

ARTICLE OPEN



Integrating quantum processor device and control optimization in a gradient-based framework

Xiaotong Ni¹✉, Hui-Hai Zhao², Lei Wang^{3,4}, Feng Wu¹ and Jianxin Chen⁵

In a quantum processor, the device design and external controls together contribute to the quality of the target quantum operations. As we continuously seek better alternative qubit platforms, we explore the increasingly large device and control design space. Thus, optimization becomes more and more challenging. In this work, we demonstrate that the figure of merit reflecting a design goal can be made differentiable with respect to the device and control parameters. In addition, we can compute the gradient of the design objective efficiently in a similar manner to the back-propagation algorithm and then utilize the gradient to optimize the device and the control parameters jointly and efficiently. Therefore, our work extends the scope of the quantum optimal control to device design and provides an efficient optimization method. We also demonstrate the viability of gradient-based joint optimization over the device and control parameters through a few examples based on the superconducting qubits.

npj Quantum Information (2022)8:106; <https://doi.org/10.1038/s41534-022-00614-3>

INTRODUCTION

Quantum mechanics provides an entirely new way to think about information processing. Information is stored in quantum systems and the dynamics are described by the Schrödinger equation or the master equation. To achieve desired operations, we need to design systems with suitable Hamiltonians and dissipation as well as time-dependent external controls. Concrete examples of such quantum information processing tasks include quantum computing, quantum simulation, quantum error correction, and quantum metrology. At this level of abstraction, it is clear that we are dealing with optimization problems that consist of both Hamiltonian design and control. Traditionally, the optimization of the design^{1–4} and the optimization of the control^{5–7} are usually studied separately. In this work, we treat them as equal components in a single optimization problem and use gradient information to speed up optimization.

We consider superconducting circuits, which are one of the most promising hardware platforms due to their device design versatility and fabrication scalability. While the design versatility allows for many qubit and coupling types^{8–17}, it also makes the quest of finding the best design harder due to the larger parameter space. As mentioned earlier, the optimization of the device design is further complicated by the fact that we need to consider the control schemes simultaneously, which has been noted in previous work^{18,19}. For example, complex control schemes are needed for bosonic code qubits, and often numerical optimization is required to obtain the best gate performances^{20–22}. In scenarios where we cannot estimate the control performances with analytical formulas, the optimization problem of the qubit design inevitably becomes a joint optimization problem of design and control with an even larger parameter space. A common strategy to handle large optimization problems is using efficiently computed gradients to help us traverse the optimization landscape. One milestone in this direction is the GRadiant Ascent Pulse Engineering (GRAPE) algorithm⁵. In this work, we showed that the figure of merit of a target goal such as

the gate fidelity can be made differentiable with respect to the parameters from the device and the control together. The gradients can be computed in a single computation similar to the GRAPE algorithm and the back-propagation algorithm²³. More accurately, the ratio of the time needed to compute the figure of merit and its gradient is independent of the number of control parameters. This makes our method feasible not only for optimizing complex pulse shapes but also for optimizing large processor device designs, and more interestingly, jointly optimizing the design and control.

RESULTS

Integrating design and control aspects

In this work, we mainly focus on the two stages of quantum processor design that involve finding optimal values of the device parameters in the system Hamiltonian and deriving optimal control pulses to maximize the performance of the target quantum processor. When the device parameters are determined and we only need to optimize over the control parameters for better performance or robustness, this is precisely what quantum optimal control and robust control do. In contrast, we may sometimes need to optimize the spectral or other properties of the device Hamiltonian itself, e.g., for determining tunable ZZ-interaction on–off ratios^{24,25} or for designing 4-local interaction couplers¹.

Now we introduce our framework that unifies the above two scenarios, optimizing a system's performance over the control parameters and the device parameters jointly as illustrated in Fig. 1. The Hamiltonian of a quantum system with control fields takes the following form:

$$H(\vec{h}, \vec{c}, t) = H_{\text{dev}}(\vec{h}) + \sum_{k=1}^{N_c} f_k(\vec{c}, t) C_k(\vec{h}), \quad (1)$$

where H_{dev} , C_k , and f_k are the device Hamiltonian, control operators, and control fields, respectively, and \vec{h} and \vec{c} denote the device and control parameters, respectively. The goal is to

¹Alibaba Quantum Laboratory, Alibaba Group, Hangzhou, Zhejiang 311121, P. R. China. ²Alibaba Quantum Laboratory, Alibaba Group, Beijing 100102, P. R. China. ³Institute of Physics, Chinese Academy of Science, Beijing 100190, P. R. China. ⁴Songshan Lake Materials Laboratory, Dongguan, Guangdong 523808, P. R. China. ⁵Alibaba Quantum Laboratory, Alibaba Group USA, Bellevue, WA 98004, USA. ✉email: xiaotong.nxt@alibaba-inc.com

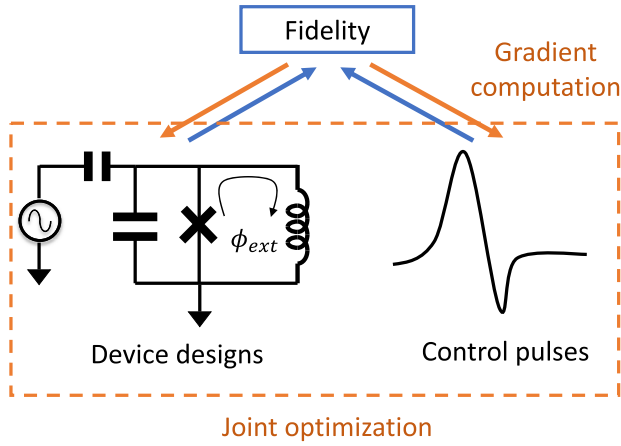


Fig. 1 Illustration of joint optimization in circuit quantum electrodynamics systems. Both device designs and control pulses contain many parameters, and together they determine the fidelity of the quantum operations. In this work, we consider the problem of optimizing them together.

