Efficient control pulses for continuous quantum gate families through coordinated re-optimization

Jason Chadwick* and Frederic T. Chong†
Department of Computer Science, University of Chicago, Chicago, IL 60637, USA
(Dated: February 3, 2023)

We present a general method to quickly generate high-fidelity control pulses for any continuously-parameterized set of quantum gates after calibrating a small number of reference pulses. We pick several reference operations in the gate family of interest and directly optimize pulses that implement these operations, then iteratively re-optimize the pulses to guide their shapes to be similar for operations that are closely related. A straightforward interpolation method can then obtain high-fidelity pulses for arbitrary operations in the continuous set. We demonstrate this procedure on the three-parameter Cartan decomposition of two-qubit gates to obtain control pulses for any two-qubit gate, up to single-qubit operations, with consistently high fidelity. The method generalizes to any number of gate parameters and can be used with any pulse optimization algorithm.

Quantum circuits, consisting of logical operations on qubits, are typically decomposed into a set of elementary basis operations that are specific to a given hardware device. These basis operations can be individually calibrated through control pulse shaping to achieve high accuracy. However, the space of all quantum operations is much larger than just the hardware basis set, meaning that operations must typically be decomposed into a sequence of basis operations. The ability to natively perform any arbitrary operation with high fidelity could significantly improve the capabilities of near-term quantum computers, avoiding the runtime and fidelity costs associated with decomposing into basis operations.

Quantum control optimization improves the fidelity of specific operations on a device by shaping control pulses to execute a target operation with high fidelity. Many software packages have been designed to solve quantum control problems for various systems and objectives [1–9]. However, pulse optimizations are computationally expensive, making it infeasible to directly optimize pulses for any possible operation that may be encountered when running a circuit.

We address this problem by describing a procedure to calibrate a pulse landscape for a continuous family of operations, from which control pulses for arbitrary operations can be instantly retrieved. We create this landscape by picking a small number of specific operations to directly optimize pulses for, re-optimizing these pulses to be similar to one another, and then defining an interpolation function to retrieve new pulses for any operation in between.

Generally, the same operation can be achieved with many distinct physical pulse sequences. Given this observation, there is no reason to believe that two arbitrary control pulses for different operations could simply be interpolated between to execute an intermediate operation. Figure 1 captures the key result of our work. The top plots visualize the fidelities of interpolated pulses for operations in the Weyl chamber, which contains the three-parameter decomposition of every two-qubit gate (up to single-qubit operations). While simply optimizing the reference pulses individually (left) yields poor interpolation accuracy, re-optimizing these reference pulses to be similar to each other (right) provides significant improvements in average fidelity. Once these reference pulses are optimized, interpolation can instantly retrieve the pulse for any operation in the parameter space.

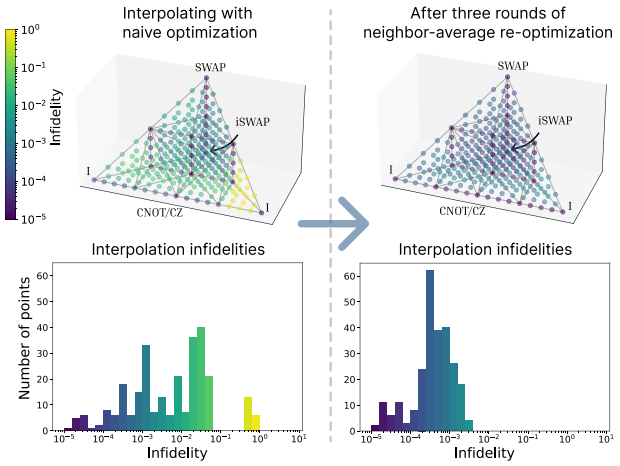

FIG. 1. Example of the re-optimization scheme. Axes on the top plots correspond to the three Cartan coordinates of two-qubit gates. Reference pulses are generated by optimal control software for 14 reference points in parameter space. Pulses for any point in the chamber can be obtained by linear interpolation between reference pulses. *Left*: interpolated pulse infidelities at 285 test points after initial optimization of reference points. Black lines connect reference points and each colored point represents an interpolated pulse. The yellow-colored region indicates poor interpolation quality. Mean infidelity is $5.1 \pm 15 \times 10^{-2}$. *Right*: results after repeatedly re-optimizing each reference point to be near the average of its neighbors. Mean infidelity is improved to $5.5 \pm 5.7 \times 10^{-4}$.

