Analysis of arbitrary superconducting quantum circuits accompanied by a Python package: SQcircuit

Taha Rajabzadeh[1], Zhaoyou Wang[2], Nathan Lee[2], Takuma Makihara[2], Yudan Guo[2], and Amir H. Safavi-Naeini[2]

[1]Department of Electrical Engineering, Stanford University, Stanford, CA 94305 USA

[2]E. L. Ginzton Laboratory and the Department of Applied Physics, Stanford University, Stanford, CA 94305 USA

Superconducting quantum circuits are a promising hardware platform for realizing a fault-tolerant quantum computer. Accelerating progress in this field of research demands general approaches and computational tools to analyze and design more complex superconducting circuits. We develop a framework to systematically construct a superconducting quantum circuit's quantized Hamiltonian from its physical description. As is often the case with quantum descriptions of multicoordinate systems, the complexity rises rapidly with the number of variables. Therefore, we introduce a set of coordinate transformations with which we can find bases to diagonalize the Hamiltonian efficiently. Furthermore, we broaden our framework's scope to calculate the circuit's key properties required for optimizing and discovering novel qubits. We implement the methods described in this work in an open-source Python package `SQcircuit`. In this manuscript, we introduce the reader to the `SQcircuit` environment and functionality. We show through a series of examples how to analyze a number of interesting quantum circuits and obtain features such as the spectrum, coherence times, transition matrix elements, coupling operators, and the phase coordinate representation of eigenfunctions.

Taha Rajabzadeh: tahar@stanford.edu

Amir H. Safavi-Naeini: safavi@stanford.edu

Contents

1 Introduction

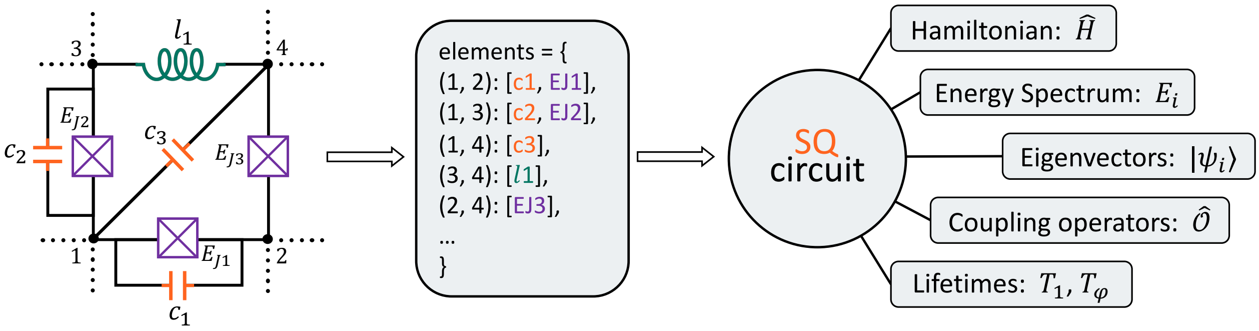
Two of the important lines of research in superconducting quantum circuits that run parallel to the current quest to realize large-scale fault-tolerant quantum computers are (1) exploring the spaces of novel circuitry that more robustly encode quantum information [1, 2, 3, 4, 5]; and (2) coupling superconducting qubits to other quantum systems to form hybrid devices [6] that implement new memory, sensing, and communications functions. Both directions are characterized by increasing complexities in the circuit layout and demand a general approach toward analyzing larger and more complex circuits. To study the quantum behavior of a superconducting quantum circuit, we need to diagonalize the Hamiltonian of the circuit, derived from the circuit quantization method [7, 8, 9, 10]. However, the computational cost for diagonalization increases exponentially with the circuit size, and a proper basis choice is necessary for alleviating this problem when solving larger circuits. Existing numerical packages have made significant and valuable contributions in this direction [11, 12, 13]. We build on these earlier efforts by directly addressing the dual challenges of scalability and generality. We move beyond restrictions on the anharmonicity or types of qubits, and realize automatic constructions that enable more efficient diagonalization of larger circuits.

In this work, we develop a systematic framework that constructs the Hamiltonian for an arbitrary superconducting quantum circuit. We pay close attention to proper accounting for time-dependent external fluxes [14, 15] and represent the Hamiltonian in a basis that requires fewer computational resources for diagonalization. Additionally, we describe how to calculate properties essential to designing new superconducting qubits, such as coherence times in the presence of various noise processes, robustness to fabrication imperfections, and coupling to external control lines. We implement these approaches and provide an open-source Python-based package `SQcircuit`. Relying on NumPy, SciPy, and QuTiP [16] packages, `SQcircuit` is a realization of the framework in this paper and provides a general approach for analyzing arbitrary circuits, complementing and extending the functionalities of the aforementioned packages [12, 11, 13]. Users can define arbitrary circuits and, through highly-abstracted routines, conveniently compute and visualize quantities of interest such as energy spectra, eigenvectors, and coherence properties (Fig.1). We expect the package to significantly accelerate the analysis and design processes of complex superconducting circuits.

The remainder of the manuscript is structured as follows: Sec. 2 reviews the general approach to obtaining the Hamiltonian for an arbitrary superconducting circuit. In Sec. 3, we derive a proper transformation of coordinates to find an appropriate basis for efficiently diagonalizing the Hamiltonian of any superconducting circuit. In Sec. 4, we walk the reader through the SQcircuit library, and use the $0-\pi$ qubit [17, 18, 19] as an example to demonstrate how to define a circuit in `SQcircuit` and obtain basic circuit properties such as eigenenergies and eigenvectors. Next, in Sec. 5, we expand the developed theory in previous sections to extract other circuit properties such as matrix elements, lifetimes, and so on, along with `SQcircuit` functionalities to obtain those features. We conclude by presenting an outlook in Sec. 7.

2 Circuit Hamiltonian

Our description of a quantum circuit follows previous studies [20, 7, 21]. We start with a circuit consisting of capacitors and inductors as branches. Applying Faraday's law around loops in the circuit leads to constraints that impose relationships between different branch fluxes and reduce the number of independent variables. A systematic approach to obtaining the independent degrees of freedom of the system is to define node fluxes by choosing a minimum spanning tree in the graph and assigning a flux variable ϕ_i at every node i [7]. Faraday's law then leads to a number of relations between the branch fluxes $\phi_{b,k}$ and node fluxes ϕ_i (subscript b denotes branch variables), usually of the form $\phi_{b,k} = \phi_i - \phi_j + \phi_{\text{ext},b,k}$, where nodes i, j are the endpoints of kth branch. Here, $\phi_{\text{ext},b,k}$ is not a dynamical variable but a constraint imposed on branches that are not part of the minimum spanning tree. For these branches, the value of $\phi_{\text{ext},b,k}$ is found by considering the external flux in the loop formed by the branch (we refer to external fluxes of the lth loop by $\phi_{\text{ext},l}$ which is different from value of the external flux at the branch $\phi_{\text{ext},b,k}$). In a circuit with



```
elements = {
(1, 2): [c1, EJ1],
(1, 3): [c2, EJ2],
(1, 4): [c3],
(3, 4): [l1],
(2, 4): [EJ3],
...
}
```

Figure 1: The general overview of SQcircuit. By defining the net-list of an arbitrary circuit in SQcircuit environment, users can compute and visualize quantities of interest such as energy spectra, coupling operators, and etc, through highly-abstracted routines.

$n_\mathrm{N}+1$ nodes and n_L inductive loops, we have only n_N independent variables, and we rewrite the relation between kth branch flux variable between nodes i,j in terms of vectorized node fluxes $\boldsymbol{\Phi}=[\phi_1,...,\phi_{n_\mathrm{N}}]$ (we have set the ground node $i=0$ to a fixed potential $\phi_0=0$) and external fluxes $\boldsymbol{\Phi}_\mathrm{ext}=[\phi_{\mathrm{ext},1},\dots,\phi_{\mathrm{ext},n_\mathrm{L}}]$ as:

$$\phi_{b,k}=\boldsymbol{w}_k^T\boldsymbol{\Phi}+\boldsymbol{b}_k^T\boldsymbol{\Phi}_\mathrm{ext}, \tag{1}$$

where $\boldsymbol{w}_k$ is the n_N dimensional vector with $\delta_{im}-\delta_{mj}$ for element $m\in\{1,2,\cdots,n_\mathrm{N}\}$ and $\boldsymbol{b}_k$ is the n_L dimensional vector that its lth element is one if kth branch is a closure branch for loop l otherwise it is zero.

The node fluxes take the role of position variables. There is an associated "potential" energy due to every inductive element. Every branch with an inductive element has an energy which is an instantaneous function of the branch flux, *i.e.*, $U_{b,k}=U_{b,k}(\phi_{b,k})$. We are primarily concerned with two types of inductors, linear inductors with inductance l where $U_L(\phi_{b,k})=\phi_{b,k}^2/2l$ and Josephson junctions $U_J(\phi_{b,k})=-E_J\cos(2\pi\phi_{b,k}/\Phi_0)$, where $\Phi_0=h/2e$ is the magnetic flux quantum and E_J is the Josephson energy.

For branches which are capacitors, the time-derivative of the branch flux is the voltage drop across the branch $V_{b,k}=\dot{\phi}_{b,k}$; therefore, we assign a "kinetic" energy $T(\dot{\phi}_{b,k})=c\dot{\phi}_{b,k}^2/2$ to any capacitor branch k with a capacitance c.

2.1 Purely Harmonic Circuits

For the case of purely harmonic circuits, *i.e.*, circuits containing only linear inductors and capacitors, the analysis is significantly simplified. As justified below, we need only to consider the inductances and capacitances between each pair of nodes i and j given by l_{ij} and c_{ij}, respectively. Constant external fluxes can be absorbed into the flux variables – the linearity of the circuit means that these shifts do not affect the dynamics. The total potential energy can then always be written as

$$U=\frac{1}{2}\sum_{ij}\frac{(\phi_i-\phi_j)^2}{l_{ij}}, \tag{2}$$

while the total kinetic energy is given by

$$T=\frac{1}{2}\sum_{ij}c_{ij}(\dot{\phi}_i-\dot{\phi}_j)^2, \tag{3}$$

where above summations are over $i,j=1,..,n_\mathrm{N}$. These equations can be conveniently expressed in terms of vectors $\boldsymbol{\Phi}$ and $\dot{\boldsymbol{\Phi}}$, and by defining capacitance $\boldsymbol{C}$ and inverse inductance (susceptance) $\boldsymbol{L}^*$ matrices such that

$$\begin{aligned} U(\boldsymbol{\Phi}) &= \frac{1}{2}\boldsymbol{\Phi}^T\boldsymbol{L}^*\boldsymbol{\Phi}, \\ T(\dot{\boldsymbol{\Phi}}) &= \frac{1}{2}\dot{\boldsymbol{\Phi}}^T\boldsymbol{C}\dot{\boldsymbol{\Phi}}. \end{aligned} \tag{4}$$

The relation between capacitance (susceptance) matrix elements and the capacitances (susceptances) of the branches is then

$$C_{ij}=[\boldsymbol{C}]_{ij}:\begin{cases} C_{ii}=c_{i0}+\sum_{k\neq i}c_{ik} & i=j\\ C_{ij}=-c_{ij} & i\neq j\end{cases}$$

and

$$\frac{1}{L_{ij}}=[\boldsymbol{L}^*]_{ij}:\begin{cases} \frac{1}{L_{ii}}=\frac{1}{l_{i0}}+\sum_{k\neq i}\frac{1}{l_{ik}} & i=j\\ \frac{1}{L_{ij}}=-\frac{1}{l_{ij}} & i\neq j\end{cases}$$

. The Lagrangian $\mathcal{L}(\boldsymbol{\Phi}, \dot{\boldsymbol{\Phi}}) = T(\dot{\boldsymbol{\Phi}}) - U(\boldsymbol{\Phi})$ gives us the dynamics of the system. The Euler-Lagrange equation generated by this Lagrangian for each variable i is effectively Kirchhoff's current law for that node. In contrast, Kirchhoff's voltage law was automatically satisfied when we applied the constraints from Faraday's law on the branch fluxes.

As a prelude to obtaining a quantum theory, we find the Hamiltonian. For this, we first calculate the conjugate momenta or charge variables for each node flux ϕ_i. The equation in vector form is given succinctly as the gradient

$$\boldsymbol{Q} \quad \equiv \quad \frac{\partial L}{\partial \dot{\boldsymbol{\Phi}}} = \boldsymbol{C}\dot{\boldsymbol{\Phi}}. \tag{5}$$

Assuming that the capacitance matrix is invertible, we obtain the circuit Hamiltonian H through the Legendre transformation

$$H = \frac{1}{2}\boldsymbol{\Phi}^T \boldsymbol{L}^* \boldsymbol{\Phi} + \frac{1}{2}\boldsymbol{Q}^T \boldsymbol{C}^{-1}\boldsymbol{Q}. \tag{6}$$

2.2 Circuits with Nonlinear Inductors

The presence of nonlinear inductors, *i.e.*, inductors where $U(\phi_b)$ is not quadratic, allows external fluxes to change the dynamics of circuits fundamentally. The simplest example is the superconducting quantum interference device (SQUID) which contains two Josephson junctions in parallel, forming a loop.

We first express the potential energy of the circuit in terms of the branch fluxes across all inductive elements in $\mathcal{S}_L \cup \mathcal{S}_J$, set of branches consisting of linear inductors $\mathcal{S}_L$ and Josephson junctions $\mathcal{S}_J$:

$$U = \sum_{k \in \mathcal{S}_L \cup \mathcal{S}_J} U_k(\phi_{b,k}). \tag{7}$$

But not all branch fluxes are independent variables. Using Eq. 1, we write the potential energy in terms of node flux variables $\boldsymbol{\Phi}$, and the external fluxes $\boldsymbol{\Phi}_{\text{ext}} = \Phi_0 \boldsymbol{\varphi}_{\text{ext}}/2\pi$:

$$\begin{aligned} U(\boldsymbol{\Phi}) = & \frac{1}{2}\boldsymbol{\Phi}^T \boldsymbol{L}^* \boldsymbol{\Phi} + \sum_{k \in \mathcal{S}_L} \frac{1}{l_k}\left(\frac{\Phi_0}{2\pi}\right) \boldsymbol{w}_k^T \boldsymbol{\Phi} (\boldsymbol{b}_k^T \boldsymbol{\varphi}_{\text{ext}}) \\ & - \sum_{k \in \mathcal{S}_J} E_{J_k} \cos\left(\frac{2\pi}{\Phi_0}\boldsymbol{w}_k^T \boldsymbol{\Phi} + \boldsymbol{b}_k^T \boldsymbol{\varphi}_{\text{ext}}\right) \end{aligned}$$

The kinetic part of the energy is given by summing over the capacitive energy of every capacitor:

$$T = \frac{1}{2}\sum_{k \in \mathcal{S}_C} c_k \dot{\phi}_{b,k}^2.$$

In the case where the external fluxes are time-independent ($\dot{\boldsymbol{\Phi}}_{\text{ext}} = 0$), we find that

$$\dot{\phi}_{b,k} = \boldsymbol{w}_k^T \dot{\boldsymbol{\Phi}},$$

and

$$T(\dot{\boldsymbol{\Phi}}) = \frac{1}{2}\dot{\boldsymbol{\Phi}}^T \boldsymbol{C} \dot{\boldsymbol{\Phi}}.$$

as in the harmonic case.

As noted recently [14, 15], a time-dependent external flux poses a challenge to this description as we end up with terms $\dot{\phi}_i$ in the Lagrangian that depend on how we assign fluxes to branches. Since time-dependent flux is essential in contexts such as flux-noise-induced dephasing [14] and Floquet engineering [22], it is crucial for `SQcircuit` to treat these cases correctly, and we provide details of our approach in Appendix D.

