-1}]

Acronyms

BBQ	Black-Box Quantisation
CW	Continuous Wave
EPR	Energy-Participation Ratio
HFSS	High-Frequency Structure Simulator (Ansys)
JJ	Josephson Junction
LOM	Lumped Oscillator Model
RCSJ	Resistively and Capacitively Shunted Junction
RMS	Root Mean Square
SPICE	Simulation Program with Integrated Circuit Emphasis
XY	in-plane microwave qubit-control line

Bibliography

- [1] Z. K. Mineev, Z. Leghtas, S. O. Mundhada, L. Christakis, I. M. Pop, and M. H. Devoret, “Energy-participation quantization of Josephson circuits,” *npj Quantum Information*, vol. 7, p. 131, 2021. doi: 10.1038/s41534-021-00461-8.
- [2] S. E. Nigg et al., “Black-box superconducting circuit quantization,” *Physical Review Letters*, vol. 108, no. 24, p. 240502, 2012. doi: 10.1103/PhysRevLett.108.240502.
- [3] N. K. Langford, *Circuit qed - lecture notes*, 2013. arXiv: 1310.1897 [quant-ph]. [Online]. Available: <https://arxiv.org/abs/1310.1897>.
- [4] U. Vool and M. Devoret, “Introduction to quantum electromagnetic circuits,” *International Journal of Circuit Theory and Applications*, vol. 45, no. 7, pp. 897–934, 2017. doi: 10.1002/cta.2359.
- [5] J. Koch et al., “Charge-insensitive qubit design derived from the Cooper pair box,” *Physical Review A*, vol. 76, no. 4, p. 042319, 2007. doi: 10.1103/PhysRevA.76.042319.
- [6] T. E. Roth, R. Ma, and W. C. Chew, “The transmon qubit for electromagnetics engineers: An introduction,” *IEEE Antennas and Propagation Magazine*, vol. 65, no. 2, pp. 8–20, Apr. 2023, issn: 1558-4143. doi: 10.1109/map.2022.3176593. [Online]. Available: <http://dx.doi.org/10.1109/MAP.2022.3176593>.
- [7] R. Barends et al., “Coherent Josephson qubit suitable for scalable quantum integrated circuits,” *Physical Review Letters*, vol. 111, no. 8, p. 080502, 2013. doi: 10.1103/PhysRevLett.111.080502.
- [8] S. Halder, K. A. Chauhan, M. B. Barfar, S. Ganguly, and S. Devendrababu, *Superconducting Qubit Design Using Qiskit Metal*. Apress Berkeley, CA, 2025.
- [9] B. Gustavsen and A. Semlyen, “Rational approximation of frequency domain responses by vector fitting,” *IEEE Transactions on Power Delivery*, vol. 14, no. 3, pp. 1052–1061, 1999. doi: 10.1109/61.772353.

- [10] B. Gustavsen, “Improving the pole relocating properties of vector fitting,” *IEEE Transactions on Power Delivery*, vol. 21, no. 3, pp. 1587–1592, 2006. DOI: 10.1109/TPWRD.2005.860281.
- [11] S. Grivet-Talocia and B. Gustavsen, *Passive Macromodeling: Theory and Applications* (Wiley Series in Microwave and Optical Engineering). Hoboken, NJ: John Wiley & Sons, 2016, ISBN: 9781118094914.
- [12] F. Solgun, D. W. Abraham, and D. P. DiVincenzo, “Blackbox quantization of superconducting circuits using exact impedance synthesis,” *Physical Review B*, vol. 90, no. 13, p. 134 504, 2014. DOI: 10.1103/PhysRevB.90.134504.
- [13] F. Solgun and D. P. DiVincenzo, “Multiport impedance quantization,” *Annals of Physics*, vol. 361, pp. 605–669, 2015. DOI: 10.1016/j.aop.2015.07.005.
- [14] A. Arsenovic et al., “Scikit-rf: An open source Python package for microwave network creation, analysis, and calibration,” *IEEE Microwave Magazine*, vol. 23, no. 1, pp. 98–105, 2022. DOI: 10.1109/MMM.2021.3117139.
- [15] M. C. Sarihan, *5-Qubit Josephson Processor Layout: Capstone project (qiskit metal)*, <https://github.com/quantum-device-consortium/qdw26-workshop-materials>, GitHub repository; capstone project material, Google Quantum AI. Accessed 13 September 2026, 2026.
- [16] Z. K. Minev et al., *Qiskit Metal: An open-source framework for quantum device design and analysis*, Software, 2021. DOI: 10.5281/zenodo.4618153.
- [17] R. Barends et al., “Superconducting quantum circuits at the surface code threshold for fault tolerance,” *Nature*, vol. 508, no. 7497, pp. 500–503, 2014. DOI: 10.1038/nature13171.
- [18] J. Delport, K. Jackman, P. le Roux, and C. Fourie, “Josim—superconductor spice simulator,” *IEEE Transactions on Applied Superconductivity*, vol. 29, pp. 1–5, Feb. 2019. DOI: 10.1109/TASC.2019.2897312.
- [19] qedalab, *pyjosim: Python bindings for JoSIM*, Software repository, accessed September 2026, 2026. Accessed: Sep. 18, 2026. [Online]. Available: <https://github.com/qedalab/pyjosim>.
- [20] qedalab, *josim-tools: Analysis and optimization tools for superconducting circuits built on JoSIM*, Software repository, version 0.1.0, accessed September 2026, 2026. Accessed: Sep. 18, 2026. [Online]. Available: <https://github.com/qedalab/josim-tools>.
- [21] T. Yamaji et al., “Spectroscopic observation of the crossover from a classical Duffing oscillator to a Kerr parametric oscillator,” *Physical Review A*, vol. 105, no. 2, p. 023 519, 2022. DOI: 10.1103/PhysRevA.105.023519.

- [22] J. Bourassa, F. Beaudoin, J. M. Gambetta, and A. Blais, “Josephson-junction-embedded transmission-line resonators: From Kerr medium to in-line transmon,” *Physical Review A*, vol. 86, no. 1, p. 013 814, 2012. doi: 10.1103/PhysRevA.86.013814.
- [23] S. H. Strogatz, *Nonlinear Dynamics and Chaos: With Applications to Physics, Biology, Chemistry, and Engineering*, 2nd. Boca Raton, FL: CRC Press, 2015, ISBN: 9780813349107.
- [24] Y. Song et al., “Microwave crosstalk in planar superconducting quantum devices,” *arXiv preprint*, 2026. arXiv: 2606.02440.
- [25] Y. Wang, P. Zhao, Y. Jin, and H. Yu, “Control and mitigation of microwave crosstalk effect with superconducting qubits,” *Applied Physics Letters*, vol. 121, no. 15, p. 152 602, 2022. doi: 10.1063/5.0115393.