The idea of generating pulses for operations in some continuous parameter space is not new. Several groups have worked on rescaling control pulses to realize rotations of arbitrary angle [10, 11] or to apply a single oper-

ation on a parameterized Hamiltonian [12] to create resilience to varying device errors. Ref. [13] examines the single-parameter case in the optimization of a cubic interaction unitary, performing interpolation between two fixed control pulses in a single-parameter space with consistently high fidelity. Ref. [14] introduces polynomial interpolation to generate continuous sets of controls to create robustness to experimental deviations.

Ref. [15] (from which we borrow the term "gate family") demonstrates the effectiveness of a neural network for generating control pulses for families of parameterized gates. The network takes as inputs the parameters of a specific gate and the time value t, and outputs the control value(s) $f(t)$. The network is trained over many iterations consisting of batches of randomly-sampled operations from the gate family.

In this work, we build upon the work of Ref. [15] by proposing an alternative approach to solving the same problem of efficiently generating control pulses for continuously-parameterized quantum operations. Our method can achieve similar average infidelities using less computation. Additionally, our method can leverage advances in pulse optimization algorithms to improve the translation between simulation and experiment. Existing techniques for robust or closed-loop pulse control such as [7–9, 16, 17] can directly replace or augment the optimal control unit in our method.

We specifically present an application of our method to the set of parameterized two-qubit gates that lie within the Weyl chamber under the belief that this class of gates is the most important for enabling near-term applications on quantum computers. However, like [15], our method can be generalized to any parameterized gate family.

We use the Boulder Opal optimization package [18, 19]. We describe a control pulse by its vector of optimizable variables $\vec{\alpha}$. We use the objective function

$$J = \underbrace{1 - \frac{1}{h^2}\left|\mathrm{Tr}\left(U_{\text{target}}^{\dagger} U_T\right)\right|^2}_{\text{gate infidelity}} + \underbrace{\widetilde{\lambda} \sum_{k}^{n_f} \|\vec{\alpha}_k - \vec{\alpha}_{0,k}\|_2^2}_{\text{Tikhonov regularization}}, \quad (1)$$

with Tikhonov weight

$$\widetilde{\lambda} = \frac{\lambda}{n_f \cdot n_p \cdot \alpha_{\max}^2}. \quad (2)$$

Gate infidelity is calculated between the target unitary $U_{\text{target}}^{\dagger}$ and the realized unitary U_T. Two-qubit gates have Hilbert space dimension $h = 4$. The Tikhonov regularization term penalizes distance between n_f optimizable pulses, each with n_p optimization variables stored in the vector $\vec{\alpha}_k$, and fixed target vector $\vec{\alpha}_{0,k}$ with user-specified weight λ. It is common to include a similar regularization term in the cost function (with $\vec{\alpha}_0 = \vec{0}$) to encourage low-amplitude control pulses, as in [20]. In this work, we found $\lambda = 10^{-2}$ to yield satisfactory results, but this will likely require tuning for different devices and gate families. Choice of $\vec{\alpha}_0$ is key to our method and is discussed in detail later.

To create the interpolation landscape, we first pick reference points p_k on a rectangular grid in parameter space, where the grid spacing is a tunable parameter that we refer to as the granularity. For each point (t_x, t_y, t_z), we obtain the corresponding unitary operation using (6) and then use the pulse optimizer to find a valid control pulse sequence that implements it. For this initial round of optimization, we choose Tikhonov target $\vec{\alpha}_0 = \vec{0}$ in (1) to encourage low-amplitude pulses.