From the Lagrangian description, we can again obtain the conjugate momenta, and through a Legendre transformation, the Hamiltonian, which is now given by:

$$\begin{aligned} H = & \frac{1}{2}\boldsymbol{Q}^T \boldsymbol{C}^{-1} \boldsymbol{Q} + \frac{1}{2}\boldsymbol{\Phi}^T \boldsymbol{L}^* \boldsymbol{\Phi} \\ & + \sum_{k \in \mathcal{S}_L} \left(\frac{\Phi_0}{2\pi} \frac{\boldsymbol{b}_k^T \boldsymbol{\varphi}_{\text{ext}}}{l_k}\right) \boldsymbol{w}_k^T \boldsymbol{\Phi} \\ & - \sum_{k \in \mathcal{S}_J} E_{J_k} \cos\left(\frac{2\pi}{\Phi_0}\boldsymbol{w}_k^T \boldsymbol{\Phi} + \boldsymbol{b}_k^T \boldsymbol{\varphi}_{\text{ext}}\right). \end{aligned} \tag{8}$$

Note that in the absence of Josephson junctions, we can absorb the external fluxes into $\boldsymbol{\Phi}$ and recover the Hamiltonian from the linear analysis. In Appendix A, we show in detail how to construct the Hamiltonian for an example circuit.

Finally, to quantize our circuit, we promote every node flux and its conjugate momentum to operators ($\phi_i \to \hat{\phi}_i$ and so on...) with the canonically conjugate operators satisfying the commutation relation

$$[\hat{\phi}_m, \hat{Q}_n] = \delta_{mn} i\hbar.$$

In matrix form, the commutators are expressed compactly as

$$\hat{\boldsymbol{\Phi}}\hat{\boldsymbol{Q}}^T - \hat{\boldsymbol{Q}}\hat{\boldsymbol{\Phi}}^T = i\hbar\boldsymbol{1},$$

where $\boldsymbol{1}$ is the identity matrix.

3 Coordinate Transformations and Hamiltonian Diagonalization

A major challenge in performing a full quantum analysis of a superconducting circuit is that, in general, increasing the size of the circuit leads to an exponential increase in the numerical resources needed for simulation. Therefore, finding the coordinate transformations in which the Hamiltonian is sparse and falls off quickly from the diagonal is essential. Although we use exact diagonalization to calculate the eigenvalues, our method paves the way for approximate methods such as tensor networks [23] for large Hilbert spaces. To meet these criteria, we are free to transform the Hamiltonian of Eq. 8 by performing a canonical transformation of charge and flux operators:

$$\begin{aligned}\hat{\tilde{\boldsymbol{Q}}} &= \boldsymbol{R}^{-1}\hat{\boldsymbol{Q}},\\ \hat{\tilde{\boldsymbol{\Phi}}} &= \boldsymbol{S}^{-1}\hat{\boldsymbol{\Phi}},\end{aligned}$$

where $\boldsymbol{R}$ and $\boldsymbol{S}$ are $n_\mathrm{N} \times n_\mathrm{N}$ real invertible matrices. For convenience, we impose the requirement that the transformation is canonical, *i.e.*, the transformed charge and flux operators satisfy the relations $[\hat{\tilde{\phi}}_m, \hat{\tilde{Q}}_n] = \delta_{mn} i\hbar$. This leads to the constraint

$$\boldsymbol{S}^T = \boldsymbol{R}^{-1}. \tag{9}$$

The transformed Hamiltonian has the following form:

$$\begin{aligned}\hat{\tilde{H}} =& \frac{1}{2}\hat{\tilde{\boldsymbol{Q}}}^T\tilde{\boldsymbol{C}}^{-1}\hat{\tilde{\boldsymbol{Q}}} + \frac{1}{2}\hat{\tilde{\boldsymbol{\Phi}}}^T\tilde{\boldsymbol{L}}^*\hat{\tilde{\boldsymbol{\Phi}}}\\ &+ \sum_{k\in\mathcal{S}_L}\left(\frac{\Phi_0}{2\pi}\frac{\boldsymbol{b}_k^T\boldsymbol{\varphi}_\mathrm{ext}}{l_k}\right)\tilde{\boldsymbol{w}}_k^T\hat{\tilde{\boldsymbol{\Phi}}}\\ &- \sum_{k\in\mathcal{S}_J} E_{J_k}\cos\left(\frac{2\pi}{\Phi_0}\tilde{\boldsymbol{w}}_k^T\hat{\tilde{\boldsymbol{\Phi}}} + \boldsymbol{b}_k^T\boldsymbol{\varphi}_\mathrm{ext}\right),\end{aligned} \tag{10}$$

where the transformed inverse capacitance and susceptance matrices are:

$$\begin{aligned}\tilde{\boldsymbol{C}}^{-1} &= \boldsymbol{R}^T\boldsymbol{C}^{-1}\boldsymbol{R},\\ \tilde{\boldsymbol{L}}^* &= \boldsymbol{S}^T\boldsymbol{L}^*\boldsymbol{S},\end{aligned}$$

and the vector assigning fluxes to junctions is given by

$$\tilde{\boldsymbol{w}}_k^T = \boldsymbol{w}_k^T\boldsymbol{S}.$$

As explained in Appendix B, we can always find $\boldsymbol{S}$ and $\boldsymbol{R}$ such that $\tilde{\boldsymbol{L}}^*$ and $\tilde{\boldsymbol{C}}$ have the block diagonal form of:

$$\tilde{\boldsymbol{C}} = \begin{bmatrix}\boldsymbol{C}^\mathrm{ha} & \boldsymbol{0}\\ \boldsymbol{0} & \boldsymbol{C}^\mathrm{ch}\end{bmatrix}, \tag{11}$$

$$\tilde{\boldsymbol{L}}^* = \begin{bmatrix}\boldsymbol{L}^{*^\mathrm{ha}} & \boldsymbol{0}\\ \boldsymbol{0} & \boldsymbol{0}\end{bmatrix}, \tag{12}$$

where the "harmonic" parts $\boldsymbol{C}^\mathrm{ha}$ and $\boldsymbol{L}^{*^\mathrm{ha}}$ are $n_\mathrm{H}\times n_\mathrm{H}$ invertible diagonal matrices, while the "charge" part $\boldsymbol{C}^\mathrm{ch}$ is a $n_\mathrm{C}\times n_\mathrm{C}$ symmetric matrix (not necessarily diagonal). The corresponding transformed charge and flux operators are also divided into two parts:

$$\begin{aligned}\hat{\tilde{\boldsymbol{Q}}}^T =& [\hat{\boldsymbol{Q}}^{\mathrm{ha}^T}|\hat{\boldsymbol{Q}}^{\mathrm{ch}^T}]\\ =& [\hat{Q}_1^\mathrm{ha}, \dots, \hat{Q}_{n_\mathrm{H}}^\mathrm{ha}|\hat{Q}_1^\mathrm{ch}, \dots, \hat{Q}_{n_\mathrm{C}}^{ch}],\\ \hat{\tilde{\boldsymbol{\Phi}}}^T =& [\hat{\boldsymbol{\Phi}}^{\mathrm{ha}^T}|\hat{\boldsymbol{\Phi}}^{\mathrm{ch}^T}]\\ =& [\hat{\phi}_1^\mathrm{ha}, \dots, \hat{\phi}_{n_\mathrm{H}}^\mathrm{ha}|\hat{\phi}_1^\mathrm{ch}, \dots, \hat{\phi}_{n_\mathrm{C}}^\mathrm{ch}].\end{aligned}$$

We similarly divide the $\tilde{\boldsymbol{w}}_k^T$ into to two parts:

$$\tilde{\boldsymbol{w}}_k^T = [\boldsymbol{w}_k^{\mathrm{ha}^T}|\boldsymbol{w}_k^{\mathrm{ch}^T}]$$

The transformation that we described above expresses the system's dynamics as that of (1) n_H uncoupled harmonic oscillators, with capacitances and inverse inductances given by the diagonals of $\boldsymbol{C}^\mathrm{ha}$ and $\boldsymbol{L}^{*^\mathrm{ha}}$, (2) n_C superconducting islands with no conducting paths connecting them and charging energy given by the capacitance matrix $\boldsymbol{C}^\mathrm{ch}$, and (3) the flux drop across several junctions as weighted by the vector $\tilde{\boldsymbol{w}}_k^T$. Next we will quantize the harmonic (1) and charge (2) modes separately, and eventually diagonalize and find the spectrum of the full circuit.

Harmonic modes: Ignoring the junctions momentarily, the part of the Hamiltonian acting on the first subspace is given by $\frac{1}{2}\hat{\boldsymbol{Q}}^{\mathrm{ha}^T}(\boldsymbol{C}^\mathrm{ha})^{-1}\hat{\boldsymbol{Q}}^\mathrm{ha} + \frac{1}{2}\hat{\boldsymbol{\Phi}}^{\mathrm{ha}^T}\boldsymbol{L}^{*^\mathrm{ha}}\hat{\boldsymbol{\Phi}}^\mathrm{ha}$. Since $\boldsymbol{C}^\mathrm{ha}$ and $\boldsymbol{L}^{*^\mathrm{ha}}$ are diagonal matrices, we have n_H uncoupled harmonic oscillators with charge and flux variables $\hat{Q}_i^\mathrm{ha}$ and $\hat{\phi}_i^\mathrm{ha}$. Our diagonalization has effectively found the normal mode decomposition of the system. We associate creation and annihilation operators $\hat{a}_i^\dagger$ and $\hat{a}_i$ with each of the harmonic oscillators. The charge and flux operators are then

$$\hat{Q}_i^\mathrm{ha} = i\sqrt{\frac{\hbar}{2Z_i}}(\hat{a}_i^\dagger - \hat{a}_i,) \tag{13}$$

$$\hat{\phi}_i^\mathrm{ha} = \sqrt{\frac{\hbar Z_i}{2}}(\hat{a}_i^\dagger + \hat{a}_i), \tag{14}$$

where Z_i and ω_i are the mode impedance and angular frequency respectively, and given by:

$$Z_i = \sqrt{1/(\boldsymbol{L}_{i,i}^{*^\mathrm{ha}}\boldsymbol{C}_{i,i}^\mathrm{ha})}, \tag{15}$$

$$\omega_i = \sqrt{\boldsymbol{L}_{i,i}^{*^\mathrm{ha}}/\boldsymbol{C}_{i,i}^\mathrm{ha}}. \tag{16}$$

Charge modes: The remaining subspace has a kinetic energy term $\frac{1}{2}\hat{\boldsymbol{Q}}^{\mathrm{ch}^T}(\boldsymbol{C}^{\mathrm{ch}})^{-1}\hat{\boldsymbol{Q}}^{\mathrm{ch}}$, and a potential energy that is only due to junctions. Consider the case where there is only a single charge mode, then the junction energy will have the form $\cos(\alpha\hat{\phi}_1^{\mathrm{ch}}+\hat{O})$, where $\hat{O}$ is an operator that represents the rest of the terms in the cosine. Since we can write the cosine as

$$\cos(\alpha\hat{\phi}_1^{\mathrm{ch}}+\hat{O})=\frac{e^{i\alpha\hat{\phi}_1^{\mathrm{ch}}}e^{i\hat{O}}+e^{-i\alpha\hat{\phi}_1^{\mathrm{ch}}}e^{-i\hat{O}}}{2},$$

and $e^{i\alpha\hat{\phi}_1^{\mathrm{ch}}}$ effects a translation in the charge basis where $e^{\pm i\alpha\hat{\phi}_1^{\mathrm{ch}}}|\hat{Q}_1^{\mathrm{ch}}=Q_1\rangle=|\hat{Q}_1^{\mathrm{ch}}=Q_1\pm\hbar\alpha\rangle$, we see that only a discrete number of charge eigenstates $|\cdots n_1\cdots\rangle\equiv|\cdots(Q_1+n_1\hbar\alpha)\cdots\rangle$ for integer n_1 need to be considered. Moreover, we are free to choose the coefficient α with an appropriate transformation of the basis, *i.e.*, a constant scaling of $\boldsymbol{R}$ and $\boldsymbol{S}$ which keeps the transformation canonical. This constant scaling is effectively equivalent to choosing the units of $\hat{Q}$ and $\hat{\phi}$ while keeping the commutation relation the same. A physical meaningful value for α is $2\pi/\Phi_0$. This choice means that Josephson tunneling changes the charge state by $2e$, and thus couples all charge states of the from $|n_1\rangle\equiv|\hat{Q}_1^{\mathrm{ch}}=Q_1\pm2en_1\rangle$.

In Appendix B.2, we generalize the above approach to systems with more charge modes, where a more careful consideration of the flux periodicity in the circuit is needed to avoid double counting of states. In that case our basis for each charge mode is given by the eigenstates of that charge operator, and rescale the operators such that

$$\hat{Q}_i^{\mathrm{ch}}\,|n_i\rangle=(Q_i+2en_i)\,|n_i\rangle\,. \tag{17}$$

The gate charge Q_i can affect the dynamics, and leads to changes in the energy levels, perhaps most dramatically in the charge limit of the Cooper pair box qubit.

We separate the basis of the transformed Hamiltonian into two subspaces. The first subspace is formed by n_{H} harmonic oscillators and the second subspace is formed by n_{C} charge modes, which is also the number of charge islands. Therefore, the general basis of the circuit is

$$|n_1^{\mathrm{ha}},\ldots,n_{n_{\mathrm{H}}}^{\mathrm{ha}}\rangle\,|n_1^{\mathrm{ch}},\ldots,n_{n_{\mathrm{C}}}^{\mathrm{ch}}\rangle\,, \tag{18}$$

where $n_m^{\mathrm{ha}}=0,1,2,\cdots$ is number of photons in the mth harmonic mode and $n_m^{\mathrm{ch}}=\cdots,-1,0,1,\cdots$ are the number of cooper pair that have tunneled onto the mth charge mode.

We now express the Josephson potential energies in the basis chosen above. Using the fact that different flux operators act on different subspaces, the Josephson energies can be written as

$$\cos\left(\frac{2\pi}{\Phi_0}\tilde{\boldsymbol{w}}_k^T\hat{\tilde{\boldsymbol{\Phi}}}+\boldsymbol{b}_k^T\boldsymbol{\varphi}_{\mathrm{ext}}\right)=$$
$$\frac{1}{2}e^{i\boldsymbol{b}_k^T\boldsymbol{\varphi}_{\mathrm{ext}}}\prod_{m=1}^{n_{\mathrm{H}}}e^{i\frac{2\pi}{\Phi_0}w_{km}^{\mathrm{ha}}\hat{\phi}_m^{\mathrm{ha}}}\prod_{m=1}^{n_{\mathrm{C}}}e^{i\frac{2\pi}{\Phi_0}w_{km}^{\mathrm{ch}}\hat{\phi}_m^{\mathrm{ch}}}+\text{h.c.},$$

where w_{km}^{λ} and $\hat{\phi}_m^{\lambda}$ are the mth element of $\boldsymbol{w}_k^{\lambda}$ and $\hat{\boldsymbol{\Phi}}^{\lambda}$ respectively for $\lambda\in\{\mathrm{ha},\mathrm{ch}\}$. By using annihilation and creation operator representation of $\hat{\phi}_k^{\mathrm{ha}}$, Eq. 14, we can write the exponential term related to harmonic modes as:

$$e^{i\frac{2\pi}{\Phi_0}w_{km}^{\mathrm{ha}}\hat{\phi}_m^{\mathrm{ha}}}=\hat{D}_m(\alpha_{km}), \tag{19}$$

where $\alpha_{km}=i\frac{2\pi}{\Phi_0}w_{km}^{\mathrm{ha}}\sqrt{\frac{\hbar Z_m}{2}}$ and $\hat{D}_m(\alpha_{km})=\exp\left(\alpha_{km}a_m^{\dagger}-\alpha_{km}^{*}a_m\right)$ is bosonic displacement operator for harmonic mode m whose off diagonal elements fall off quickly from the diagonal for small $|\alpha_{km}|\ll1$. The exponential due to the charge mode is

$$e^{i\frac{2\pi}{\Phi_0}w_{km}^{\mathrm{ch}}\hat{\phi}_m^{\mathrm{ch}}}=(\hat{d}_m)^{w_{km}^{\mathrm{ch}}}, \tag{20}$$

where $\hat{d}_m=\sum_{n_m^{\mathrm{ch}}}|n_m^{\mathrm{ch}}+1\rangle\langle n_m^{\mathrm{ch}}|$ is the charge raising operator. Note that as a result of how we performed the circuit transformation, w_{km}^{ch} is either 0, -1 or $+1$. The operator raised to these powers corresponds to the identity, reducing the charge on island m by one Cooper pair, or increasing it by a Cooper pair.