optimize \vec{h} and \vec{c} for certain objectives, such as implementing a specific unitary operator within a given time τ . This extended setup formulates a device-control codesign problem. In Supplementary Note I, we explain why we expect that optimizing \vec{h} and \vec{c} simultaneously will be more efficient compared to the case where we artificially break the optimization problem into two and optimize \vec{h} and \vec{c} alternatively. Compared to the optimal control problem where \vec{h} is fixed, the extended setup will open more possibilities. For example, we can optimize the device parameters \vec{h} for a gate implementation that is robust against control fluctuations and deviations (see “optimizing device and control parameters for robust control”). In comparison, with conventional quantum control, we only optimize over control parameters for robust schemes. In “efficient gradient computation for both device and control parameters”, we give a procedure to efficiently compute the gradients for both device and control parameters through backpropagation.

iSWAP gate with fluxonium qubits

We will demonstrate our method for an iSWAP between two capacitively coupled fluxonium qubits, the diagram of which is depicted in Fig. 2a. The Hamiltonian of the system is

$$H(t) = H_{f,1} + H_{f,2}(t) + J_C n_1 n_2, \quad (2)$$

where J_C is the coupling strength between the two qubits, and $H_{f,i}$ is the Hamiltonian of the i -th fluxonium qubit¹⁵, defined as follows:

$$H_{f,i} = 4E_{C,i}n_i^2 + \frac{1}{2}E_{L,i}(\varphi_i + \varphi_{\text{ext},i})^2 - E_{J,i}\cos(\varphi_i), \quad (3)$$

where E_C , E_J , and E_L are the charging energy, the Josephson energy, and the inductive energy, respectively, φ_i is the phase operator, and n_i is the conjugate charge operator. We set the external flux of the second qubit to be a time-dependent control field initially at π , while for the first qubit, the external flux is fixed to be π , that is, $\varphi_{\text{ext},1} = \pi$ and $\varphi_{\text{ext},2}(0) = \pi$. The waveform of $\varphi_{\text{ext},2}(t)$ is chosen to be a trapezoid pulse as shown in Fig. 2b, which is given by

$$\varphi_{\text{ext},2}(t) = \pi + \varphi_p \min \left\{ \text{ReLU} \left(\frac{t}{t_{\text{ramp}}} \right), 1, \text{ReLU} \left(\frac{2t_{\text{ramp}} + t_{\text{plateau}} - t}{t_{\text{ramp}}} \right) \right\}, \quad (4)$$

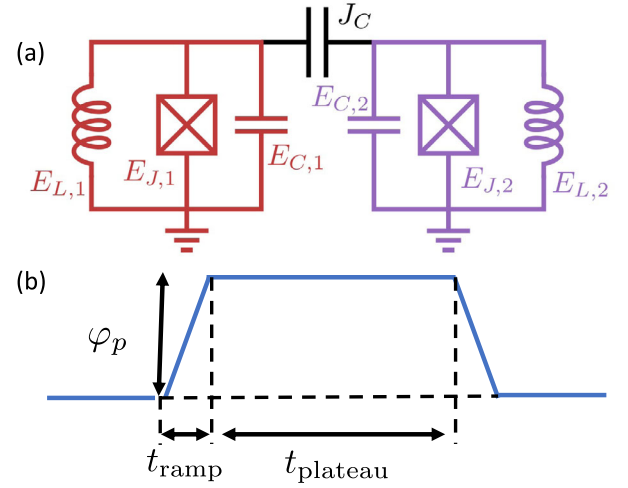


Fig. 2 Device and control parameters optimized for the iSWAP gate. **a** Circuit diagram of two fluxonium qubits coupled by a capacitor. Each circuit component corresponds to a term in the Hamiltonian, and we label the components with their energy parameters. The capacitive coupling strength J_C is determined by the capacitance of each qubit and between these two qubits. **b** Time-dependent part of the control pulse shape. It corresponds to the second term on the right-hand side in Eq. (4).

where φ_p and t_{plateau} are the external flux bias and holding time of the plateau of the waveform, t_{ramp} is the time for the rising and falling edges, and $\text{ReLU}(x) = \max(x, 0)$ is a commonly used activation function in neural networks. We call E_C , E_J , E_L , and J_C the device parameters, and the parameters in $\varphi_{\text{ext},2}(t)$ are the control parameters (Though $\varphi_{\text{ext},2}(t)$ was part of the device Hamiltonian, it comes from the external flux control, so the parameters therein are more appropriately regarded as control parameters). These are all parameters involved in the optimization.

Some details of the construction of the finite dimensional Hamiltonian are provided in what follows. We first write the Hamiltonian of the single-qubit fluxonium in the discretized phase basis over a finite range of phase values. To obtain sufficiently accurate low-energy eigenstates for the optimization task, we select the range to be $\varphi \in [-5\pi, 5\pi]$ and the number of basis points to be $n_{\text{basis}} = 400$.