As seen in Figure 1a, this initial pulse set does not always interpolate well in all parts of parameter space (our interpolation method is discussed later in the text). There is no guarantee that control pulses for similar unitaries will themselves be similar, as observed in Ref. [15]; there are often many distinct control sequences that can realize the same unitary operation, yielding near-useless interpolations in between different reference pulses. The previously-mentioned Tikhonov regularization with $\vec{\alpha}_0 = \vec{0}$ helps by restricting the space of allowed solutions, but does not fully resolve this issue. This motivates a more intelligent approach to finding reference pulses, which we call *neighbor-average re-optimization*.

A d-simplex is the simplest d-dimensional polytope, formed by the convex hull of $d+1$ vertices. For example, a 2-simplex is a trangle and a 3-simplex is a tetrahedron. For a set of $k \geq d+1$ points in d dimensions, a *simplicial mesh* of contiguous d-simplices can generally [21] be generated to cover the space between the points. Each point is the vertex of one or more simplices in the mesh.

We generate a simplicial mesh across our reference points using the `scipy.spatial.Delaunay` function [22]. From this mesh, we can query the *neighbors* of a given reference point, which are all other points that are connected to the point of interest by an edge.

Starting with the initial naive reference pulses from the initial optimization round, we generate a simplicial mesh over the reference points. For each reference point p_i, we find the set of neighboring vertices $\eta(p_i)$ and calculate the *neighbor-average* pulse vector

$$\widehat{\vec{\alpha}}_i = \frac{1}{|\eta(p_i)|} \sum_{p_j \in \eta(p_i)} \vec{\alpha}_j, \quad (3)$$

which is the average over the pulse variables $\vec{\alpha}_j$ of all neighboring points p_j. We then calculate the Tikhonov penalty between the optimized pulse $\vec{\alpha}_i$ and the neighbor-average pulse $\widehat{\vec{\alpha}}_i$.

We sort the optimized points in descending order of neighbor-average Tikhonov penalty. For each point in this order, we recalculate $\widehat{\vec{\alpha}}_i$ (taking into account all optimized pulses up to this point) and then re-optimize $\vec{\alpha}_i$ using cost function (1), with initial guess and $\vec{\alpha}_0$ both set to be $\widehat{\vec{\alpha}}_i$. The Tikhonov regularization term encourages

the optimized pulse $\vec{\alpha}_i$ to be as similar as possible to $\widehat{\vec{\alpha}}_i$ while still yielding low infidelity.

This tune-up procedure can be repeated multiple times, with each round of re-optimization building upon the last and steering the reference pulses to be increasingly similar to each other.

To calculate the interpolated control pulse for some new point $\widetilde{p}$, we construct a weighted sum of reference pulses $\vec{\alpha}_i$ at nearby reference points $\{p_i\}$. We first locate the point within one of the simplices in the same Delaunay mesh as used for neighbor-average re-optimization. The interpolated pulse vector is then a linear combination of the reference pulses at the vertices of this simplex, weighted by barycentric coordinates.

There are $d+1$ barycentric coordinates of a point within a d-dimensional simplex, each corresponding to a vertex. Barycentric coordinates are uniquely determined by the requirement that the target point is equal to the coordinate-weighted sum of the vertices,

$$\widetilde{p} = \sum_{p_i \in S_{\widetilde{p}}} b_i p_i, \tag{4}$$

where b_i is the barycentric coordinate of $\widetilde{p}$ with respect to vertex p_i and $S_{\widetilde{p}}$ is the set of vertices defining the simplex that contains point $\widetilde{p}$. The closer the point is to a given vertex, the larger the corresponding coordinate will be [23]. Each coordinate is bounded by $[0,1]$ and the coordinates sum to 1.

Given these coordinates, our interpolated pulse vector is determined by

$$\vec{\alpha}_{\widetilde{p}} = \sum_{p_i \in S_{\widetilde{p}}} b_i \vec{\alpha}_i \tag{5}$$

where the vector $\vec{\alpha}_i$ is the optimized reference pulse at vertex p_i. This barycentric coordinate approach is the generalization of linear interpolation to arbitrary dimension.