Finally, the transformed Hamiltonian of Eq. 10 in the specified harmonic and charge operators has the following representation:

$$\begin{aligned}\hat{\tilde{H}}=\hbar\sum_{m=1}^{n_{\mathrm{H}}}\omega_m\hat{a}_m^{\dagger}\hat{a}_m+\frac{1}{2}\sum_{mn}(\boldsymbol{C}^{\mathrm{ch}})^{-1}_{mn}\hat{Q}_m^{\mathrm{ch}}\hat{Q}_n^{\mathrm{ch}}\\+\sum_{k\in\mathcal{S}_L}\left(\frac{\Phi_0}{2\pi}\frac{\boldsymbol{b}_k^T\boldsymbol{\varphi}_{\mathrm{ext}}}{l_k}\right)\sum_{m=1}^{n_{\mathrm{H}}}w_{km}^{\mathrm{ha}}\sqrt{\frac{\hbar Z_m}{2}}(\hat{a}_m^{\dagger}+\hat{a}_m)\\-\sum_{k\in\mathcal{S}_J}\frac{E_{J_k}}{2}\Big(e^{i\boldsymbol{b}_k^T\boldsymbol{\varphi}_{\mathrm{ext}}}\prod_{m=1}^{n_{\mathrm{H}}}\hat{D}_m(\alpha_{km})\prod_{m=1}^{n_{\mathrm{C}}}(\hat{d}_m)^{w_{km}^{\mathrm{ch}}}\\+\text{h.c.}\Big).\end{aligned}$$

To represent and diagonalize the Hamiltonian numerically, we must set the truncation number for each harmonic and charge mode. For example, if the truncation number of the second harmonic

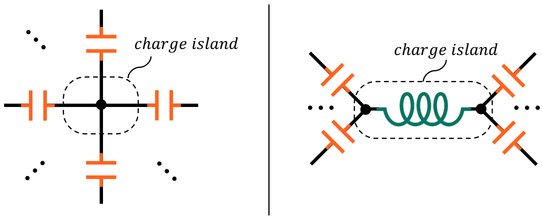


Figure 2: Examples of the conserved charge on the isolated island. Charges cannot escape the island because there is no Josephson junction connected to the island. SQcicuit detects these situations and removes them automatically.

mode is N, the n_2^{ha} in Eq. 18 can take any value from 0 to $N-1$.

There are several key ideas associated to our method for the Hamiltonian diagonalization: (1) for the circuits only consist of linear elements such as capacitors and inductors, we show that proper coordinate transformation and basis selection is sufficient for diagonalizing the Hamiltonian, without a need to solve the eigenvalue problem; (2) for the circuits with nonlinearities, we employ the idea of finding and omitting the decouple modes using the Josephson junction part of the Hamiltonian. This significantly reduces the dimension of the Hilbert space required for computing the spectrum. For example, these decouple modes frequently happen in symmetric circuits such as $0-\pi$ qubit (see Sec. 4 for details). Another example of decoupled modes are conserved charges on isolated islands in superconducting circuits that cannot tunnel through Josephson junctions. These modes appear in the Hamiltonian with terms proportional to $\hat{Q}^2$ and do not have presence in the Josephson junction part of the Hamiltonian (see Fig. 2). Since these charges are conserved, they have no dynamics, and we can remove them from the problem.

4 Circuit Description in SQcircuit

We have developed `SQcircuit`, a Python package that utilizes and automates the methods described in the previous sections to build and diagonalize the circuit Hamiltonian from its circuit description and extract experimentally relevant properties such as transition matrix elements and decay rates. In this section, by using $0-\pi$ qubit [18] shown in Fig. 3a as an example, we demonstrate how to describe a circuit in `SQcircuit` and to calculate its spectrum and eigenfunctions. We begin by importing the `SQcircuit` package:

```python
import SQcircuit as sq
```

In `SQcircuit`, each circuit component has an associated class definition and properties. To define the capacitors, we create an object of `Capacitor` class. For instance, the capacitors in Fig. 3a can be defined by

```python
C = sq.Capacitor(value=0.15, unit="GHz")
CJ = sq.Capacitor(value=10, unit="GHz")
```

In the above lines of the code, the `unit` argument can be either hertz or farad. If the `unit` is in hertz, such as `"GHz"`, `"MHz"`, the `value` argument specifies the charging energy of the capacitor, i.e., $E_c = e^2/2c$. If the `unit` argument is in farad, such as `"nF"`, `"pF"`, etc., the value specifies the capacitance in farad. For example, the previous line of code for the `CJ` capacitor is approximately equivalent to

```python
CJ = sq.Capacitor(value=1.94, unit="fF")
```

since the 1.94 femtofarad capacitor has a charging energy of approximately 10 gigahertz.

Before defining the inductive elements, we need to define the inductive loops (closed path of inductive elements) within which the element resides and through which the flux will be set, and so we define the loops for the circuit. For example, for the $0-\pi$ qubit, we define a single loop by creating an object of `Loop` class as

```python
loop1 = sq.Loop(value=0)
```

where `value` is the external flux value that can be altered later by `set_flux()` method.

The inductor in Fig. 3a can be created as an object from the `Inductor` class:

```python
L = sq.Inductor(value=0.13, unit="GHz",
                loops = [loop1])
```

Like the `Capacitor` class, if the `unit` argument is in hertz, the `value` describes the inductive energy, $E_l = (\Phi_0/2\pi)^2/l$. If the `unit` argument is in henry, such as `"uH"`, `"nH"`, etc., the `value` is the inductance in henry. For inductive elements that are part of a loop, we indicate the loops in which they reside. For example, if the inductive element is part of the two loops, namely `loop1` and `loop2` (both objects of the Loop class), we would pass the arguments of `loops=[loop1, loop2]` to

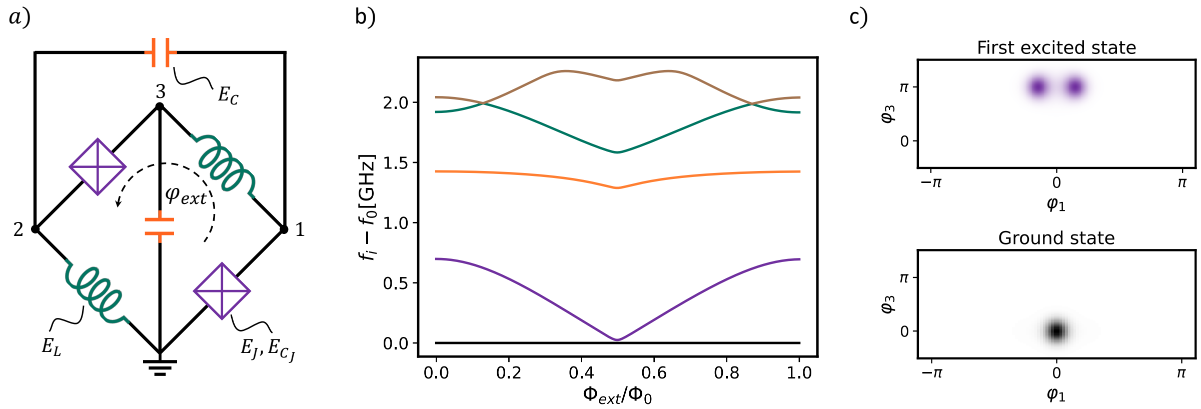


Figure 3: **a)** A symmetric $0-\pi$ qubit with inductive, Josephson, and charging energies in gigahertz unit as follow: $E_L/h = 0.13$ GHz, $E_J/h = 5$ GHz, $E_C/h = 0.15$ GHz, and $E_{C_J}/h = 10$ GHz. **b)** The energy spectrum of the circuit calculated by `diag()` functionality of the `SQcircuit`. **c)** The probability distribution of the ground and first excited states in phase space coordinates, calculated at $\varphi_{\text{ext}} = 0$, by using `eig_phase_coord()` functionality of the `SQcircuit`. Figures are generated by the Python code in Appendix E.

the definition of the inductive element. In our example, the inductor is only part of `loop1`. Thus, we set `loops=[loop1]`. The Josephson junction as an inductive element has an analogous definition to inductors and can be defined by using the `Junction` class. The Josephson junction in Fig.3a can be defined as

```
JJ = sq.Junction(value=5, unit="GHz",
                 loops=[loop1])
```

The default unit for all elements is gigahertz. However, we can change the default unit for each type of element via `sq.set_unit_cap()`, `sq.set_unit_ind()`, and `sq.set_unit_JJ()` functions for capacitors, inductors, and Josephson junctions respectively. For example the default unit of capacitors can be set to `"fF"` with

```
sq.set_unit_cap("fF")
```

After defining all circuit components, to describe the circuit topology in `SQcircuit`, we create an object of `Circuit` class by passing a Python dictionary that contains the list of all elements at each edge. For the circuit of Fig.3a

```
elements = {
    (0, 1): [CJ, JJ],
    (0, 2): [L],
    (0, 3): [C],
    (1, 2): [C],
    (1, 3): [L],
    (2, 3): [CJ, JJ]
}

cr = sq.Circuit(elements)
```

By creating an object of `Circuit` class, `SQcircuit` systematically finds the set of transformations and bases for diagonalization of the Hamiltonian. Before setting the truncation numbers for each mode and diagonalizing the Hamiltonian, we can gain more insight into our circuit by calling the `description()` method. `SQcircuit` prints out the transformed Hamiltonian Eq. 10 and a listing of the modes, whether they are harmonic or charge modes, and the frequency for each harmonic mode as given by Eq. 16. Moreover, it shows the external flux distribution over inductive elements $\boldsymbol{b}_k^T$ and prefactors in the Josephson junction part of the Hamiltonian $\tilde{\boldsymbol{w}}_k^T$. For example, by executing it on the `cr` object:

```
cr.description()
```

$\hat{H} = \omega_1\hat{a}_1^\dagger\hat{a}_1 + E_{C_{22}}(\hat{n}_2 - n_{g_2})^2 - E_{J_1}\cos(\hat{\varphi}_1 + \hat{\varphi}_2 + 0.5\varphi_{ext_1}) - E_{J_2}\cos(\hat{\varphi}_1 - \hat{\varphi}_2 - 0.5\varphi_{ext_1})$

− −

mode 1: harmonic $\hat{\varphi}_1 = \varphi_{zp_1}(\hat{a}_1 + \hat{a}_1^\dagger)$ $\omega_1/2\pi = 3.22$

mode 2: charge $n_{g_2} = 0$

$E_{C_{22}} = 0.3$ $E_{J_1} = 5$ $E_{J_2} = 5$ $\varphi_{zp_1} = 2.49$

− −

The output above indicates that the first mode of the `cr` circuit is a harmonic mode with a natural frequency of 3.22 gigahertz, as indicated by the frequency unit of `SQcircuit`. The second mode is a charge mode. There is a third mode that is not displayed in the Hamiltonian due to `SQcircuit` automatically eliminating decoupled modes. The normalized charge offset for the second mode (n_{g_2}) can be altered through `set_charge_offset()` method as

```
cr.set_charge_offset(mode=2, ng=1)
```

This method is useful for understanding the charge dispersion of the energy spectrum due to random offsets on charge islands [24].

The main step before diagonalizing the circuit is to set the truncation numbers. To do that, we pass a list of truncation numbers for each mode to the `set_trunc_nums()` method. For `cr` circuit:

```
cr.set_trunc_nums([25, 25])
```

Lastly, we use `diag()` to extract a specified number of eigenfrequencies and eigenvectors. We find the first five eigenstates of the `cr` circuit via:

```
efreqs, evecs = cr.diag(n_eig=5)
```

where `n_eig` specifies the number of eigenvalues to output. The lower `n_eig`, the faster `SQcircuit` finds the eigenvalues. `efreqs` is a Numpy array that contains the eigenfrequencies in `SQcircuit` frequency units. `evecs` is a list that includes the eigenvectors, each as a QuTiP object.

To generate the spectrum of the circuit, firstly, we need to change and sweep the external flux of `loop1` by the `set_flux()` method. Then, we need to find the eigenfrequencies of the circuit that correspond to that external flux. The following lines of code find the flux tuning of the spectrum, `spec`, which is a 2D NumPy array where each column contains the eigenfrequencies with respect to the external flux.

```
# array of external fluxes
phi = np.linspace(0, 1, 100)

# number of eigenvalues we aim for
n_eig = 5

# array that contains the spectrum
spec = np.zeros(n_eig, len(phi))

for i in range(len(phi)):

    # set the value of the flux
    loop1.set_flux(phi[i])

    # get the eigenfrequencies
    spec[:, i], _ = cr.diag(n_eig)
```

The results are plotted in Fig. 3b (see Appendix E.) Note that the default unit of frequency in `SQcircuit` is gigahertz. We can change the default unit by calling `sq.set_unit_freq()` method. For example, we can change the unit to megahertz via

```
sq.set_unit_freq("MHz")
```

5 Circuit Features and Properties

We have developed a systematic way to construct a Hamiltonian for a superconducting quantum circuit and diagonalize it in a physically-motivated and efficient basis. However, for many quantum hardware design problems, the spectrum of the circuit is not sufficient. In this section, we extend the theory of Sec. 2 and 3 to extract other essential circuit properties, and describe how to obtain those features using `SQcircuit`.