We select the average fidelity to be the objective function. The leakage is small for the iSWAP gate scheme, and the optimized results have a leakage smaller than 10^{-4} . Therefore, it is a good approximation to compute the fidelity based on the truncated 4×4 unitary matrix. However, there are usually single qubit phases accumulated during the implementation of the gate, and we can compensate for these phases at a very low cost²⁶. Therefore, we will use a modified average fidelity $F_m(U)$ with phase compensation included (see Supplementary Note IV). We include other terms in the total objective function, and the exact forms of these terms will be given in Supplementary Note V. First, we include a penalty P_{decoh} which roughly estimates the infidelity caused by the decoherence of T_1 and T_2 . We will not pursue an accurate decoherence model as it varies drastically for different experimental setups. Instead, all the losses listed in ref. ²⁷ have closed-form expressions that allow us to compute their gradients straightforwardly. In practice, we usually go through a few iterations of experimental testing for one chip design. So in the later iterations, we can optimize based on the decoherence model measured in the earlier iterations. Another two penalties we include are P_{fDiff} and P_{zz} . P_{fDiff} discourages the $0 \leftrightarrow 1$ frequencies of the two fluxonium qubits from being too close. P_{zz} prevents the

ZZ-crosstalk from being too large when both fluxonium qubits are at their sweet spots. Both P_{fDiff} and P_{zz} are useful for realizing single qubit gates including the idling gate with low crosstalk errors. The last one is P_{fm} , which forces the fluxonium parameters to be realistic for fabrication. In total, we set the objective function for minimization to be

$$O = \ln(1 - F_m(U)) + P_{\text{decoh}} + P_{\text{fDiff}} + P_{\text{zz}} + P_{\text{fm}}. \quad (5)$$

Another desired feature of the iSWAP gate scheme is robustness against control errors. Here, we demonstrate how we can improve the robustness of the gate fidelity $F_m(U)$ with respect to the amplitude parameter φ_p in Eq. (4). To do this, we set $\Delta\vec{c}_{1,2} = (\pm\Delta\varphi_p, 0, \dots, 0)$, i.e., all the control parameters in $\Delta\vec{c}_{1,2}$ other than φ_p are set to 0. Then, we can construct the robustness objective function O_{avg} from O according to Eq. (15). We use the Adam optimizer²⁸ to minimize the objective functions O and O_{avg} . For the initial condition and optimizer hyperparameters listed in Supplementary Note V, the objective functions during optimization are shown in Fig. 3, and the comparison of the result's robustness is shown in Fig. 4. As shown in Fig. 4, the parameters

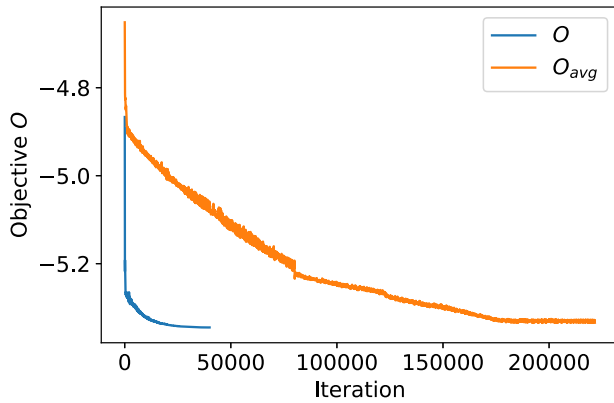


Fig. 3 iSWAP gate optimization loss during training. The blue curve corresponds to the optimization of O in Eq. (5) and the orange curve corresponds to O_{avg} in Eq. (15). The curves are smoothed using the Savitzky–Golay filter. The optimization of O_{avg} is interrupted and continued again at iteration 80000 (see Supplementary Note V for more discussion).

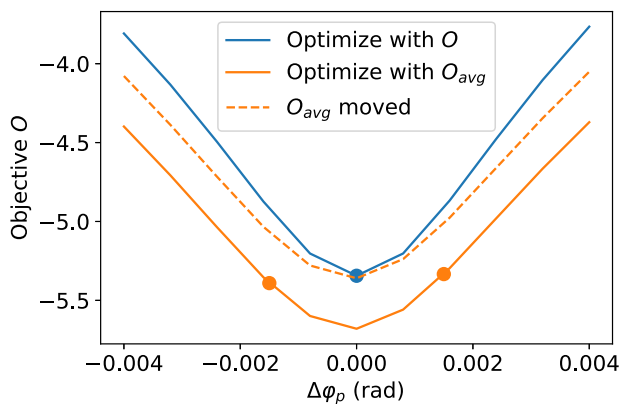


Fig. 4 Robustness of the iSWAP gate with respect to the control parameter φ_p . For different $\Delta\varphi_p$, we compute $O(\text{add}(\vec{p}, \varphi_p, \Delta\varphi))$. The blue and orange dots mark the $\Delta\varphi_p$ values used to perform the optimization. The orange dotted curve is a translation of the orange curve. The curvature of the orange dotted curve is smaller than that of the blue curve. Therefore, the parameters obtained from the robustness optimization are more robust.

\vec{p}_{avg} obtained by optimizing O_{avg} are indeed more robust with respect to the errors in φ_p compared to \vec{p} obtained by optimizing O , and the minimum is lower. The latter effect was unintended. Retrospectively, due to the change of the objective function, \vec{p}_{avg} escaped the local minimum of \vec{p} .