We demonstrate this re-optimization and interpolation scheme for the three-parameter Cartan decomposition of two-qubit gates. Any two-qubit quantum logic gate $U \in SU(4)$ can be written as

$$U = k_1 \exp\left(-i\frac{\pi}{2}\sum_{j=x,y,z} t_j \sigma_j^{(1)}\sigma_j^{(2)}\right) k_2 \tag{6}$$

in terms of Pauli matrices $\sigma_x, \sigma_y, \sigma_z$ and Cartan coordinates t_x, t_y, t_z [24]. The operations $k_1, k_2 \in SU(2) \otimes SU(2)$ represent single-qubit gates acting on the qubits independently, i.e. $k_1 = U_0 \otimes U_1$ for some single-qubit gates U_0 and U_1. Distinct two-qubit gates are referred to as "locally equivalent" or "equivalent up to single-qubit gates" if they have the same Cartan coordinates.

The Weyl chamber is a tetrahedron in Cartan coordinate space defined by the equations

$$\begin{aligned} 0 &\le t_x \le 1 \\ 0 &\le t_y \le \min(t_x, 1-t_x) \\ 0 &\le t_z \le t_y. \end{aligned} \tag{7}$$

The Weyl chamber contains the Cartan coordinates of every two-qubit gate, making it a compelling example to demonstrate our interpolation method; the ability to perform any operation in the Weyl chamber with high fidelity using a single pulse could significantly improve the performance of existing quantum computers. For a more in-depth introduction to Cartan coordinates we refer the reader to Ref. [25]. In this work, we demonstrate our interpolation scheme for operations of the form (6) with $k_1 = k_2 = I$ and the set of all coordinates that lie within the Weyl chamber (7).

To evaluate the performance of our approach in comparison to [15], we use the same two-qubit Hamiltonian

$$H(t) = f_{xx}^{\vec{\alpha}}(t)\sigma_x^{(1)}\sigma_x^{(2)} + \sum_{j=1}^{2} f_{jy}^{\vec{\alpha}}(t)\sigma_y^{(j)} + f_{jz}^{\vec{\alpha}}(t)\sigma_z^{(j)} \tag{8}$$

for specific parameter values $\vec{\alpha}$. The 5 control functions $f(t)$ are each restricted to values in $[-1,1]$. We opti-

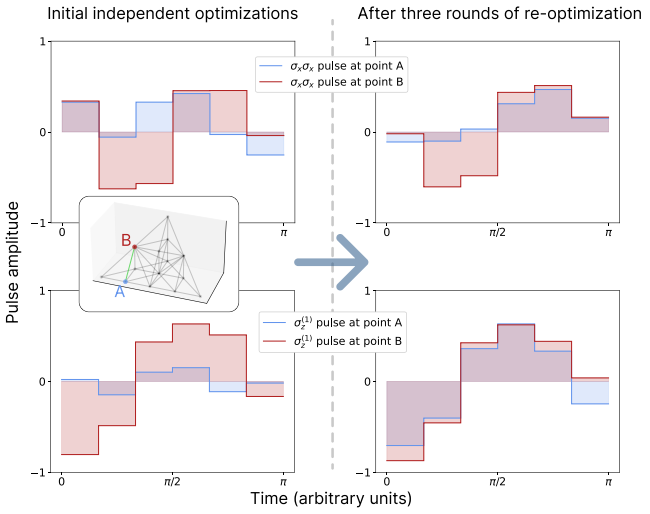

FIG. 2. Comparison of pulse shapes for two adjacent reference points before and after several rounds of re-optimization. Point A corresponds to Cartan coordinates $(\frac{1}{4}, 0, 0)$ and point B corresponds to coordinates $(\frac{1}{4}, \frac{1}{4}, \frac{1}{4})$ in the same results as displayed in Figure 1. Controls from the two points are shown for the $\sigma_x\sigma_x$ control (top) and $\sigma_z^{(1)}$ control (bottom) of the Hamiltonian (Equation (8)). *Inset*: locations of points A and B in the Weyl chamber. *Left*: the pulse shapes are initially significantly different between points A and B. *Right*: the pulses become far more similar after re-optimization, making interpolation easier, but still retain certain differences in their shapes that account for the differences in the resulting operations.