5.1 Coordinate Representation of Wavefunctions

Since many physical processes involve operators that are local in the flux (or equivalently phase $\varphi = 2\pi/\Phi_0 \times \phi$), it is sometimes informative to plot the wavefunctions in this space. Given an eigenstate of the circuit in basis of Eq. 18

$$|\psi\rangle = \sum_{n_1^{\mathrm{ha}},\dots,n_{n_\mathrm{C}}^{\mathrm{ch}}} C_{n_1^{\mathrm{ha}},\dots,n_{n_\mathrm{C}}^{\mathrm{ch}}} |n_1^{\mathrm{ha}},\dots,n_{n_\mathrm{H}}^{\mathrm{ha}}\rangle\, |n_1^{\mathrm{ch}},\dots,n_{n_\mathrm{C}}^{\mathrm{ch}}\rangle\,,$$

we are interested in finding the inner product in phase coordinates:

$$\langle\varphi_1^{\mathrm{ha}},\dots,\varphi_{n_\mathrm{H}}^{\mathrm{ha}},\varphi_1^{\mathrm{ch}},\dots,\varphi_{N_c}^{\mathrm{ch}}|\psi\rangle = \sum_{n_1^{\mathrm{ha}},\dots,n_{n_\mathrm{C}}^{\mathrm{ch}}} C_{n_1^{\mathrm{ha}},\dots,n_{n_\mathrm{C}}^{\mathrm{ch}}} \langle\varphi_1^{\mathrm{ha}}|n_1^{\mathrm{ha}}\rangle\dots\langle\varphi_{n_\mathrm{H}}^{\mathrm{ha}}|n_{n_\mathrm{H}}^{\mathrm{ha}}\rangle\,\langle\varphi_1^{\mathrm{ch}}|n_1^{\mathrm{ch}}\rangle\dots\langle\varphi_{n_\mathrm{C}}^{\mathrm{ch}}|n_{n_\mathrm{C}}^{\mathrm{ch}}\rangle\,. \tag{21}$$

It suffices to express each harmonic and charge mode eigenstate in phase coordinate basis. In other words, we need to find $\langle\varphi_m^\lambda|n_m^\lambda\rangle$ for $\lambda \in \{\mathrm{ha},\mathrm{ch}\}$. For the harmonic modes we use the phase coordinate representation of Fock states given by

$$\langle\varphi^{\mathrm{ha}}|n\rangle = \frac{1}{\sqrt{\sqrt{\pi}2^n n! x_0}} e^{-(\frac{\Phi_0}{2\pi}\varphi^{\mathrm{ha}})^2/2x_0^2} H_n\left(\frac{\Phi_0\varphi^{\mathrm{ha}}}{2\pi x_0}\right), \tag{22}$$

where $x_0 = \sqrt{\hbar Z_i}$ and H_n are the Hermite polynomials. On the other hand, the phase coordinate representation of the charge mode eigenstates are given by

$$\langle\varphi^{\mathrm{ch}}|n\rangle = \frac{1}{\sqrt{2\pi}}\exp{(in\varphi^{\mathrm{ch}})}. \tag{23}$$

The eigenvectors in the phase coordinate can be obtained by substituting the Eq. 22-23 into Eq. 21.

We implement the above basis transformation in `SQcircuit` within the function `eig_phase_coord()`. For example, we obtain the

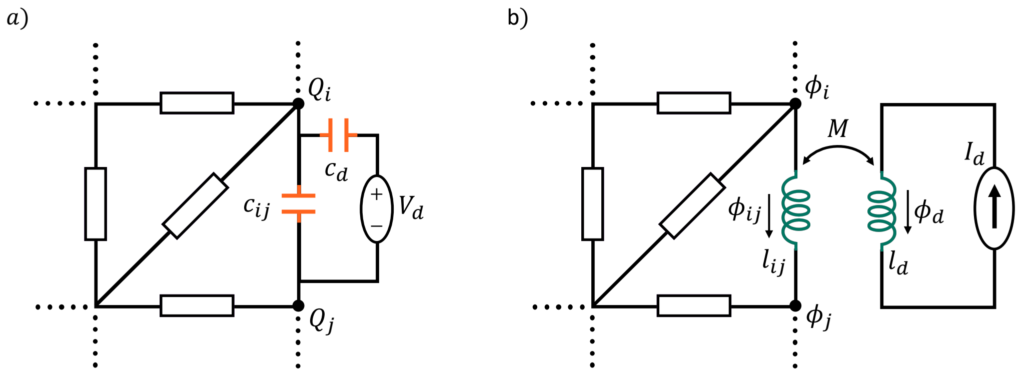


Figure 4: **a)** A drive voltage source coupled with small capacitance of c_d to arbitrary nodes of i and j of a circuit. The Q_i and Q_j are the charge node operators of the node i and j respectively, and C_{ij} is the capacitance between those two nodes. **b)** A current source coupled to an inductor between arbitrary nodes of i and j of a circuit with small mutual inductance of M. φ_{ij} is the phase drop across the inductor L_{ij} between nodes i and j with phase node operators of φ_i and φ_j.

ground state and first excited state of the $0-\pi$ qubit of Sec. 4 at $\varphi_{ext}=0$ by executing the commands:

```python
# create a range for each mode
phi1 = np.pi*np.linspace(-1,1,100)
phi2 = np.pi*np.linspace(-0.5,1.5,100)

# the ground state
state0 = cr.eig_phase_coord(
    k=0,
    grid=[phi1, phi2]
)

# the first excited state
state1 = cr.eig_phase_coord(
    k=1,
    grid=[phi1, phi2]
)
```

Here, `k` is an eigenvector index, which is 0 for the ground state and 1 for the first excited state. The argument `grid` is a list that specifies the values of phase φ to evaluate the wavefunction at, which can be either a single number or a range of values for each mode. We therefore choose a range of values for `phi1` and `phi2`. The resulting phase coordinate representations of the states are `state0` and `state1`, which are 100×100-complex Numpy arrays holding the values of the eigenfunction evaluated on a grid corresponding to `phi1` and `phi2`. In Fig. 3c we plot $|\langle\varphi_1,\varphi_2|\psi\rangle|^2$ for the ground state `state0` and the first excited state `state1` (see Appendix E for more details).

5.2 Coupling Operators and Matrix Elements

Having shown how to analyze an isolated circuit, we now move to describing our approach for understanding the interaction of a circuit with the outside world. More specifically, we aim to calculate the rates at which transitions occur between energy eigenstates under the effect of different driving mechanisms, such as the capacitive and inductive driving via control electronics. The same calculations also apply to estimate the effect of vacuum noise or other random fluctuations driving the system, enabling us to estimate quantities such as the qubit lifetime.

Capacitive coupling: As shown in Fig. 4a, we can capacitively couple a voltage source to arbitrary nodes of i and j of a circuit. If the coupling capacitor, c_d, is small enough that the coupling elements do not change the spectrum of the primary circuit, the interaction Hamiltonian between the primary circuit and drive circuit will be (Appendix C.1)

$$\hat{H}_c^{\mathrm{dr}} = (c_d V_d)\boldsymbol{e}_{ij}^T\boldsymbol{C}^{-1}\boldsymbol{R}\hat{\hat{\boldsymbol{Q}}},$$

where V_d is a voltage of a source $\boldsymbol{e}_{ij}$ is the n_{N} dimensional vector with $\delta_{ik}-\delta_{kj}$ for element $k\in\{1,2,\cdots,n_{\mathrm{N}}\}$. We therefore define the capacitive coupling operator as

$$\hat{O}_c \equiv \boldsymbol{e}_{ij}^T\boldsymbol{C}^{-1}\boldsymbol{R}\hat{\hat{\boldsymbol{Q}}}. \tag{24}$$

Inductive coupling: The inductive coupling case, shown in Fig. 4b, is essentially a dual of the capacitive case and can be analyzed very similarly. Given an inductor l_{ij} in the circuit, we introduce a mutual coupling M which is small enough to have negligible effect on the main circuit's spectrum. The effect of the coupling is then captured by an interaction Hamiltonian (Appendix C.2)

$$\hat{H}_l^{\mathrm{dr}} = (MI_d)\frac{1}{l_{ij}}\boldsymbol{e}_{ij}^T\boldsymbol{S}\hat{\tilde{\boldsymbol{\Phi}}},$$

where I_d is the source current. The inductive coupling operator is defined as

$$\hat{\mathcal{O}}_l \equiv \frac{1}{l_{ij}}\boldsymbol{e}_{ij}^T\boldsymbol{S}\hat{\tilde{\boldsymbol{\Phi}}}. \tag{25}$$

We now use the coupling operators defined above to estimate the transition rates between states. We first calculate the transition matrix element of each type of coupling operator between the energy eigenstates $|m\rangle$ and $|n\rangle$:

$$\langle m|\,\hat{\mathcal{O}}_\lambda\,|n\rangle\,, \tag{26}$$

where $\lambda \in \{c, l\}$. In `SQcircuit`, the `coupling_op()` method returns the coupling operator for each type of coupling. For example, to obtain $\hat{\mathcal{O}}_c$ as a QuTiP object in the constructed Fock/charge basis, we issue the command:

```
O_c = cr.coupling_op(ctype="capacitive",
                     nodes=(i, j))
```

The argument `ctype` sets the type of coupling to either `"capacitive"` or `"inductive"`, while the `nodes` argument is a Python tuple containing the nodes that couple to the drive circuit. Sometimes we are interested only in a specific element of the transition matrix element (Eq. 26) as opposed to the whole matrix. For this we use the `matrix_elements()` method:

```
g_mn = cr.matrix_elements(ctype="capacitive",
                          nodes=(i, j),
                          states=(m, n))
```

where the states of interest are passed through the `states` argument.

5.3 Estimating Decay and Dephasing Rates

We need to understand the effects of imperfect components and fluctuating parameters on the resulting dynamics in the quantum description of superconducting circuits. In this section, we show how `SQcircuit` can take properties – such as the dissipation present in the materials within capacitors, losses in inductors, the presence of quasiparticles in junctions, parasitic coupling to output channels, and fluctuations in parameters – and convert them into dissipation rates and jump operators that enter a master equation description of the open quantum system's dynamics.

5.3.1 Depolarization

Consider a circuit that is coupled to a degree of freedom of the environment $\hat{\xi}$ with the interaction Hamiltonian

$$H_{\mathrm{int}} = \hat{\mathcal{O}}\hat{\xi}.$$

The operator $\hat{\mathcal{O}}$ is in the circuit Hilbert space, while $\hat{\xi}$ is an operator acting on the environment. Within the context of linear response theory [25], we find the decay rate

$$\Gamma_{m\rightarrow n} = \frac{1}{\hbar^2}|\,\langle m|\,\hat{\mathcal{O}}\,|n\rangle\,|^2 S_{\xi\xi}(\omega_{mn}), \tag{27}$$

from level m to n. The transition frequency $\omega_{mn} = \omega_m - \omega_n$ is positive when the circuit emits energy to the environment and negative when circuit absorbs energy from it. The spectral density function (SDF) $S_{\xi\xi}(\omega_{mn})$ is defined as

$$S_{\xi\xi}(\omega) = \int_{-\infty}^{\infty} d\tau\,\langle\xi(\tau)\xi(0)\rangle\,e^{-i\omega\tau},$$

and satisfies the detailed balance relation [25] of

$$\frac{S_{\xi\xi}(\omega)}{S_{\xi\xi}(-\omega)} = \exp\left(\frac{\hbar\omega}{k_BT}\right),$$

where k_B is the Boltzmann's constant and T is the bath temperature. In `SQcircuit`, the default bath temperature is 0.015 kelvin, and it can be changed it by

```
cr.set_temp(T=0.010)
```

where `cr` is an object of the `Circuit` class.

To make quantitative predictions, we need to specify both operator $\hat{\mathcal{O}}$ and the SDF $S_{\xi\xi}(\omega)$. `SQcircuit` includes three common loss channels below that are currently considered to be significant in state-of-the-art circuits.

Capacitive Loss: Dissipation in the capacitor dielectric can be modeled by a capacitor connected to a bath with voltage operator $\hat{V}$. For the capacitance c_{ij} between nodes of i and j, the circuit operator is

$$\hat{\mathcal{O}} = c_{ij}\boldsymbol{e}_{ij}^T\boldsymbol{C}^{-1}\boldsymbol{R}\hat{\tilde{\boldsymbol{Q}}},$$

with spectral density function of

$$S_{VV}(\omega) = \frac{\hbar}{c_{ij}Q_{\mathrm{cap}}(\omega)}\left(1 + \coth\frac{\hbar\omega}{2k_BT}\right),$$

where $Q_{\mathrm{cap}}(\omega)$ is a quality factor which may be weakly frequency dependent. To calculate $\Gamma_{m\rightarrow n}$,

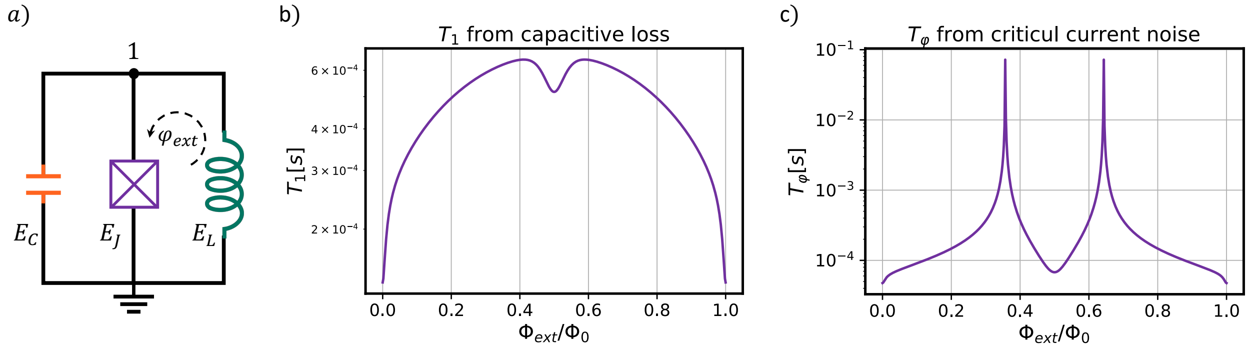


Figure 5: **a)** A Fluxonium qubit with inductive, Josephson, and charging energies in gigahertz unit as follow: $E_L/h = 0.46$ GHz, $E_J/h = 10.2$ GHz, and $E_C/h = 3.6$ GHz. **b,c)** T_1 and T_φ due to capacitive loss and critical current noise of a Fluxonium with $Q_{cap} = 10^6$ and $A_{cc}/E_J = 5\times10^{-7}$ is plotted as a function of external flux by using `dec_rate()` functionality of the `SQcircuit`. Appendix E contains the Python code that produces these figures.

the SDF $S_{VV}(\omega)$ is evaluated at the relevant transition frequency ω_{mn}. In `SQcircuit` the default capacitor quality factor is [26]

$$Q_{\text{cap}}(\omega) = 10^6\left(\frac{2\pi\times 6\text{GHz}}{|\omega|}\right)^{0.7}, \tag{28}$$

but this can be set independently for each capacitor in the circuit, either as a scalar value or a function of ω. For example, to set $Q_{\text{cap}}(\omega) = 2\times10^6$ for the capacitor of the fluxonium in Fig.5a, we define the capacitor with

```
C = sq.Capacitor(3.6, "GHz", Q=2e6)
```

Inductive Loss: This is a loss due to quasiparticle tunneling across Josephson junctions of the superinductors and highly depends on their design. To model this loss, we can assume that the inductor between i and j nodes is coupled to a bath with current $\hat{I}$. The circuit operator $\hat{O}$, hence, is equivalent to

$$\hat{\mathcal{O}} = \boldsymbol{e}_{ij}^T\boldsymbol{S}\hat{\tilde{\boldsymbol{\Phi}}},$$

and

$$S_{II}(\omega) = \frac{\hbar}{l_{ij}Q_{\text{ind}}(\omega)}\left(1+\coth\frac{\hbar\omega}{2k_BT}\right),$$

where $Q_{\text{ind}}(\omega)$ is a frequency-dependent quality factor, with the following default value in `SQcircuit` [27]:

$$Q_{\text{ind}}(\omega) = 500\times10^6\frac{K_0(\frac{h\times0.5\text{GHz}}{2k_BT})\sinh(\frac{h\times0.5\text{GHz}}{2k_BT})}{K_0(\frac{\hbar|\omega|}{2k_BT})\sinh(\frac{\hbar|\omega|}{2k_BT})},$$

in which K_0 is the Bessel function of the second kind. More generally, $Q_{\text{ind}}(\omega)$ can be set to any desired value or function, and for the Fluxonium inductor in Fig.5a with $Q_{\text{ind}} = 300\times10^6$, we could define

```
L = sq.Inductor(0.46, "nH", Q=300e6)
```