This example shows that the joint gradient optimization is viable for a practical problem. Both capacitively coupled fluxonium qubits and the control pulse Eq. (4) have been realized in the experiment²⁹, and the results match the numerical simulation well (see the supplementary material of ref. 29). Thus we expect the gain from the optimization can be transferred to experiments as long as we have an accurate decoherence model. We reach a local minimum of objective O for both the device and control parameters in a single run of the gradient optimization. The time T_{fwd} and T_{grad} needed for computing O and ∇O are reported in Supplementary Note V. The ratio $T_{\text{grad}}/T_{\text{fwd}} \sim 4$ is close to the bound proved in ref. 30. In Supplementary Note III, we show the ratio $T_{\text{grad}}/T_{\text{fwd}} < 4$ for diagonalizing fluxonium Hamiltonians as the system size and the number of parameters grow. The speedup compared to finite-difference is about three times for iSWAP optimization with seven device parameters and three control parameters, but it will scale linearly with the number of parameters needed to describe more complex qubit structures or large processors with more qubits. Since simulating quantum processors is highly computationally intensive, we will have an even more significant absolute speedup.

Adiabatic flux-tuning gates with transmon qubits

Several popular cQED 2-qubit gate schemes are adiabatic^{16,18,25,31,32}. For these adiabatic gate schemes, we can obtain a good estimate of the gate performance solely based on the spectral properties of the Hamiltonian. An example is that we can use the on–off ratio of the ZZ-interactions to benchmark the design of a tunable coupler. More generally, there is a class of “optimal design” problems where we only optimize the design parameters, while the control parameters are implicitly determined from the design parameters. Unsurprisingly, our gradient optimization techniques can also be applied to this class of problems, which we will demonstrate in the following example. Moreover, we will use the example to show how our method can be applied to transmons and how it can be applied to a chip-level optimization problem. The example is based on adiabatic flux-tuning CPhase gates obtained via ZZ-interactions caused by introducing one of the computational bases near an avoided crossing with a non-computational eigenstate.

For two capacitively coupled tunable transmons, the Hamiltonian has the form

$$H(\varphi_{\text{ext}}) = H_{t,1}(\varphi_{\text{ext}}) + H_{t,2}(0) + J_C n_1 n_2, \quad (6)$$

where $H_{t,i}$ is the transmon Hamiltonian

$$H_{t,i}(\varphi_{\text{ext}}) = 4E_{C,i}n_i^2 - \frac{1}{2}E_{J,i,\text{eff}}(\varphi_{\text{ext}}) \sum_n (|n\rangle\langle n+1| + \text{h.c.}), \quad (7)$$

with $E_{J,i,\text{eff}}(\varphi_{\text{ext}}) = E_{J,i} |\cos(\varphi_{\text{ext}}/2)|$. For simplicity, we have set all effective offset charges to be $n_g = 0$.

The desired properties that we will try to optimize in this example are:

- Large ZZ-interactions at the gate operating point;
- Sufficiently small ZZ-interactions at the idle point, so that single qubit gates including the identity gate have good fidelities;
- Long coherence times.

As we approach an avoided crossing between a computational basis and an eigenstate outside the computational subspace, the

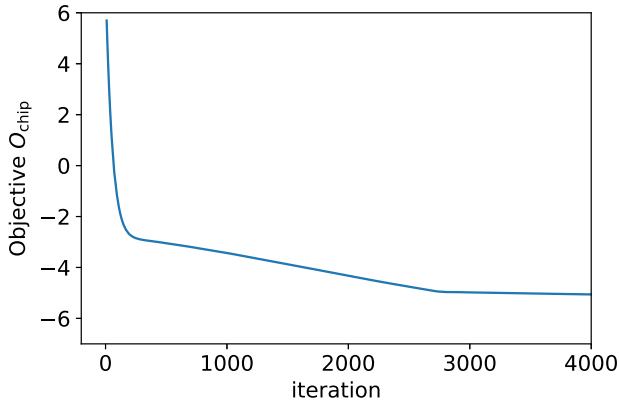


Fig. 5 Value of objective O_{chip} during the optimization. O_{chip} is defined in Eq. (11). The curve is almost flat after 3000 iterations. This suggests we have reached a local minimum or a region with small gradients.

ZZ-interactions will be larger, and we need to change the Hamiltonian slower to avoid leakage. For typical Hamiltonian parameters, we cannot reach the avoided crossing point adiabatically in a reasonable time compared to the coherence time. Therefore, as a first attempt, we use the following routine to estimate the gate operating points:

1. Initialize the flux $\varphi_{\text{ext}} = 0$ to the idle points.
2. Changing the flux $\varphi_{\text{ext}} + \Delta\varphi \rightarrow \varphi_{\text{ext}}$.
3. Use the following stop condition Eq. (8) to check whether it is close to the avoided crossing. If the stop condition is satisfied, we set $\varphi_{\text{ext}}^{\text{stop}} = \varphi_{\text{ext}}$ and use $\varphi_{\text{ext}}^{\text{stop}}$ to compute the objective function Eq. (10). If not, go back to step 2.

We choose the stop condition to be based on the overlap between the instantaneous eigenstates $\{\psi_i(\varphi_{\text{ext}})\}$ and the bare states $\{\Psi_{jk}(\varphi_{\text{ext}})\}$, where $1 \leq i \leq 4$ and $1 \leq j, k \leq 2$. The instantaneous eigenstates $\{\psi_i(\varphi_{\text{ext}})\}$ are the eigenstates of the total Hamiltonian $H(\varphi_{\text{ext}})$, and the bare states $\{\Psi_{jk}(\varphi_{\text{ext}})\}$ are the tensor products of j th and k th single transmon eigenstates. The stop condition is then

$$\min_i \sum_{jk} |\langle \psi_i(\varphi_{\text{ext}}) | \Psi_{jk}(\varphi_{\text{ext}}) \rangle|^2 < M, \quad (8)$$

where we set $M = 0.8$ in this example. Then, we can use the procedure specified in Supplementary Note VI to compute the gradient of the above algorithm with a conditional loop.