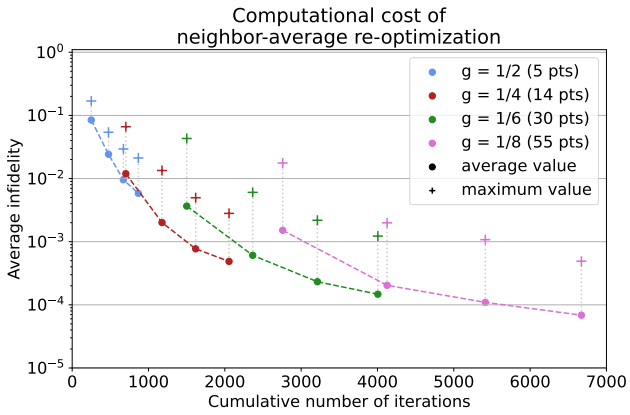

FIG. 3. Different reference point granularities translate to varying amounts of classical computation time needed to optimize all reference pulses. Consecutive points with the same granularity g correspond to subsequent re-optimization rounds, which can further improve average (and maximum) infidelity at the cost of more optimizer iterations. Each optimization iteration corresponds to one system evolution. The points indicated by the plus sign indicate the worst infidelity of any test point.

mize reference pulses using the procedures described previously and then test the interpolation quality by evaluating the infidelities of interpolated pulses for new operations in the space. We perform several rounds of neighbor-average re-optimization and track the improvement in average infidelity. For details on procedure, see Supplemental Material [26].

In Figure 2 we examine two specific control pulses for two different reference points in the Weyl chamber (for the same re-optimization example displayed in Figure 1). The two points of interest are connected by an edge in the simplicial mesh, and thus are influenced by each other in the neighbor-average re-optimization process. Before the re-optimization steps, the pulse shapes are noticably different, providing an explanation for the lower-quality interpolation in between the pulses. After re-optimization, the pulse shapes are much more similar, although some differences remain (the pulses cannot be identical because they perform different operations). The infidelity at the test point directly between these two points improved from 4.8×10^{-2} initially to 1.7×10^{-3} after the final round of re-optimization.

We evaluate the performance of our method for several different reference point granularities and varying numbers of re-optimization rounds, which both influence the total computation time. The relationship between computation time and average infidelity is shown in Figure 3. We use the cumulative sum of all iteration counts as a measurement of the classical computation time required to calibrate the interpolation landscape. We find that performance generally improves with more computation, whether this is achieved by more rounds of re-optimization or a denser sampling of reference points. The data suggests that computation time is an accurate predictor of interpolation quality regardless of a specific reference point distribution.

Finally, we evaluate our approach in comparison to existing methods. Using the same Hamiltonian model, the neural network approach presented in Ref. [15] uses 51,200 system evolutions in total (400 training iterations with 128 sample points per iteration) and achieves infidelities of $4 \pm 4 \times 10^{-4}$ for the 3-parameter Cartan gate with parameter bounds $t_x, t_y, t_z \in [0, 1]$ (a space much larger than the Weyl chamber [26]). In comparison, for the same Hamiltonian and same parameter space, we obtain $3.55 \pm 6.72 \times 10^{-4}$ using 34,407 system evolutions, which can be refined to $2.54 \pm 4.33 \times 10^{-4}$ with an additional round of optimization using 6,882 more iterations. For more details, see the Supplemental Material [26].

Our method thus improves on the work of Ref. [15] by achieving similar pulse accuracy with less total computation required. The procedure has the additional benefits of explainability and modularity. Our method is more explainable because the interpolation space contains known reference points with high-fidelity pulses, which has the benefit that specific points can be re-optimized without needing to recalibrate the entire space. Modularity is important in the case where simple optimal control pulses do not transfer well to experimental settings (which is generally expected due to significant device variations over time [27–29]); more advanced optimization routines such as data-driven or robust methods [7–9, 16, 17] can directly replace the optimal control unit without needing to redesign the rest of the procedure. Our method could also be extended to generate pulses with variable duration (discussed in the Supplemental Material [26]), as is done in Ref. [15].