Quasiparticle Loss: This loss occurs due to quasiparticle tunnelling in each Josephson junction. Based on [28], the circuit operator for Josephson junction at the edge k is

$$\hat{\mathcal{O}} = \sin\left(\frac{1}{2}\left(\frac{2\pi}{\Phi_0}\tilde{\boldsymbol{w}}_k^T\hat{\tilde{\boldsymbol{\Phi}}} - \boldsymbol{b}_k^T\boldsymbol{\varphi}_{\text{ext}}\right)\right)[1], \tag{29}$$

and SDF is

$$S_{\text{qp}}(\omega) = \hbar\omega\text{Re}[Y_{\text{qp}}(\omega)]\left(1+\coth\frac{\hbar\omega}{2k_BT}\right),$$

where real part of admittance is

$$\text{Re}[Y_{\text{qp}}(\omega)] = \sqrt{\frac{2}{\pi}}\frac{8E_{J_k}}{R_{\text{K}}\Delta}\left(\frac{2\Delta}{\hbar\omega}\right)^{3/2}x_{\text{qp}}\sqrt{\frac{\hbar|\omega|}{2k_BT}} \times K_0\left(\frac{\hbar|\omega|}{2k_BT}\right)\sinh\left(\frac{\hbar|\omega|}{2k_BT}\right),$$

in which $R_{\text{K}} = h/e^2$ is the von Klitzing constant, Δ is the superconducting gap, and x_{qp} is the quasiparticle density. `SQcircuit` has default value of $\Delta = 3.4\times10^{-4}$ eV and $x_{\text{qp}} = 3\times10^{-6}$, but these values can be altered when defining the circuit Josephson junctions. For the Fluxonium in Fig.5a, we change Δ and x_{qp} to different values via

[1] To calculate the matrix element of $\hat{\mathcal{O}}$ in the charge basis, we need to transform the Cooper pair charge basis to the single electron charge basis, similar to the approach in [13].

```
JJ = sq.Junction(10.2, "GHz",
                 delta=2.5e-4, x=8e-6)
```

Finally, after defining all lossy elements and diagonalizing the circuit, the decay rate of Eq. 27 can be calculated with

```
Gamma_mn = cr.dec_rate(dec_type=dec_type,
                       states=(m, n),
                       total=total)
```

In the above code, `dec_type` argument specifies the decay types and can be either `"capacitive"`, `"inductive"`, or `"quasiparticle"`. If `total=False`, `SQcircuit` calculates only the downward decay rate $\Gamma_{m\to n}$ if $m > n$, or $\Gamma_{n\to m}$ if $n > m$. If `total=True` which is the default value, `SQcircuit` returns the summation of both downward and upward transitions, that is $\frac{1}{T_1} = \Gamma_{m\to n} + \Gamma_{n\to m}$. As an example, Fig.5b plots T_1 of the Fluxonium qubit due to capacitive loss as a function of external fluxes. The Python code that generates this plot is provided in Appendix E.

5.3.2 Dephasing

Besides causing transitions between different eigenstates, noise processes also lead to dephasing errors. More specifically, if the transition frequency $\omega_{mn}(\lambda)$ depends on some external parameter $\lambda = \lambda_0 + \delta\lambda(t)$ where λ_0 is the desired value and $\delta\lambda(t)$ is some noise process with $\langle\delta\lambda(t)\rangle = 0$, then information about the relative phase between the mth and nth eigenstates is degraded due to fluctuations in λ. Superconducting circuits are mainly affected by $1/f$ noise such as flux, charge, and critical current noise. The SDF for the $1/f$ noises is given by

$$S_{\lambda\lambda}(\omega) = \frac{2\pi A_\lambda^2}{|\omega|},$$

where A_λ is the noise amplitude. The pure dephasing time and its rate between mth and nth eigenstates is given by [18, 29]

$$\frac{1}{T_\phi} = \kappa_{mn} = \Big(2A_\lambda^2(\frac{\partial\omega_{mn}}{\partial\lambda})^2|\ln\omega_{\text{low}}t_{\text{exp}}| + 2A_\lambda^4(\frac{\partial^2\omega_{mn}}{\partial\lambda^2})^2(\ln^2\frac{\omega_{\text{hi}}}{\omega_{\text{low}}} + 2\ln^2\omega_{\text{low}}t_{\text{exp}})\Big)^{1/2}, \tag{30}$$

where ω_{low} and ω_{hi} are the low-frequency and high-frequency cutoff and t_{exp} is the measurement time. The default value for those parameters in `SQcircuit` are as follow: $\omega_{\text{low}} = 1\text{Hz}$, $\omega_{\text{hi}} = 3.0\text{GHz}$, and $t_{\text{exp}} = 10\mu s$ which can be changed by

```
cr.set_low_freq(value, unit)
cr.set_high_freq(value, unit)
cr.set_t_exp(value, unit)
```

where `cr` is an object of `Circuit` class. Below we give details on how `SQcircuit` treats the three major $1/f$ noise channels that contribute to dephasing.

Critical current noise: This noise is due to fluctuations in the Josephson energy. In `SQcircuit`, the default value for the normalized noise amplitude is $A \equiv A_J/E_J = 10^{-7}$, and a different value such as $A = 5 \times 10^{-7}$ can be specified in the junction definition (Fig.5a) with

```
JJ = sq.Junction(10.2, "GHz", x=8e-6,
                 delta=2.5e-4, A=5e-7)
```

Charge noise: This noise is due to randomness of the charge in the circuit charge islands. `SQcircuit` has the default normalized noise amplitude of $A \equiv A_{n_g}/e = 10^{-4}$, which can be modified for each charge mode of the circuit via `set_charge_noise()` method

```
cr.set_charge_noise(mode=1, A=2e-4)
```

Here `mode` argument specifies the index of the charge mode. To see which circuit mode is a charge mode, one can use `description()` method explained in Sec. 4.

Flux noise: This noise is due to the fluctuations of the external fluxes coupled to inductive loops of the circuit. $A \equiv A_{\varphi_{\text{ext}}}/2\pi = 10^{-6}$ is the default value for the normalized noise amplitude and can be altered in inductive loop definition with

```
loop1 = sq.Loop(value=0, A=5e-6)
```

As discussed in [14, 15] and Appendix D, it is vital to correctly specify the distribution of the external fluxes over inductive elements to properly calculate the dephasing rate for the circuit. This requires knowing the capacitor associated with each inductive elements, which can be specified via the `cap` argument in the inductor and junction definitions in `SQcircuit`. It also requires to pass `flux_dist="all"` option to `Circuit()` class definition. For the Fluxonium in Fig.5a, we could define the circuit with following codes:

```python
# user—defined flux noise
loop1 = sq.Loop(value=0, A=5e—6)

C = sq.Capacitor(3.6, "GHz", Q=1e6)
L = sq.Inductor(0.46, "GHz", loops=[loop1])

# assign C to JJ
JJ = sq.Junction(10.2, "GHz", cap=C,
                 A=5e—7, loops=[loop1])

# C is inside the JJ
elements = {(0, 1): [L, JJ]}

# define the Fluxonium
cr = sq.Circuit(elements, flux_dist="all")
```

where we assumed that the inductor has a tiny capacitor and the Josephson junction has `C` capacitor. Notice that since we have assigned `C` to `JJ`, there is no need to add `C` to the edge list of `elements`.

Finally, to calculate the dephasing rate of Eq. 30, we again use the `dec_rate()` method:

```python
kappa_mn = cr.dec_rate(dec_type=dec_type,
                       states=(m, n))
```

where `dec_type` argument can be either, `"cc"`, `"flux"`, or `"charge"`. In Fig.5c, we plot the dephasing time T_ϕ due to the critical current noise of the Fluxonium qubit with the Python code in Appendix E.

6 Online presence

We published `SQcircuit` as an open source Python package under BSD-3 license, which discloses that it is free to use and to be developed. The main source code can be found under the [30] GitHub repository, which provides the link to download and install the software. In addition, we prepared numerous examples that reproduce the result of the state of the art superconducting circuits in the literature to show the efficiency and convenience of the `SQcircuit` usage. We also provided an online documentation over [31] for detailed information and tutorial of the `SQcircuit`. The Sphinx code that generate this documentation is made available on different GitHub repository of [32]. The list of the online links related to GitHub pages, website, documentation, and Anaconda and PyPi package index repository pages are summarized at table 1. We welcome any feedback regarding bugs/comments/improvements to the `SQcircuit` software packages as well as any initial pull request on the corresponding GitHub pages. If you find this useful, you are encouraged to cite this paper, where the framework has been developed.

Table 1: Online links related to website, GitHub pages, as well as package index repository pages of `SQcircuit`.

SQcircuit website: `sqcircuit.org`
SQcircuit online documentation: `docs.sqcircuit.org`
GitHub repository for SQcircuit source code: `github.com/stanfordLINQS/SQcircuit`
GitHub repository for Sphinx source code: `github.com/stanfordLINQS/SQcircuit-doc`
GitHub repository for Jupyter notebook examples: `github.com/stanfordLINQS/SQcircuit-examples`
Anaconda package repository page: `anaconda.org/conda-forge/sqcircuit`
PyPi package repository page: `pypi.org/project/SQcircuit/`

7 Conclusions and Future Works

In this work, we propose a general approach for analyzing any arbitrary superconducting circuit. We theoretically develop a systematic framework that models the Hamiltonian of the circuit. Then, we show a proper transformation of the coordinates that efficiently diagonalize the Hamiltonian to model its quantum behavior. We also extend our frameworks to capture other important features of the circuits such as lifetime, matrix elements, coupling operators, phase space representation of eigenfunctions, etc. Numerically, we introduce the `SQcircuit` library: an open-source Python library enabling the simulation and analysis of any superconducting circuit. We explain its functionalities to diagonalize the circuit Hamiltonian and to obtain the circuit features such as lifetime, matrix elements. Future work will aim to broaden the scope of the `SQcircuit` by adding more loss channels, creating a framework for optimization and machine learning approaches to discover and design sophisticated superconducting circuits, and including new functionalities to build a composite circuit created out of established superconducting circuits.

Note added

During preparation of this manuscript, we were made aware of ongoing work in Jens Koch's group with a similar scope. Our two groups did not discuss details of our projects, but agreed to post preprints on the arXiv simultaneously.

Acknowledgments

A.H.S.-N. is indebted to Dr. Marek Pechal for invaluable discussions. We acknowledge funding by Amazon Web Services Inc. and by the U.S. government through the Office of Naval Research (ONR) under grant No. N00014-20-1-2422 and the National Science Foundation CAREER award No. ECCS-1941826. The authors wish to thank NTT Research for their financial and technical support and also acknowledge support by Keysight Inc. through the SystemX program at Stanford.

A Example for Circuit Hamiltonian

To elaborate upon the theory discussed in Sec. 2, we construct the Hamiltonian for the circuit shown in Fig. 6. To obtain the Hamiltonian of the form shown in Eq. 8, we specify $\boldsymbol{C}$, $\boldsymbol{L}^*$, $\boldsymbol{w}_k$, and $\boldsymbol{b}_k$. To construct the capacitance matrix, we set the diagonals elements to be the sum of the capacitance connected to a node and the off-diagonal elements to be negative the sum of capacitance between two nodes:

$$\boldsymbol{C} = \begin{bmatrix} c_1 + c_{J_2} + c_{J_3} & 0 & -c_{J_3} \\ 0 & c_2 + c_{J_1} + c_{J_4} & -c_2 - c_{J_4} \\ -c_{J_3} & -c_2 - c_{J_4} & c_2 + c_{J_3} \end{bmatrix}.$$

We construct the inverse inductance or susceptance matrix $\boldsymbol{L}^*$ similarly, by setting diagonal elements to be the sum of the susceptance connected to a node and off-diagonal elements to be negative the sum of susceptance between two nodes:

$$\boldsymbol{L}^* = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & \frac{1}{l} \end{bmatrix}.$$

To calculate $\boldsymbol{w}_k$ and $\boldsymbol{b}_k$ defined in Eq. 1, we find a relation between the kth inductive branch flux, node fluxes, and external fluxes by specifying a minimum spanning tree. In Fig. 6b the purple solid lines specify the chosen spanning tree and black dashed lines specify closure branches. The index of each branch is also specified; thus, the Josephson junction index set is $\mathcal{S}_J = \{1, 2, 3, 4\}$ and linear inductor index set is $\mathcal{S}_L = \{5\}$. The Josephson junction on branch $k = 1$ is part of the spanning tree, so its branch flux includes no offset from external fluxes:

$$\phi_{b,1} = (\phi_1 - 0) + 0 \times \phi_{\text{ext},1} + 0 \times \phi_{\text{ext},2}.$$

Hence,

$$\boldsymbol{w}_1^T = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}, \quad \boldsymbol{b}_1^T = \begin{bmatrix} 0 & 0 \end{bmatrix}.$$

In contrast, the Josephson junction on branch $k = 3$ is not part of the spanning tree, but is on a closure branch that encloses $\phi_{\text{ext},2}$. Therefore the branch flux depends on $\phi_{\text{ext},2}$ as,

$$\phi_{b,1} = (\phi_1 - \phi_3) + 0 \times \phi_{\text{ext},1} + 1 \times \phi_{\text{ext},2},$$

which specifies $\boldsymbol{w}_3$ and $\boldsymbol{b}_3$:

$$\boldsymbol{w}_3^T = \begin{bmatrix} 1 & 0 & -1 \end{bmatrix}, \quad \boldsymbol{b}_3^T = \begin{bmatrix} 0 & 1 \end{bmatrix}.$$

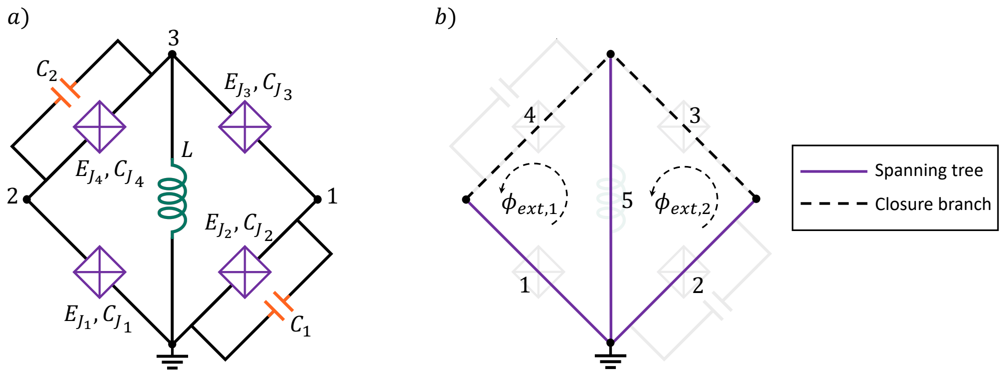


Figure 6: **a)** An arbitrary superconducting circuit with two inductive loops. **b)** The spanning tree of the circuit is specified by purple solid lines and closure branches are the black dashed lines. Branch numbers are specified next to each line.

Using the same approach, the $\boldsymbol{w}_k$ and $\boldsymbol{b}_k$ for other branches are

$$\begin{aligned} \boldsymbol{w}_2^T &= \begin{bmatrix} 0 & 1 & 0 \end{bmatrix}, & \boldsymbol{b}_2^T &= \begin{bmatrix} 0 & 0 \end{bmatrix}, \\ \boldsymbol{w}_4^T &= \begin{bmatrix} 0 & 1 & -1 \end{bmatrix}, & \boldsymbol{b}_4^T &= \begin{bmatrix} 1 & 0 \end{bmatrix}, \\ \boldsymbol{w}_5^T &= \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}, & \boldsymbol{b}_5^T &= \begin{bmatrix} 0 & 0 \end{bmatrix}. \end{aligned}$$

We note that this approach for specifying $\boldsymbol{b}_k$ via spanning tree is complete only for time-independent external fluxes [14, 15]. A generalized description including time-dependent external fluxes is provided in Appendix D.