The magnitudes of the ZZ interactions at the idle point and the gate operating point are computed from the energy of four computational basis states:

$$E_{ZZ}(\varphi_{\text{ext}}) = |E_{00} + E_{11} - E_{01} - E_{10}|, \quad (9)$$

where E_{jk} are the eigenvalues corresponding to instantaneous eigenstates by $\{\psi_i(\varphi_{\text{ext}})\}$, which has the largest overlap with $\{\Psi_{jk}(\varphi_{\text{ext}})\}$. Because we never reach the middle of the avoided crossing, the above E_{jk} 's can be defined unambiguously. Again, we will include a few additional terms in the objective function to estimate the infidelity caused by decoherence and ensure the functionality of single qubit operations. The objective function for minimization is

$$O = E_{ZZ}(\varphi_{\text{ext}}^{\text{stop}}) / P_{\text{decoh}} \cdot C_{\text{scale}} + P_{ZZ, \text{idle}} + \sum_i (P_{\text{tm}, i} + P_{\text{ah}, i}) + P_{\text{fDiff}}, \quad (10)$$

where P_{decoh} is a penalty determined by the coherence time, $P_{ZZ, \text{idle}}$ punishes large ZZ-interactions when qubits are at sweet spots, $P_{\text{tm}, i}$ ensures the E_J/E_C ratio is large, $P_{\text{ah}, i}$ requires the anharmonicity to be large enough for single qubit gates, and P_{fDiff} forces the $0 \leftrightarrow 1$ frequencies of the two transmons to be

different. The exact formulas for these terms can be found in Supplementary Note VII. In the above equation, $\varphi_{\text{ext}}^{\text{stop}}$ is determined by Eq. (8).

We can extend this optimization problem to a chip-level setting. We consider the surface code scheme presented in ref. ³³. In short, the plan is to have three types of transmon qubits with different parameters arranged in a certain way on a 2D lattice. Sorting by their $0 \leftrightarrow 1$ transition frequencies, we label the three different types of transmon qubits as H(igh), M(iddle), and L(ow). We want to have a CPhase gate between the H-M and M-L qubit pairs, while we still want the crosstalk to be small when idling. Since H and L transmons are spatially separated, we will assume their interactions are small and ignore them in this example. Therefore, we will add the objective function for H-M and H-L qubit pairs together:

$$O_{\text{chip}} = O_{\text{H-M}} + O_{\text{M-L}} - P_{\text{tm}, \text{M}} - P_{\text{ah}, \text{M}}. \quad (11)$$

We need to subtract the last two terms because they are counted twice in $O_{\text{H-M}}$ and $O_{\text{M-L}}$. We can then use Adam²⁸ to perform the optimization. The loss during optimization is shown in Fig. 5, and details about the optimization can be found in Supplementary Note VII.

DISCUSSION

In this work, we extend the optimal control framework to superconducting circuit parameters. We showcase that we can optimize over the device and control parameters jointly for a figure of merit reflecting a design goal through reverse-mode gradient computation. This suggests that there is almost no downside in terms of the efficiency to add device parameters to the optimization variables.

We can extend the proposed framework to other stages of quantum processor design or to cover more realistic scenarios, for example:

1. Robust design. We give an example of joint optimization with uncertainties in control parameters, but in practice, the device parameters can also be off-target. In “optimization with uncertainties of device parameters”, we provide one method to modify the objective function to include the variation of device parameters. Another way to handle inaccuracies in the fabrication of superconducting quantum processors is using laser annealing³⁴ to modify the device parameters after measuring them in the experiments.
2. Electromagnetic simulation. The Hamiltonian representation with device parameters such as E_C , E_J , and E_L we discussed in the paper is an intermediate representation when designing a quantum processor. This representation is derived from the underlying lumped-element circuit representation, which is derived from the processor's two-dimensional (2D)/three-dimensional (3D) layout representation. The lumped-element circuit or layout representations could have more freedom or design choices. Moreover, the layout representation allows us to estimate substantial losses, such as the dielectric loss, based on the electric field energy distribution on the interfaces with contamination. However, electromagnetic simulations, which are used to evaluate the layout design, are very computationally expensive. This means that the gradient evaluation of the Hamiltonian parameters for the layout parameters may be of substantial importance, as they will significantly reduce the number of iterations. We leave this for future study.
3. Task-oriented optimization. Most optimal control use cases focus on optimizing control solutions to realize a given quantum operation. However, when designing a large quantum processors, we need to consider multiple types of quantum operations acting on different qubits. Apart from unitary gates, we also need to take other basic

operations like reset and readout into account. To write down the objective function for optimization, we need to first choose a concrete application such as a NISQ (noisy intermediate-scale quantum) algorithm or an implementation of quantum error correction. When considering NISQ applications, the overall performance will also depend on native gates available on the device and the compilation scheme used. In “adiabatic flux-tuning gates with transmon qubits”, we illustrate how we can simultaneously optimize the ZZ-interaction on-off ratios on different pairs of transmon qubits on a square lattice amenable to fault-tolerant quantum error correction architectures such as the surface code. This can be viewed as the first step toward task-oriented optimization for chip design and control.