The Weyl chamber example demonstrates the potential of this calibration routine, allowing the pulses for any two-qubit gates (up to single-qubit operations) to be instantly obtained with infidelities of $2.3 \pm 3.4 \times 10^{-4}$ after only 3213 iterations of the pulse optimizer. On a real device, a similar calibration routine could allow quantum hardware to natively support any two-qubit operation directly at the pulse level, which could yield significant improvements in both execution time and accuracy by avoiding expensive gate decompositions.

Our neighbor-average re-optimization procedure makes no assumptions about the structure of the parameter space of interest, and thus may not be optimal; indeed, we expect that better-motivated approaches [26] could potentially yield higher-accuracy or more efficient interpolations by taking into account the structure of the gate family of interest. The specific method we use in this paper is easily-implemented and conceptually simple, so we provide it as a baseline for future improvements.

We have described a method to generate a continuous manifold of control pulses which are constrained to be as similar as possible to each other. In addition to integrating robust pulse optimization methods, it may be beneficial to recalibrate this manifold over time to account for device variation over time [27–29]. We suspect that this recalibration could be less expensive than individually recalibrating each reference pulse due to the high degree of similarity among nearby pulses (e.g. Figure 2), and leave this as an interesting direction for future research.

Python code used to generate the results in this work is publicly available in a Github repository [30].

We thank Lennart Maximilian Seifert and Andy Goldschmidt for helpful discussions on quantum control and the interpolation scheme, and Andre Carvalho and the Q-CTRL team for enabling our use of their software tools in this paper. This work is funded in part by EPiQC, an NSF Expedition in Computing, under award CCF-1730449; in part by STAQ under award NSF Phy-1818914; in part by the US Department of Energy Office of Advanced Scientific Computing Research, Accelerated Research for Quantum Computing Program; and in part by the NSF Quantum Leap Challenge Institute for Hybrid Quantum Architectures and Networks (NSF Award 2016136) and in part based upon work supported by the U.S. Department of Energy, Office of Science, National Quantum Information Science Research Centers. FTC is Chief Scientist for Quantum Software at Infleqtion and an advisor to Quantum Circuits, Inc.