B Coordinate Transformations

In this section we systematically construct transformed capacitance and inverse inductance matrices as described in Sec. 3, to obtain the form in Eqs. 11 and 12. The transformation occurs in two steps. In the first step we diagonalize the LC part of the Hamiltonian and separate the Hilbert space into two subspaces of harmonic and charge modes. In the second step we transform only the charge-mode subspace to simplify the boundary conditions and representation of the charge modes. Thus, we write the $\boldsymbol{S}$ and $\boldsymbol{R}$ matrices as,

$$\begin{aligned} \boldsymbol{R} &= \boldsymbol{R}_1 \boldsymbol{R}_2, \\ \boldsymbol{S} &= \boldsymbol{S}_1 \boldsymbol{S}_2, \end{aligned}$$

where $\boldsymbol{R}_\lambda$ and $\boldsymbol{S}_\lambda$ specify the λth transformation.

B.1 First Transformation

We write the first transformation as:

$$\boldsymbol{C}_1^{-1} = \boldsymbol{R}_1^T \boldsymbol{C}^{-1} \boldsymbol{R}_1, \tag{31}$$

$$\boldsymbol{L}_1^* = \boldsymbol{S}_1^T \boldsymbol{L}^* \boldsymbol{S}_1. \tag{32}$$

The goal is to find the $\boldsymbol{R}_1$ and $\boldsymbol{S}_1$, such that $\boldsymbol{C}_1$ and $\boldsymbol{L}_1^*$ are diagonal. By taking the inverse of Eq. 31 and imposing the transformation is canonical, *i.e.*, $\boldsymbol{R}_1^T = \boldsymbol{S}_1^{-1}$, we obtain:

$$\begin{aligned} \boldsymbol{C}_1 &= \boldsymbol{S}_1^T \boldsymbol{C} \boldsymbol{S}_1, \\ \boldsymbol{L}_1^* &= \boldsymbol{S}_1^T \boldsymbol{L}^* \boldsymbol{S}_1. \end{aligned} \tag{33}$$

Therefore, the problem reduces to finding the matrix $\boldsymbol{S}_1$ such that $\boldsymbol{C}_1$ and $\boldsymbol{L}_1^*$ are diagonal. Because ($\boldsymbol{C}$, $\boldsymbol{L}^*$) are symmetric and positive-semidefinite, we can take their square root and rewrite $\boldsymbol{C}$ and $\boldsymbol{L}^*$ as [33]

$$\begin{aligned} \boldsymbol{C} &= \sqrt{\boldsymbol{C}}^T \sqrt{\boldsymbol{C}}, \\ \boldsymbol{L}^* &= \sqrt{\boldsymbol{L}^*}^T \sqrt{\boldsymbol{L}^*}. \end{aligned}$$

By substituting in Eq. 33 we find

$$\begin{aligned} \boldsymbol{C}_1 &= \boldsymbol{S}_1^T \sqrt{\boldsymbol{C}}^T \sqrt{\boldsymbol{C}} \boldsymbol{S}_1 = \boldsymbol{F}_c^T \boldsymbol{F}_c, \\ \boldsymbol{L}_1^* &= \boldsymbol{S}_1^T \sqrt{\boldsymbol{L}^*}^T \sqrt{\boldsymbol{L}^*} \boldsymbol{S}_1 = \boldsymbol{F}_l^T \boldsymbol{F}_l, \end{aligned} \tag{34}$$

where

$$\begin{aligned} \boldsymbol{F}_c &= \sqrt{\boldsymbol{C}} \boldsymbol{S}_1, \\ \boldsymbol{F}_l &= \sqrt{\boldsymbol{L}^*} \boldsymbol{S}_1. \end{aligned} \tag{35}$$

Because $\boldsymbol{C}_1$ and $\boldsymbol{L}_1^*$ are assumed to be diagonal, Eq. 34 implies that

$$\begin{aligned} \boldsymbol{F}_c &= \boldsymbol{U} \boldsymbol{D}_c, \\ \boldsymbol{F}_l &= \boldsymbol{V} \boldsymbol{D}_l, \end{aligned} \tag{36}$$

where $\boldsymbol{D}_c$ and $\boldsymbol{D}_l$ are diagonal matrices and $\boldsymbol{U}$ and $\boldsymbol{V}$ are orthogonal matrices. By combining Eq. 35 and Eq. 36, we have:

$$\sqrt{\boldsymbol{C}}\boldsymbol{S}_1 = \boldsymbol{U}\boldsymbol{D}_c, \tag{37}$$

$$\sqrt{\boldsymbol{L}^*}\boldsymbol{S}_1 = \boldsymbol{V}\boldsymbol{D}_l. \tag{38}$$

Because $\boldsymbol{C}$ is invertible, we can find $\boldsymbol{S}_1$ by multiplying the left side of Eq. 37 by $\sqrt{\boldsymbol{C}}^{-1}$:

$$\boldsymbol{S}_1 = \sqrt{\boldsymbol{C}}^{-1}\boldsymbol{U}\boldsymbol{D}_c \tag{39}$$

Substituting Eq. 39 in Eq. 38:

$$\sqrt{\boldsymbol{L}^*}\sqrt{\boldsymbol{C}}^{-1} = \boldsymbol{V}\boldsymbol{D}_l\boldsymbol{D}_c^{-1}\boldsymbol{U}^T = \boldsymbol{V}\boldsymbol{D}\boldsymbol{U}^T, \tag{40}$$

where

$$\boldsymbol{D} = \boldsymbol{D}_l\boldsymbol{D}_c^{-1}. \tag{41}$$

The right side of Eq. 40 has the form of a singular value decomposition (SVD). Hence, we can specify $\boldsymbol{S}_1$ from $\boldsymbol{U}$ and $\boldsymbol{D}_c$ matrices. We are free to choose $\boldsymbol{D}_c$ and $\boldsymbol{D}_l$ as long as they satisfy Eq. 41. However, the zero diagonal entries of $\boldsymbol{D}$ imply that the corresponding diagonal entries of $\boldsymbol{D}_l$ should also be zero because $\boldsymbol{D}_c$ is invertible and cannot have zero diagonal entries. In addition, permuting the $\boldsymbol{D}$ diagonal entries can create another acceptable form of SVD. We sort the $\boldsymbol{D}$ diagonal entries to place the zero entries at the last indices, defining a 2 × 2 block matrix partition in which we call the first n_H non-zero diagonal elements the “harmonic” block and the last n_C zero elements the “charge” block. Thus, $\boldsymbol{C}_1$ and $\boldsymbol{L}_1^*$ have the following form:

$$\boldsymbol{C}_1 = \boldsymbol{D}_c^2 = \begin{bmatrix} \boldsymbol{C}^{\mathrm{ha}_1} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{C}^{\mathrm{ch}_1} \end{bmatrix},$$

$$\boldsymbol{L}_1^* = \boldsymbol{D}_l^2 = \begin{bmatrix} \boldsymbol{L}^{*\mathrm{ha}_1} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} \end{bmatrix},$$

where $\boldsymbol{C}_1^{\mathrm{ha}}$ and $\boldsymbol{L}^{*\mathrm{ha}}_1$ are $n_\mathrm{H} \times n_\mathrm{H}$ diagonal matrices and $\boldsymbol{C}^{\mathrm{ch}_1}$ is a $n_\mathrm{C} \times n_\mathrm{C}$ diagonal matrix.

B.2 Second Transformation

Since the elements of $\boldsymbol{L}_1^*$ are zero for the charge block, there is no quadratic potential for this subspace. Instead, the Josephson junction cosine potentials form a n_C-dimensional periodic potential:

$$U(\boldsymbol{\Phi}^{\mathrm{ha}_1}, \boldsymbol{\Phi}^{\mathrm{ch}_1} + \boldsymbol{a}) = U(\boldsymbol{\Phi}^{\mathrm{ha}_1}, \boldsymbol{\Phi}^{\mathrm{ch}_1}), \tag{42}$$

where $\boldsymbol{\phi}^{\mathrm{ha}_1^T} = [\phi_1^{\mathrm{ha}_1}, \dots, \phi_{n_\mathrm{H}}^{\mathrm{ha}_1}]$ and $\boldsymbol{\Phi}^{\mathrm{ch}_1^T} = [\phi_1^{\mathrm{ch}_1}, \dots, \phi_{n_\mathrm{C}}^{\mathrm{ch}_1}]$ are flux variables after the first transformation for the harmonic and charge subspaces respectively, and $\boldsymbol{a}$ is an n_C-dimensional vector defined as

$$\boldsymbol{a} = \sum_{i=1}^{n_\mathrm{C}} m_i \boldsymbol{a}_i, \quad m_i \in \mathbb{Z},$$

where $\boldsymbol{a}_i$ are primitive lattice vectors. We can write the periodic potential of Eq. 42 as a Fourier series:

$$U(\boldsymbol{\Phi}^{\mathrm{ha}_1}, \boldsymbol{\Phi}^{\mathrm{ch}_1}) = \sum_{\boldsymbol{k}} c_{\boldsymbol{k}}(\boldsymbol{\Phi}^{\mathrm{ha}_1}) e^{\frac{i}{\hbar}\boldsymbol{k}^T\boldsymbol{\Phi}^{\mathrm{ch}_1}}, \tag{43}$$

where $\boldsymbol{k}$ is a reciprocal lattice vector formed from basis vectors $\boldsymbol{k}_i$ satisfying $\boldsymbol{k}_i^T\boldsymbol{a}_j = 2\pi\hbar\delta_{ij}$.

Consider $|\boldsymbol{Q}_0^{\mathrm{ch}_1}\rangle = |Q_1^{\mathrm{ch}_1}, \dots, Q_{n_\mathrm{C}}^{\mathrm{ch}_1}\rangle$, an eigenvector of all charge operators $\hat{Q}_i^{\mathrm{ch}_1}$ in the charge block after the first transformation, with

$$\hat{Q}_i^{\mathrm{ch}_1} |\boldsymbol{Q}_0^{\mathrm{ch}_1}\rangle = Q_i^{\mathrm{ch}_1} |\boldsymbol{Q}_0^{\mathrm{ch}_1}\rangle. \tag{44}$$

Because $[\hat{\phi}_m^{\mathrm{ch}_1}, \hat{Q}_n^{\mathrm{ch}_1}] = i\hbar\delta_{mn}$, the operator $e^{\frac{i}{\hbar}\boldsymbol{k}^T\hat{\boldsymbol{\Phi}}^{\mathrm{ch}_1}}$ performs a charge displacement on $|\boldsymbol{Q}_0^{\mathrm{ch}_1}\rangle$:

$$e^{\frac{i}{\hbar}\boldsymbol{k}^T\hat{\boldsymbol{\Phi}}^{\mathrm{ch}_1}} |\boldsymbol{Q}_0^{\mathrm{ch}_1}\rangle = |\boldsymbol{Q}_0^{\mathrm{ch}_1} + \boldsymbol{k}\rangle. \tag{45}$$

As a consequence of Eqs. 43-45, the Hilbert space corresponding to the charge block is spanned by a discrete set of charge eigenvectors $|\boldsymbol{Q}^{\mathrm{ch}_1}\rangle$ on a reciprocal lattice. The lattice so far is defined up to a reference vector which we can take to be $|\boldsymbol{Q}_0^{\mathrm{ch}_1}\rangle$, defining the following basis for the charge-block subspace:

$$\mathcal{H}_{\boldsymbol{Q}_0^{\mathrm{ch}_1}} = \{|\boldsymbol{Q}_0^{\mathrm{ch}_1} + \boldsymbol{k}\rangle\,|\boldsymbol{k} = \sum_{i=1}^{n_\mathrm{C}} m_i\boldsymbol{k}_{\boldsymbol{i}}, m_i \in \mathbb{Z}\}.$$

Any vector $\boldsymbol{k}$ on the reciprocal lattice can be absorbed into the definition of $|\boldsymbol{Q}_0^{\mathrm{ch}_1}\rangle$, so we can take $|\boldsymbol{Q}_0^{\mathrm{ch}_1}\rangle$ to describe a possible non-integer offset to the reciprocal lattice. The offset, if present, corresponds to a conserved quantity that is equivalent to a quasi-momentum and is often called "gate charge" or "charge offset".

For calculations, charge operators and the potential of Eq. 43 have a more physically meaningful representations if each reciprocal vector has the the following form:

$$\tilde{\boldsymbol{k}}_i = (2e)\boldsymbol{e}_i, \tag{46}$$

where $2e$ is the Cooper pair charge and $\boldsymbol{e}_i$ is the n_C-dimensional vector with the ith element equal to one and the rest equal to zero (standard basis). We therefore construct $\boldsymbol{S}_2$ and $\boldsymbol{R}_2$ for the second transformation such that Eq. 46 is satisfied. The transformation acts only on the charge block, so $\boldsymbol{S}_2$ and $\boldsymbol{R}_2$ are block-diagonal:

$$\boldsymbol{S}_2 = \begin{bmatrix} \boldsymbol{I} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{S}_2^\mathrm{ch} \end{bmatrix},$$

$$\boldsymbol{R}_2 = \begin{bmatrix} \boldsymbol{I} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{R}_2^\mathrm{ch} \end{bmatrix},$$

where $\boldsymbol{I}$ is an $n_\mathrm{H} \times n_\mathrm{H}$ identity matrix and $(\boldsymbol{S}_2^\mathrm{ch})^T = (\boldsymbol{R}_2^\mathrm{ch})^{-1}$. Under these transformations Eq. 43 becomes,

$$U(\boldsymbol{\Phi}^{\mathrm{ha}_2}, \boldsymbol{\Phi}^{\mathrm{ch}_2}) = \sum_{\boldsymbol{k}} c_{\boldsymbol{k}}(\boldsymbol{\Phi}^{\mathrm{ha}_2}) e^{\frac{i}{\hbar}\boldsymbol{k}^T \boldsymbol{S}_2^\mathrm{ch} \boldsymbol{\Phi}^{\mathrm{ch}_2}},$$

where $\boldsymbol{\Phi}^{\mathrm{ha}_2}$ and $\boldsymbol{\Phi}^{\mathrm{ch}_2}$ are the flux variables after the second transformation, and $\boldsymbol{\Phi}^{\mathrm{ha}_2} = \boldsymbol{\Phi}^{\mathrm{ha}_1}$. Hence, the new reciprocal vectors $\tilde{\boldsymbol{k}}_i$ are related to previous ones by

$$\tilde{\boldsymbol{k}}_i^T = \boldsymbol{k}_i^T \boldsymbol{S}_2^\mathrm{ch}.$$

To ensure the $\tilde{\boldsymbol{k}}_i^T$ have the form of Eq. 46, $\boldsymbol{S}_2^\mathrm{ch}$ should satisfy:

$$\boldsymbol{S}_2^\mathrm{ch} = (\boldsymbol{K}^T)^{-1},$$

where the ith column of the $\boldsymbol{K}$ matrix is $\boldsymbol{k}_i$.