METHODS

Efficient gradient computation for both device and control parameters

Here, we discuss an efficient method for computing the gradient of both the device and control parameters. It is efficient in the sense that the gradient is computed in a single run. Assuming the time T_{fwd} needed for computing objective function $O(\vec{x})$ is estimated in terms of FLOPs where a FLOP is a floating point multiply-add operation, then the time T_{grad} for computing the gradient ∇O satisfies

$$T_{\text{grad}} < mT_{\text{fwd}}, \quad (12)$$

where m is a constant and independent of the dimension of \vec{x} ³⁰. However, if one use finite difference $O(\vec{x} + \Delta\vec{x}) - O(\vec{x})$ to estimate ∇O , one need to evaluate O for $n = \dim(\vec{x}) + 1$ times. Therefore, the relative speedup n/m grows linearly with the number of parameters $\dim(\vec{x})$. Please refer to Supplementary Note II for more details about the comparison.

For a superconducting processor using qubits as its basic components, the total Hilbert space can be decomposed correspondingly as $\mathcal{H} = \bigotimes_i \mathcal{H}_i$, where each \mathcal{H}_i is typically an infinite-dimensional Hilbert space. The time-independent Hamiltonian H_{dev} in Eq. (1) can be typically described by the Hamiltonian

$$H_{\text{dev}}(\vec{h}) = \sum_i H_i(\vec{h}) + \sum_{ij} g_{ij}(\vec{h}) S_i(\vec{h}) S_j(\vec{h}), \quad (13)$$

where H_i and S_i are single component Hamiltonian and coupling operators defined on \mathcal{H}_i , respectively. The concrete form of S_i depends on the type of coupling between qubits. For notational simplicity, we assume that there is a single type of coupling, but our computation method naturally generalizes to the situation where there are multiple types. We will also assume that there is a single control operator $C_i(\vec{h})$ (see Eq. (1)) for each subsystem \mathcal{H}_i .

Even though each component \mathcal{H}_i has infinite dimensions, typically only the lowest few energy eigenstates will participate in the computation. This is indeed the case for the examples presented in this work. Therefore, an efficient method for performing numerical simulations is to first truncate the total Hilbert space into this relevant subspace $\bigotimes_i \mathcal{H}'_i$, where \mathcal{H}'_i is the low energy subspace under the above assumption. This can be done by diagonalizing $H_i(\vec{h})$. We can then use the operators from projecting H_i , S_i , and C_i into the subspace \mathcal{H}'_i as an approximation and obtain an efficient finite dimensional Hamiltonian H' for numerical computation. Both the Hamiltonian diagonalization and the Hilbert space truncation can be made differentiable. Please refer to Supplementary Note II for technical details. Therefore, our approach extend the GRAPE algorithm⁵ to superconducting circuit device parameters.

The evolved unitary operator at time τ can be then derived by solving the ordinary differential equation (ODE)

$$i\hbar \frac{d}{dt} U(t) = H(t)U(t). \quad (14)$$

After we compute $U(\tau)$, we can use objective functions such as the average gate fidelity³⁵ to compute how close $U(\tau)$ and the desired operator U_{target} are. In general, we need to add more terms to the objective

functions to ensure the devices are practical. For now, we assume that it is easy to perform the reverse-mode differentiation for the computation steps to compute the objective function $O(\vec{h}, \vec{c})$ from $U(\tau)$. To make the whole design workflow differentiable, we still need to perform reverse-mode differentiation through the ODE solver. The adjoint sensitivity method³⁶ makes this possible. This method computes gradients by solving a second, augmented ODE backwards in time, independent of the parameter size. With the gradients computed in reverse mode, we can now optimize over the device and control parameters jointly and efficiently for an objective $O(\vec{h}, \vec{c})$.

Optimizing device and control parameters for robust control

The proposed framework may also be used to optimize for solutions robustly against control noise. Robust optimization over control parameters has been extensively studied^{37–39}. Here, we take a sample-based approach^{37,38} and define

$$O_{\text{avg}}(\vec{h}, \vec{c}) = \frac{1}{N} \sum_{i=1}^N O(\vec{h}, \vec{c} + \vec{\Delta c}_i), \quad (15)$$

where $O(\vec{h}, \vec{c})$ is the original design goal, e.g., the fidelity or another figure of merit, N is the number of samples, and $\vec{\Delta c}_i$ is the deviation of \vec{c} caused by noise in the i -th sample. Optimizing O_{avg} will make parameters (\vec{h}, \vec{c}) robust against the selected noise. Since we can compute the gradient of $O(\vec{h}, \vec{c})$, we can compute the gradient of $O_{\text{avg}}(\vec{h}, \vec{c})$ by summing over the gradients of $O(\vec{h}, \vec{c} + \vec{\Delta c}_i)$. In this way, we can also use the gradient information to speed up the robustness optimization.