So far we have considered only the generic form of a periodic potential in Eq. 42. In practice, the reciprocal lattice vectors $\boldsymbol{k}_i$ for the Josephson potential are proportional to the prefactor vectors $\boldsymbol{w}_i^{\mathrm{ch}_1}$ within the cosines after the *first* transformation, defined by truncating $\boldsymbol{S}_1^T \boldsymbol{w}_i$ to the charge block. We can therefore form the columns of $\boldsymbol{K}$ using the n_C linearly-independent $\boldsymbol{w}_i^{\mathrm{ch}_1}$:

$$\boldsymbol{K} = (2e) \begin{bmatrix} \boldsymbol{w}_1^{\mathrm{ch}_1} & \dots & \boldsymbol{w}_{n_\mathrm{C}}^{\mathrm{ch}_1} \end{bmatrix} \tag{47}$$

This guarantees that all $\boldsymbol{w}_k^{\mathrm{ch}_2}$ (prefactors of the flux variables after the *second* transformation) are integers, and that the reciprocal vectors have the form of Eq. 46. Thus, after the second transformation the capacitance and inductance matrices $\boldsymbol{C}_2$ and $\boldsymbol{L}_2^*$ are

$$\boldsymbol{C}_2 = \begin{bmatrix} \boldsymbol{C}^{\mathrm{ha}_2} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{C}^{\mathrm{ch}_2} \end{bmatrix},$$

$$\boldsymbol{L}_2^* = \begin{bmatrix} \boldsymbol{L}^{*\mathrm{ha}}_2 & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} \end{bmatrix},$$

where $\boldsymbol{C}^{\mathrm{ha}_1} = \boldsymbol{C}^{\mathrm{ha}_2}$, $\boldsymbol{L}^{*\mathrm{ha}}_2 = \boldsymbol{L}^{*\mathrm{ha}}_1$, and $\boldsymbol{C}^{\mathrm{ch}_2} = (\boldsymbol{S}_2^\mathrm{ch})^T \boldsymbol{C}^{\mathrm{ch}_1} \boldsymbol{S}_2^\mathrm{ch}$. Although $\boldsymbol{C}^{\mathrm{ch}_1}$ is diagonal, $\boldsymbol{C}^{\mathrm{ch}_2}$ is not necessarily diagonal, so $\boldsymbol{C}_2$ and $\boldsymbol{L}_2^*$ have the same form as Eqs. 11-12.

C Interaction Hamiltonian

Here we obtain interaction Hamiltonians for the two generic circuits shown in Fig. 4. To do so, we describe how the drive circuit changes the Lagrangian of the main circuit, then compute the Hamiltonian by Legendre transform. We derive additional terms describing capacitive drives and inductive drives in Sec. C.1 and Sec. C.2, respectively.

C.1 Capacitive Drive

The coupling capacitor and voltage drive of Fig. 4a only affect the kinetic energy of the circuit as follows,

$$T'(\dot{\boldsymbol{\Phi}}) = T(\dot{\boldsymbol{\Phi}}) + \frac{1}{2} c_d (\dot{\phi}_i - \dot{\phi}_j - V_d)^2, \tag{48}$$

$$U'(\boldsymbol{\Phi}) = U(\boldsymbol{\Phi}), \tag{49}$$

where $T(\dot{\boldsymbol{\Phi}}) = \frac{1}{2}\dot{\boldsymbol{\Phi}}^T \boldsymbol{C} \dot{\boldsymbol{\Phi}}$ and $U(\boldsymbol{\Phi})$ are respectively the kinetic and potential energies related to the circuit without drive components, and (ϕ_i, ϕ_j) are flux variables for the nodes i and j that are connected through the drive circuit. The Lagrangian is $\mathcal{L} = T - U$, from which we compute redefined charge operators (as momenta conjugate to the flux variables):

$$Q'_t = \frac{\partial}{\partial \dot{\phi}_t}\mathcal{L} = \sum_m C_{tm}\dot{\phi}_m + c_d(\delta_{it} - \delta_{jt})(\dot{\phi}_i - \dot{\phi}_j - V_d). \tag{50}$$

We define the new capacitance matrix $\boldsymbol{C}'$ and charge drive vector $\boldsymbol{q_d}$ with following elements

$$C'_{mn} = C_{mn} + c_d(\delta_{im} - \delta_{jm})\delta_{mn},$$
$$q_{d_m} = c_d V_d (\delta_{im} - \delta_{jm}),$$

and rewrite Eqns . 48 and 50 in those parameters:

$$\boldsymbol{Q}' = \boldsymbol{C}' \dot{\boldsymbol{\Phi}} - \boldsymbol{q_d}, \tag{51}$$

$$T' = \frac{1}{2}\dot{\boldsymbol{\Phi}}^T C' \dot{\boldsymbol{\Phi}} - \boldsymbol{q_d}^T \dot{\boldsymbol{\Phi}} + \frac{1}{4}\boldsymbol{q_d}^T \frac{1}{c_d}\boldsymbol{q_d}. \tag{52}$$

The Hamiltonian of the circuit is

$$H = \boldsymbol{Q'}^T \dot{\boldsymbol{\Phi}} - T'(\dot{\boldsymbol{\Phi}}) + U'(\boldsymbol{\Phi}), \tag{53}$$

Thus, by substituting Eqns. 49, 51, 52 in Eq. 53:

$$\begin{aligned} H =& \frac{1}{2}\boldsymbol{Q'}^T \boldsymbol{C'}^{-1}\boldsymbol{Q'} + U(\boldsymbol{\Phi}) + \boldsymbol{q_d}^T \boldsymbol{C'}^{-1}\boldsymbol{Q'} \\ &+ \frac{1}{2}\boldsymbol{q_d}^T \boldsymbol{C'}^{-1}\boldsymbol{q_d} - \frac{1}{4}\boldsymbol{q_d}^T \frac{1}{c_d}\boldsymbol{q_d}. \end{aligned}$$

The last two terms contain no dynamical variables, so we neglect them as they describe only a global shift to the energy spectrum. Assuming c_d is much smaller than capacitors connected to nodes i and j, we can substitute the $\boldsymbol{Q'}$ and $\boldsymbol{C'}$ with $\boldsymbol{Q}$ and $\boldsymbol{C}$. Accordingly, the Hamiltonian can be written as

$$\hat{H} = \hat{H}_0 + \boldsymbol{q_d}^T \boldsymbol{C}^{-1}\hat{\boldsymbol{Q}}, \tag{54}$$

where $\hat{H}_0$ is the Hamiltonian of the circuit without drives. The interaction Hamiltonian for capacitive driving can be written using transformed charge operators as:

$$\hat{H}_c^{\mathrm{dr}} = (c_d V_d)\boldsymbol{e}_{ij}^T \boldsymbol{C}^{-1}\boldsymbol{R}\hat{\tilde{\boldsymbol{Q}}},$$

where we used $\boldsymbol{q}_d = c_d V_d \boldsymbol{e}_{ij}$ and $\boldsymbol{e}_{ij}$ is the n_{N}- dimensional vector with element $k \in \{1, 2, \cdots, n_{\mathrm{N}}\}$ equal to $\delta_{ik} - \delta_{kj}$.

C.2 Inductive Drive

In contrast to capacitive coupling, inductive coupling only changes the potential of the circuit and leaves the kinetic energy intact. The potential energy of the inductor l_{ij} coupled to the drive circuit in Fig.4b is,

$$\Delta U = \frac{1}{2} l_{ij} I_{ij}^2 + M I_d I_{ij} + \frac{1}{2} l_d I_d^2, \tag{55}$$

where I_{ij} is the current through the l_{ij} inductor and I_d is the source current. The flux drop across the l_{ij} and l_d inductors has the following relation with their currents due to mutual inductance M:

$$\begin{aligned} \phi_{ij} &= l_{ij} I_{ij} + M I_d, \\ \phi_d &= M I_{ij} + l_d I_d, \end{aligned} \tag{56}$$

Assuming $M \ll l_{ij}$, we express the Eq. 55 in ϕ_{ij} using Eq. 56:

$$\Delta U = \frac{1}{2 l_{ij}}\phi_{ij}^2 + \frac{M I_d}{l_{ij}}\phi_{ij} + \frac{1}{2} l_d I_d^2.$$

Thus, the kinetic and the potential energy of the circuit are

$$\begin{aligned} T'(\dot{\boldsymbol{\Phi}}) &= T(\dot{\boldsymbol{\Phi}}), \\ U'(\boldsymbol{\Phi}) &= U(\boldsymbol{\Phi}) + \frac{M I_d}{l_{ij}}\phi_{ij} + \frac{1}{2} l_d I_d^2, \end{aligned} \tag{57}$$

in which $T(\dot{\boldsymbol{\Phi}})$ and $U(\boldsymbol{\Phi})$ are the kinetic and potential energy respectively related to the circuit without drive components. By neglecting $\frac{1}{2} l_d I_d^2$ (since it is just a shift to energy spectrum of a circuit), the Hamiltonian can be obtained from the Lagrangian formed by Eq. 57 as

$$\hat{H} = \hat{H}_0 + \frac{M I_d}{l_{ij}}\boldsymbol{e}_{ij}^T \hat{\boldsymbol{\Phi}},$$

where $\hat{H}_0$ is the Hamiltonian of the circuit without drive elements, and we used $\hat{\phi}_i - \hat{\phi}_j = \boldsymbol{e}_{ij}^T \hat{\boldsymbol{\Phi}}$. Note that if the i or j is the ground node (that is $i = 0$) the corresponding flux operator is zero ($\hat{\phi}_i = 0$). The interaction Hamiltonian in transformed flux operators from inductive drive is:

$$\hat{H}_l^{\mathrm{dr}} = (M I_d)\frac{1}{l_{ij}}\boldsymbol{e}_{ij}^T \boldsymbol{S}\hat{\tilde{\boldsymbol{\Phi}}}.$$

D Time-dependent External Fluxes

Different forms of spanning tree lead to different distributions $\boldsymbol{b}_k$ of the external fluxes over the inductive elements, and therefore to different forms of the Hamiltonian. As discussed in [14, 15], for stationary external fluxes, all distributions $\boldsymbol{b}_k$ lead to Hamiltonians with the same energy spectrum. However, for time-dependent external fluxes terms proportional to $\propto \dot{\phi}\dot{\varphi}_{\mathrm{ext}}$ appear in the kinetic energy part of the Lagrangian; this is not captured in the description of Sec. 2. In this section, we explain a method to find the $\boldsymbol{b}_k$ for a Hamiltonian with time-dependent external fluxes, that remains consistent with the time-independent case in Sec. 2. Let all branch fluxes be denoted by

$$\boldsymbol{\Phi}_b^T = [\phi_{b,1}, \ldots, \phi_{b,(n_{\mathrm{N}}+n_{\mathrm{L}})}]$$

where $\phi_{b,k}$ is the kth branch flux, n_{N} is the number of circuit nodes, and n_{L} is the number of independent loops ($n_{\mathrm{L}} + n_{\mathrm{N}}$ is the number of branches). Fluxoid quantization and Faraday's law determine constraints for the kth loop $\mathcal{A}_k$ threaded by external flux $\phi_{\mathrm{ext},k}$:

$$\phi_{\mathrm{ext},k} = \boldsymbol{g}_k^T \boldsymbol{\Phi}_b, \tag{58}$$

where

$$[\boldsymbol{g}_k]_i = \begin{cases} 1, & \text{if } \phi_{b,i} \in \mathcal{A}_k \text{ with same direction as } \phi_{\mathrm{ext},k} \\ \text{-1}, & \text{if } \phi_{b,i} \in \mathcal{A}_k \text{ with opposite direction to } \phi_{\mathrm{ext},k} \\ 0, & \text{if } \phi_{b,i} \notin \mathcal{A}_k \end{cases}$$

Thus, Eq. 58 for all $n_{\rm L}$ loops of the circuit can be written as

$$\boldsymbol{\Phi}_{\rm ext} = \boldsymbol{G}\boldsymbol{\Phi}_b, \tag{59}$$

where $\boldsymbol{G}$ is a $n_{\rm L} \times (n_{\rm N} + n_{\rm L})$ matrix with kth row equal to $\boldsymbol{g}_k^T$. Moreover, the relation between kth branch flux $\phi_{b,k}$, node flux variables $\boldsymbol{\Phi}$ (circuit degrees of freedom), and external fluxes $\boldsymbol{\Phi}_{\rm ext}$ can be expressed by,

$$\phi_{b,k} = \boldsymbol{w}_k^T\boldsymbol{\Phi} + \boldsymbol{b}_k^T\boldsymbol{\Phi}_{\rm ext}, \tag{60}$$

where $\boldsymbol{w}_k$ are calculated as in Sec. 2 and $\boldsymbol{b}_k$ are yet to be specified. Eq. 60 for all k can be written in the matrix form of

$$\boldsymbol{\Phi}_b = \boldsymbol{W}\boldsymbol{\Phi} + \boldsymbol{B}\boldsymbol{\Phi}_{\rm ext}, \tag{61}$$

where $\boldsymbol{W}$ and $\boldsymbol{B}$ are $(n_{\rm N} + n_{\rm L}) \times n_{\rm N}$ and $(n_{\rm N} + n_{\rm L}) \times n_{\rm L}$ matrices with respective kth rows given by $\boldsymbol{w}_k^T$ and $\boldsymbol{b}_k^T$. We define an $(n_{\rm N} + n_{\rm L})$-dimensional vector $\boldsymbol{\Phi}^+$ and $(n_{\rm N} + n_{\rm L}) \times (n_{\rm N} + n_{\rm L})$ matrix of $\boldsymbol{P}$ as,

$$\begin{aligned}\boldsymbol{\Phi}^+ &= \begin{bmatrix}\boldsymbol{\Phi}\\ \boldsymbol{\Phi}_{\rm ext}\end{bmatrix},\\ \boldsymbol{P} &= \begin{bmatrix}\boldsymbol{W} & \boldsymbol{B}\end{bmatrix},\end{aligned}$$

to rewrite Eq. 61 compactly as

$$\boldsymbol{\Phi}_b = \boldsymbol{P}\boldsymbol{\Phi}^+. \tag{62}$$

The kinetic energy of the Lagrangian is,

$$T = \frac{1}{2}\dot{\boldsymbol{\Phi}}_b^T\boldsymbol{C}_{\rm ed}\dot{\boldsymbol{\Phi}}_b,$$

where $\boldsymbol{C}_{\rm ed}$ is an $(n_{\rm N} + n_{\rm L}) \times (n_{\rm N} + n_{\rm L})$ diagonal matrix with kth diagonal element equal to the capacitance of the kth branch. Using Eq. 62, the kinetic energy can be written as

$$T = \frac{1}{2}(\dot{\boldsymbol{\Phi}}^+)^T\boldsymbol{C}\dot{\boldsymbol{\Phi}}^+,$$

where,

$$\begin{aligned}\boldsymbol{C} = \boldsymbol{P}^T\boldsymbol{C}_{\rm ed}\boldsymbol{P} &= \begin{bmatrix}\boldsymbol{W}^T\\ \boldsymbol{B}^T\end{bmatrix}\boldsymbol{C}_{\rm ed}\begin{bmatrix}\boldsymbol{W} & \boldsymbol{B}\end{bmatrix}\\ &= \begin{bmatrix}\boldsymbol{W}^T\boldsymbol{C}_{\rm ed}\boldsymbol{W} & \boldsymbol{W}^T\boldsymbol{C}_{\rm ed}\boldsymbol{B}\\ \boldsymbol{B}^T\boldsymbol{C}_{\rm ed}\boldsymbol{W} & \boldsymbol{B}^T\boldsymbol{C}_{\rm ed}\boldsymbol{B}\end{bmatrix}\end{aligned}$$

The off-diagonal blocks of $\boldsymbol{C}$ lead to terms $\propto \dot{\phi}\varphi_{\rm ext}$. To avoid this, the $\boldsymbol{B}$ matrix should satisfy the following condition:

$$\boldsymbol{W}^T\boldsymbol{C}_{\rm ed}\boldsymbol{B} = \boldsymbol{0}, \tag{63}$$

where $\boldsymbol{0}$ is a $n_{\rm N} \times n_{\rm L}$ zero matrix. The $\boldsymbol{B}$ matrix has $(n_{\rm N} + n_{\rm L}) \times n_{\rm L}$ elements that are unknown and Eq. 63 introduces $n_{\rm N} \times n_{\rm L}$ number of equations. Thus, $n_{\rm L}^2$ equations are needed to specify $\boldsymbol{B}$ uniquely. Substituting Eq. 62 in Eq. 59, we have:

$$\begin{aligned}\boldsymbol{G}\boldsymbol{\Phi}_b = \boldsymbol{G}\boldsymbol{P}\boldsymbol{\Phi}^+ &= \boldsymbol{G}\begin{bmatrix}\boldsymbol{W} & \boldsymbol{B}\end{bmatrix}\begin{bmatrix}\boldsymbol{\Phi}\\ \boldsymbol{\Phi}_{\rm ext}\end{bmatrix}\\ &= \boldsymbol{G}\boldsymbol{W}\boldsymbol{\Phi} + \boldsymbol{G}\boldsymbol{B}\boldsymbol{\Phi}_{\rm ext}.\end{aligned}$$

$\boldsymbol{G}\boldsymbol{W}$ should be zero by definition, since by summing the flux drop at each branch of the loop, the net result should be independent of the flux node operators. Hence,

$$\boldsymbol{G}\boldsymbol{B}\boldsymbol{\Phi}_{\rm ext} = \boldsymbol{\Phi}_{\rm ext}.$$

Because this equation should hold for all $\boldsymbol{\Phi}_{\rm ext}$, we find the following constraint for $\boldsymbol{B}$, which provides the $n_{\rm L}^2$ needed equations:

$$\boldsymbol{G}\boldsymbol{B} = \boldsymbol{I}, \tag{64}$$

where $\boldsymbol{I}$ is an $n_{\rm L} \times n_{\rm L}$ identity matrix. We combine Eqs. 63 and 64 as

$$\begin{bmatrix}\boldsymbol{W}^T\boldsymbol{C}_{\rm ed}\\ \boldsymbol{G}\end{bmatrix}\boldsymbol{B} = \begin{bmatrix}\boldsymbol{0}\\ \boldsymbol{I}\end{bmatrix},$$

and specify $\boldsymbol{B}$ by

$$\boldsymbol{B} = \begin{bmatrix}\boldsymbol{W}^T\boldsymbol{C}_{\rm ed}\\ \boldsymbol{G}\end{bmatrix}^{-1}\begin{bmatrix}\boldsymbol{0}\\ \boldsymbol{I}\end{bmatrix}.$$

This process has been implemented in `SQcircuit` to describe time-dependent external fluxes, in particular to accurately predict dephasing rates due to flux noise.