Optimization with uncertainties of device parameters

Likely, we cannot design and manufacture quantum processors with the precisely desired device parameters \vec{h} . Thus, the robustness against variation of \vec{h} should be considered in the proposed optimization framework. As a first attempt, by assuming accurate control, let us define an objective function similar to Eq. (15)

$$O_{\text{avg}}(\vec{h}, \vec{c}) = \frac{1}{N} \sum_{i=1}^N O(\vec{h} + \vec{\Delta h}_i, \vec{c}), \quad (16)$$

where $\vec{\Delta h}_i$ are possible variations of \vec{h} sampled from some probability distribution. However, this is not the most reasonable objective, because role of device and control parameters are not symmetric. Rather, we have a chance to calibrate control parameters after we find out the value of real device parameters $\vec{h} + \vec{\Delta h}_i$. Assuming the calibration is perfect such that we can find a local minimum for the control parameters

$$\vec{c}_i = \arg \min_{\vec{c}} O(\vec{h} + \vec{\Delta h}_i, \vec{c}). \quad (17)$$

Then a more reasonable objective function is

$$O_{\text{avg}}(\vec{h}) = \frac{1}{N} \sum_{i=1}^N \min_{\vec{c}_i} O(\vec{h} + \vec{\Delta h}_i, \vec{c}_i). \quad (18)$$

Gradient optimization of nested objective functions has been studied in literature (see ref. ^{40,41} and references therein). Here, we simply take a similar approach as the previous problem Eq. (15). Let us define

$$f(\vec{h}, \{\vec{c}_i\}) = \frac{1}{N} \sum_{i=1}^N O(\vec{h} + \vec{\Delta h}_i, \vec{c}_i). \quad (19)$$

We can transform the optimization of Eq. (18) $\min O_{\text{avg}}(\vec{h})$ to a slightly different problem

$$\min_{\vec{h}, \{\vec{c}_i\}} f(\vec{h}, \{\vec{c}_i\}). \quad (20)$$

If indeed the above optimization Eq. (20) converges to a local minimum, then we know

$$\frac{\partial f}{\partial c_i} = 0, \quad \text{for } i = 1, \dots, N. \quad (21)$$

Therefore, \vec{c}_i achieves a local minimum of $O(\vec{h} + \vec{\Delta h}_i, \vec{c})$, and \vec{h} from optimization Eq. (20) is also a local minimum of the objective Eq. (18). The optimization Eq. (20) can be done very similar to the optimization of Eq. (15).

The only difference is that now we will optimize multiple sets of control parameters. However, we can still compute the gradient with respect to all sets of control parameters $\{\vec{C}_i\}$ in one run of reverse-mode differentiation and each iteration will take a similar time as optimizing Eq. (15) if N is the same.

DATA AVAILABILITY

All data used to create the figures in the main text as well as in the Appendix is available from the corresponding author upon request.

CODE AVAILABILITY

The computer code used to generate these results is available from the corresponding author upon request.

Received: 29 January 2022; Accepted: 5 August 2022;