E SQcircuit Codes

Here we provide Python code that uses the `SQcircuit` library to produce results in the main body of this paper. We obtain the results of Fig.3 by

```python
# import the libraries
import SQcircuit as sq
import numpy as np
import matplotlib.pyplot as plt

# define the loop of the circuit
loop1 = sq.Loop()

# define the circuit's elements
C = sq.Capacitor(0.15, "GHz")
```

```python
CJ = sq.Capacitor(10, "GHz")
JJ = sq.Junction(5, "GHz", loops=[loop1])
L = sq.Inductor(0.13, "GHz", loops=[loop1])

# define the zero-pi circuit
elements = {
    (0, 1): [CJ, JJ],
    (0, 2): [L],
    (0, 3): [C],
    (1, 2): [C],
    (1, 3): [L],
    (2, 3): [CJ, JJ]
}
cr = sq.Circuit(elements)

# set the truncation numbers
cr.set_trunc_nums([25, 25])

# external flux for sweeping over
phi = np.linspace(0, 1,100)

# spectrum of the circuit
n_eig=5
spec = np.zeros((n_eig, len(phi)))

for i in range(len(phi)):
    # set the external flux for the loop
    loop1.set_flux(phi[i])

    # diagonlize the circuit
    spec[:, i], _ = cr.diag(n_eig)

# plot the energy spectrum
plt.figure()
for i in range(n_eig):
    plt.plot(phi, spec[i,:] - spec[0,:])
plt.xlabel(r"$\Phi_{ext}/\Phi_0$")
plt.ylabel(r"$f_i-f_0$[GHz]")
plt.show()

# set the external flux back to zero
loop1.set_flux(0)
_, _ = cr.diag(n_eig=2)

# create a range for each mode
phi1 = np.pi*np.linspace(-1,1,100)
phi2 = np.pi*np.linspace(-0.5,1.5,100)

# the ground state
state0 = cr.eig_phase_coord(
    k=0, grid=[phi1, phi2])

# the first excited state
state1 = cr.eig_phase_coord(
    k=1, grid=[phi1, phi2])

# plot the ground state
plt.figure()
plt.pcolor(phi1, phi2,
           np.abs(state0)**2,
           shading='auto')
plt.xlabel(r'$\varphi_1$')
plt.ylabel(r'$\varphi_2$')
plt.show()

# plot the first excited state
plt.figure()
plt.pcolor(phi1, phi2,
           np.abs(state1)**2,
           shading='auto')
plt.xlabel(r'$\varphi_1$')
plt.ylabel(r'$\varphi_2$')
plt.show()
```

and the results of Fig.5 by

```python
# import the libraries
import SQcircuit as sq
import numpy as np
import matplotlib.pyplot as plt

# define the loop of the circuit
loop1 = sq.Loop()

# define the circuit's elements
C = sq.Capacitor(3.6, "GHz", Q=1e6)
L = sq.Inductor(0.46, "GHz", loops=[loop1])
JJ = sq.Junction(10.2, "GHz", cap=C,
                 A=5e-7, loops=[loop1])

# define the Fluxonium circuit
elements = {(0, 1): [L, JJ]}
cr = sq.Circuit(elements, flux_dist="all")

# set the truncation numbers
cr.set_trunc_nums([100])

# external flux for sweeping over
phi = np.linspace(0, 1, 300)

# T_1 and T_phi
T_1 = np.zeros_like(phi)
T_phi = np.zeros_like(phi)

for i in range(len(phi)):
    # set the external flux for the loop
    loop1.set_flux(phi[i])

    # diagonalize the circuit
    _, _ = cr.diag(n_eig=2)

    # get the T_1 for capacitive loss
    T_1[i] = 1/cr.dec_rate(
             dec_type="capacitive",
             states=(1,0))

    # get the T_phi for cc noise
    T_phi[i] = 1/cr.dec_rate(
               dec_type="cc",
               states=(1,0))

# plot the T_1 from the capacitive loss
plt.figure()
plt.semilogy(phi, T_1)
plt.xlabel(r"$\Phi_{ext}/\Phi_0$")
plt.ylabel(r"$s$")
plt.show()

# plot the T_phi from the critical
# current noise
plt.figure()
plt.semilogy(phi, T_phi)
plt.xlabel(r"$\Phi_{ext}/\Phi_0$")
plt.ylabel(r"$s$")
plt.show()
```

References

[1] A. Gyenis, A. D. Paolo, J. Koch, A. Blais, A. A. Houck, and D. I. Schuster, “Moving beyond the transmon: Noise-protected superconducting quantum

circuits," 6 2021. [Online]. Available: https://doi.org/10.1103/PRXQuantum.2.030101

[2] P. Krantz, M. Kjaergaard, F. Yan, T. P. Orlando, S. Gustavsson, and W. D. Oliver, "A quantum engineer's guide to superconducting qubits," *Appl. Phys. Rev*, vol. 6, p. 21318, 6 2019. [Online]. Available: https://doi.org/10.1063/1.5089550

[3] Z. Wang, T. Rajabzadeh, N. Lee, and A. H. Safavi-Naeini, "Automated discovery of autonomous quantum error correction schemes," *PRX Quantum*, vol. 3, no. 2, p. 020302, 2022. [Online]. Available: https://doi.org/10.1103/PRXQuantum.3.020302

[4] T. Rajabzadeh, C. J. Sarabalis, O. Atalar, and A. H. Safavi-Naeini, "Photonics-to-free-space interface in lithium niobate-on-sapphire," pp. STu4J–6, 2020. [Online]. Available: https://doi.org/10.1364/CLEO_SI.2020.STu4J.6

[5] N. R. Lee, Y. Guo, A. Y. Cleland, E. A. Wollack, R. G. Gruenke, T. Makihara, Z. Wang, T. Rajabzadeh, W. Jiang, F. M. Mayor *et al.*, "Strong dispersive coupling between a mechanical resonator and a fluxonium superconducting qubit," *arXiv preprint arXiv:2304.13589*, 2023. [Online]. Available: https://doi.org/10.48550/arXiv.2304.13589

[6] Z. L. Xiang, S. Ashhab, J. Q. You, and F. Nori, "Hybrid quantum circuits: Superconducting circuits interacting with other quantum systems," *Reviews of Modern Physics*, vol. 85, pp. 623–653, 4 2013. [Online]. Available: https://doi.org/10.1103/RevModPhys.85.623

[7] U. Vool and M. Devoret, "Introduction to quantum electromagnetic circuits," vol. 45. John Wiley and Sons Ltd, 7 2017, pp. 897–934. [Online]. Available: https://doi.org/10.1002/cta.2359

[8] T. Menke, F. Häse, S. Gustavsson, A. J. Kerman, W. D. Oliver, and A. Aspuru-Guzik, "Automated design of superconducting circuits and its application to 4-local couplers," *npj Quantum Information*, vol. 7, 2021. [Online]. Available: https://doi.org/10.1038/s41534-021-00382-6

[9] W. C. Smith, A. Kou, U. Vool, I. M. Pop, L. Frunzio, R. J. Schoelkopf, and M. H. Devoret, "Quantization of inductively shunted superconducting circuits," *Physical Review B*, vol. 94, 10 2016. [Online]. Available: https://doi.org/10.1103/PhysRevB.94.144507

[10] A. J. Kerman, "Efficient numerical simulation of complex josephson quantum circuits." [Online]. Available: https://doi.org/10.48550/arXiv.2010.14929

[11] M. F. Gely and G. A. Steele, "Qucat: Quantum circuit analyzer tool in python," 8 2019. [Online]. Available: http://doi.org/10.1088/1367-2630/ab60f6

[12] P. Groszkowski and J. Koch, "Scqubits: a python package for superconducting qubits," *Quantum*, vol. 5, p. 583, 11 2021. [Online]. Available: https://doi.org/10.22331/q-2021-11-17-583

[13] P. Aumann, T. Menke, W. D. Oliver, and W. Lechner, "Circuitq: An open-source toolbox for superconducting circuits," 6 2021. [Online]. Available: https://doi.org/10.48550/arXiv.2106.05342

[14] X. You, J. A. Sauls, and J. Koch, "Circuit quantization in the presence of time-dependent external flux," *Physical Review B*, vol. 99, 5 2019. [Online]. Available: https://doi.org/10.1103/PhysRevB.99.174512

[15] R.-P. Riwar and D. P. DiVincenzo, "Circuit quantization with time-dependent magnetic fields for realistic geometries," 3 2021. [Online]. Available: https://doi.org/10.48550/arXiv.2103.03577

[16] J. R. Johansson, P. D. Nation, and F. Nori, "Qutip: An open-source python framework for the dynamics of open quantum systems," 10 2011. [Online]. Available: http://doi.org/10.1016/j.cpc.2012.02.021

[17] P. Brooks, A. Kitaev, and J. Preskill, "Protected gates for superconducting qubits," 2 2013. [Online]. Available: http://doi.org/10.1103/PhysRevA.87.052306

[18] P. Groszkowski, A. D. Paolo, A. L. Grimsmo, A. Blais, D. I. Schuster, A. A. Houck, and J. Koch, "Coherence properties of the 0-π qubit," 8 2017. [Online]. Available: http://doi.org/10.1088/1367-2630/aab7cd

[19] A. Gyenis, P. S. Mundada, A. D. Paolo, T. M. Hazard, X. You, D. I. Schuster, J. Koch, A. Blais, and A. A. Houck, "Experimental realization of a protected superconducting circuit derived from the $0 - \pi$ qubit," *PRX Quantum*, vol. 2, 3 2021. [Online]. Available: https://doi.org/10.1103/PRXQuantum.2.010339

[20] B. Yurke and J. S. Denker, "Quantum network theory," *Physical Review A*, vol. 29, pp. 1419–1437, 3 1984. [Online]. Available: https://doi.org/10.1103/PhysRevA.29.1419

[21] G. Burkard, R. H. Koch, and D. P. DiVincenzo, "Multilevel quantum description of decoherence in superconducting qubits," *Physical Review B - Condensed Matter and Materials Physics*, vol. 69, 2 2004. [Online]. Available: https://doi.org/10.1103/PhysRevB.69.064503

[22] Z. Huang, P. S. Mundada, A. Gyenis, D. I. Schuster, A. A. Houck, and J. Koch, "Engineering dynamical sweet spots to protect qubits from $1/f$ noise," 4 2020. [Online]. Available: http://doi.org/10.1103/PhysRevApplied.15.034065

[23] A. Di Paolo, T. E. Baker, A. Foley, D. Sénéchal, and A. Blais, "Efficient modeling of superconducting quantum circuits with tensor networks," *npj Quantum Information*, vol. 7, no. 1, pp. 1–11, 2021. [Online]. Available: https://doi.org/10.1038/s41534-020-00352-4

[24] J. Koch, T. M. Yu, Jay, A. A. Houck, D. I. Schuster, J. Majer, A. Blais, M. H. Devoret, S. M. Girvin, and R. J. Schoelkopf, "Charge-insensitive qubit design derived from the cooper pair box," *Physical Review A - Atomic, Molecular, and Optical Physics*, vol. 76, 10 2007. [Online]. Available: https://doi.org/10.1103/PhysRevA.76.042319

[25] A. A. Clerk, M. H. Devoret, S. M. Girvin, F. Marquardt, and R. J. Schoelkopf, "Introduction to quantum noise, measurement, and amplification," *Reviews of Modern Physics*, vol. 82, pp. 1155–1208, 4 2010. [Online]. Available: https://doi.org/10.1103/RevModPhys.82.1155

[26] V. B. Braginsky, V. S. Ilchenko, and K. S. Bagdassarov, "Experimental observation of fundamental microwave absorption in high-quality dielectric crystals," 1987. [Online]. Available: https://doi.org/10.1016/0375-9601(87)90676-1

[27] W. C. Smith, A. Kou, X. Xiao, U. Vool, and M. H. Devoret, "Superconducting circuit protected by two-cooper-pair tunneling," *npj Quantum Information*, vol. 6, 12 2020. [Online]. Available: https://doi.org/10.1038/s41534-019-0231-2

[28] G. Catelani, R. J. Schoelkopf, M. H. Devoret, and L. I. Glazman, "Relaxation and frequency shifts induced by quasiparticles in superconducting qubits," *Physical Review B - Condensed Matter and Materials Physics*, vol. 84, 8 2011. [Online]. Available: https://doi.org/10.1103/PhysRevB.84.064517

[29] G. Ithier, E. Collin, P. Joyez, P. J. Meeson, D. Vion, D. Esteve, F. Chiarello, A. Shnirman, Y. Makhlin, J. Schriefl, and G. Schön, "Decoherence in a superconducting quantum bit circuit," *Physical Review B - Condensed Matter and Materials Physics*, vol. 72, 10 2005. [Online]. Available: https://doi.org/10.1103/PhysRevB.72.134519

[30] "The sqcircuit github is located at the following address." [Online]. Available: https://github.com/stanfordLINQS/SQcircuit

[31] "The sqcircuit documentation can be found online." [Online]. Available: https://sqcircuit.org/

[32] "The sphinx code for sqcircuit documentation is located at the following github address." [Online]. Available: https://github.com/stanfordLINQS/SQcircuit-doc

[33] M. Pechal and A. H. Safavi-Naeini, "Millimeter-wave interconnects for microwave-frequency quantum machines," *PHYSICAL REVIEW A*, vol. 96, p. 42305, 2017. [Online]. Available: https://doi.org/10.1103/PhysRevA.96.042305