Published online: 09 September 2022

REFERENCES

- Menke, T. et al. Automated design of superconducting circuits and its application to 4-local couplers. *NPJ Quantum Inf.* **7**, 1–8 (2021).
- Gyenis, A. et al. Moving beyond the Transmon: Noise-Protected Superconducting Quantum Circuits. *PRX Quantum* **2**, 030101 (2021).
- Mineev, Z. K. et al. Qiskit metal: an open-source framework for quantum device design & analysis. <https://doi.org/10.5281/zenodo.4618153> (2021).
- Liu, F.-M. et al. Quantum design for advanced qubits. Preprint at <https://arxiv.org/abs/2109.00994> (2021).
- Khanuja, N., Reiss, T., Kehlet, C., Schulte-Herbrüggen, T. & Glaser, S. J. Optimal control of coupled spin dynamics: design of NMR pulse sequences by gradient ascent algorithms. *J. Magn. Reson.* **172**, 296–305 (2005).
- Machnes, S., Assémat, E., Tannor, D. & Wilhelm, F. K. Tunable, flexible, and efficient optimization of control pulses for practical qubits. *Phys. Rev. Lett.* **120**, 150401 (2018).
- Egger, D. J. et al. Qiskit dynamics. <https://github.com/Qiskit/qiskit-dynamics> (2021).
- Yamamoto, T., Pashkin, Y. A., Astafiev, O., Nakamura, Y. & Tsai, J. S. Demonstration of conditional gate operation using superconducting charge qubits. *Nature* **425**, 941–944 (2003).
- Chiorescu, I., Nakamura, Y., Harmans, C. J. P. M. & Mooij, J. E. Coherent quantum dynamics of a superconducting flux qubit. *Science* **299**, 1869–1871 (2003).
- Izmalkov, A. et al. Evidence for entangled states of two coupled flux qubits. *Phys. Rev. Lett.* **93**, 037003 (2004).
- Plourde, B. L. T. et al. Entangling flux qubits with a bipolar dynamic inductance. *Phys. Rev. B* **70**, 140501 (2004).
- Koch, J. et al. Charge-insensitive qubit design derived from the Cooper pair box. *Phys. Rev. A* **76**, 042319 (2007).
- Majer, J. et al. Coupling superconducting qubits via a cavity bus. *Nature* **449**, 443–447 (2007).
- Barends, R. et al. Superconducting quantum circuits at the surface code threshold for fault tolerance. *Nature* **508**, 500–503 (2014).
- Manucharyan, V. E., Koch, J., Glazman, L. I. & Devoret, M. H. Fluxonium: single cooper-pair circuit free of charge offsets. *Science* **326**, 113–116 (2009).
- Yan, F. et al. Tunable coupling scheme for implementing high-fidelity two-qubit gates. *Phys. Rev. Appl.* **10**, 054062 (2018).
- Ofek, N. et al. Extending the lifetime of a quantum bit with error correction in superconducting circuits. *Nature* **536**, 441–445 (2016).
- Chu, J. & Yan, F. Coupler-assisted controlled-phase gate with enhanced adiabaticity. *Phys. Rev. Appl.* **16**, 054020 (2021).
- Goerz, M. H., Motzoi, F., Whaley, K. B. & Koch, C. P. Charting the circuit QED design landscape using optimal control theory. *NPJ Quantum Inf.* **3**, 1–10 (2017).
- Heeres, R. W. et al. Implementing a universal gate set on a logical qubit encoded in an oscillator. *Nat. Commun.* **8**, 94 (2017).
- Hu, L. et al. Quantum error correction and universal gate set operation on a binomial bosonic logical qubit. *Nat. Phys.* **15**, 503–508 (2019).
- Wang, Z., Rajabzadeh, T., Lee, N. & Safavi-Naeini, A. H. Automated discovery of autonomous quantum error correction schemes. *PRX Quantum* **3**, 020302 (2022).
- Rumelhart, D. E., Hinton, G. E. & Williams, R. J. Learning representations by back-propagating errors. *Nature* **323**, 533–536 (1986).
- DiCarlo, L. et al. Demonstration of two-qubit algorithms with a superconducting quantum processor. *Nature* **460**, 240–244 (2009).
- Stehlik, J. et al. Tunable coupling architecture for fixed-frequency transmon superconducting qubits. *Phys. Rev. Lett.* **127**, 080505 (2021).
- McKay, D. C., Wood, C. J., Sheldon, S., Chow, J. M. & Gambetta, J. M. Efficient $\$Z\$$ gates for quantum computing. *Phys. Rev. A* **96**, 022330 (2017).
- Nguyen, L. B. et al. High-coherence fluxonium qubit. *Phys. Rev. X* **9**, 041041 (2019).
- Kingma, D. P. & Ba, J. Adam: A method for stochastic optimization. In Bengio, Y. & LeCun, Y. (eds) *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7–9, 2015, Conference Track Proceedings* (2015).
- Bao, F. et al. Fluxonium: an alternative qubit platform for high-fidelity operations. *Phys. Rev. Lett.* **129**, 010502 (2022).
- Baur, W. & Strassen, V. The complexity of partial derivatives. *Theor. Comput. Sci.* **22**, 317–330 (1983).
- Rol, M. A. et al. Fast, high-fidelity conditional-phase gate exploiting leakage interference in weakly anharmonic superconducting qubits. *Phys. Rev. Lett.* **123**, 120502 (2019).
- Sung, Y. et al. Realization of high-fidelity CZ and Z Z -Free iSWAP gates with a tunable coupler. *Phys. Rev. X* **11**, 021058 (2021).
- Versluijs, R. et al. Scalable quantum circuit and control for a superconducting surface code. *Phys. Rev. Appl.* **8**, 034021 (2017).
- Hertzberg, J. B. et al. Laser-annealing Josephson junctions for yielding scaled-up superconducting quantum processors. *NPJ Quantum Inf.* **7**, 1–8 (2021).
- Nielsen, M. A. & Chuang, I. L. Quantum computation and quantum information (Cambridge University Press, 2000).
- Pontryagin, L. S., Mishchenko, E., Boltyanskii, V. & Gamkrelidze, R. The mathematical theory of optimal processes (Wiley, 1962).
- Zhang, J., Greenman, L., Deng, X. & Whaley, K. B. Robust control pulses design for electron shuttling in solid-state devices. *IEEE Trans. Cont. Syst. Technol.* **22**, 2354–2359 (2014).
- Chen, C., Dong, D., Long, R., Petersen, I. R. & Rabitz, H. A. Sampling-based learning control of inhomogeneous quantum ensembles. *Phys. Rev. A* **89**, 023402 (2014).
- Kabytayev, C. et al. Robustness of composite pulses to time-dependent control noise. *Phys. Rev. A* **90**, 012316 (2014).
- Chen, T., Sun, Y. & Yin, W. Closing the gap: tighter analysis of alternating stochastic gradient methods for bilevel problems. In *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6–14, 2021, virtual*, 25294–25307 (2021).
- Blondel, M. et al. Efficient and Modular Implicit Differentiation. Preprint at <https://arXiv.org/abs/2105.15183> (2021).

ACKNOWLEDGEMENTS

The authors would like to thank Dawei Ding and Fei Yan for their constructive comments and Yaoyun Shi for his encouragement and proof-reading of the manuscript. We are also grateful to the referees for their constructive input. L.W. is supported by the Strategic Priority Research Program of the Chinese Academy of Sciences under Grant No. XDB30000000 and National Natural Science Foundation of China under Grant No. T2121001.

AUTHOR CONTRIBUTIONS

X.N. conceived the project and developed gradient computation and optimization code. H.Z. and F.W. developed the code for simulating fluxonium and transmon systems. L.W. and J.C. provided high-level advice on gradient computation and objective-oriented optimization. X.N. and J.C. led the writing of the manuscript. All authors read and approved the final manuscript.

COMPETING INTERESTS

Alibaba Singapore Holding Private Limited has filed patent application CN202110512605.9 for inventor X.N. The authors declare no other competing interests.

ADDITIONAL INFORMATION

Supplementary information The online version contains supplementary material available at <https://doi.org/10.1038/s41534-022-00614-3>.

Correspondence and requests for materials should be addressed to Xiaotong Ni.

Reprints and permission information is available at <http://www.nature.com/reprints>

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this license, visit <http://creativecommons.org/licenses/by/4.0/>.

© The Author(s) 2022