* jchadwick@uchicago.edu
† chong@cs.uchicago.edu

[1] N. Khaneja, T. Reiss, C. Kehlet, T. Schulte-Herbrüggen, and S. J. Glaser, Journal of Magnetic Resonance **172**, 296 (2005).
[2] P. Doria, T. Calarco, and S. Montangero, Physical Review Letters **106**, 190501 (2011).
[3] T. Caneva, T. Calarco, and S. Montangero, Physical Review A **84**, 022326 (2011).
[4] N. Leung, M. Abdelhafez, J. Koch, and D. Schuster, Physical Review A **95**, 042318 (2017).
[5] N. A. Petersson, F. M. Garcia, A. E. Copeland, Y. L. Rydin, and J. L. DuBois, "Discrete Adjoints for Accurate Numerical Optimization with Application to Quantum Control," (2020), number: arXiv:2001.01013 arXiv:2001.01013 [quant-ph].
[6] S. Günther, N. A. Petersson, and J. L. DuBois, in *2021 IEEE/ACM Second International Workshop on Quantum Computing Software (QCS)* (2021) pp. 88–98.
[7] R.-B. Wu, B. Chu, D. H. Owens, and H. Rabitz, Physical Review A **97**, 042122 (2018).
[8] C. H. Valahu, I. Apostolatos, S. Weidt, and W. K. Hensinger, Journal of Physics B: Atomic, Molecular and Optical Physics **55**, 204003 (2022).
[9] A. J. Goldschmidt, J. L. DuBois, S. L. Brunton, and J. N. Kutz, Quantum **6**, 837 (2022), arXiv:2201.05266 [quant-ph].
[10] P. Gokhale, A. Javadi-Abhari, N. Earnest, Y. Shi, and F. T. Chong, in *2020 53rd Annual IEEE/ACM International Symposium on Microarchitecture (MICRO)* (2020) pp. 186–200.
[11] N. Earnest, C. Tornow, and D. J. Egger, Physical Review Research **3**, 043088 (2021).
[12] P. Luchi, F. Turro, V. Amitrano, F. Pederiva, X. Wu, K. Wendt, J. L. Dubois, and S. Quaglioni, "Control optimization for parametric hamiltonians by pulse reconstruction," (2022), arXiv:2102.12316 [quant-ph].
[13] Y. Shi, A. R. Castelli, X. Wu, I. Joseph, V. Geyko, F. R. Graziani, S. B. Libby, J. B. Parker, Y. J. Rosen, L. A. Martinez, and J. L. DuBois, Physical Review A **103**, 062608 (2021), arXiv:2004.06885 [physics, physics:quant-ph].
[14] J.-S. Li, J. Ruths, T.-Y. Yu, H. Arthanari, and G. Wagner, Proceedings of the National Academy of Sciences **108**, 1879 (2011).
[15] F. Sauvage and F. Mintert, Physical Review Letters **129**, 050507 (2022).
[16] P. H. Leung, K. A. Landsman, C. Figgatt, N. M. Linke, C. Monroe, and K. R. Brown, Physical Review Letters **120**, 020501 (2018).
[17] K. R. Brown, A. W. Harrow, and I. L. Chuang, Physical Review A **70**, 052318 (2004).
[18] H. Ball, M. J. Biercuk, A. R. R. Carvalho, J. Chen, M. Hush, L. A. De Castro, L. Li, P. J. Liebermann, H. J. Slatyer, C. Edmunds, V. Frey, C. Hempel, and A. Milne, Quantum Science and Technology **6**, 044011 (2021).
[19] Q-CTRL, "Boulder Opal," (2022), https://q-ctrl.com/boulder-opal.
[20] N. A. Petersson and F. Garcia, "Optimal Control of Closed Quantum Systems via B-Splines with Carrier Waves," (2021), number: arXiv:2106.14310 arXiv:2106.14310 [quant-ph].
[21] Issues may arise in specific configurations of points. For example, if all points are coplanar in a 3D space, no nonzero-volume tetrahedra can be generated. However, in practice, we can reliably generate a simplicial mesh for any reasonable set of points.
[22] P. Virtanen, R. Gommers, T. E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovski, P. Peterson, W. Weckesser, J. Bright, S. J. van der Walt, M. Brett, J. Wilson, K. J. Millman, N. Mayorov, A. R. J. Nelson, E. Jones, R. Kern, E. Larson, C. J. Carey, I. Polat, Y. Feng, E. W. Moore, J. VanderPlas, D. Laxalde, J. Perktold, R. Cimrman, I. Henriksen, E. A. Quintero, C. R. Harris, A. M. Archibald, A. H. Ribeiro, F. Pedregosa, P. van Mulbregt, SciPy 1.0 Contributors, A. Vijaykumar, A. P. Bardelli, A. Rothberg, A. Hilboll, A. Kloeckner, A. Scopatz, A. Lee, A. Rokem, C. N. Woods, C. Fulton, C. Masson, C. Häggström, C. Fitzgerald, D. A. Nicholson, D. R. Hagen, D. V. Pasechnik, E. Olivetti, E. Martin, E. Wieser, F. Silva, F. Lenders, F. Wilhelm, G. Young, G. A. Price, G.-L. Ingold, G. E. Allen, G. R. Lee, H. Audren, I. Probst, J. P. Dietrich, J. Silterra, J. T. Webber, J. Slavič, J. Nothman, J. Buchner, J. Kulick, J. L. Schönberger, J. V. de Miranda Cardoso, J. Reimer, J. Harrington, J. L. C. Rodríguez, J. Nunez-Iglesias, J. Kuczynski, K. Tritz, M. Thoma, M. Newville, M. Kümmerer, M. Bolingbroke, M. Tartre, M. Pak, N. J. Smith, N. Nowaczyk,

N. Shebanov, O. Pavlyk, P. A. Brodtkorb, P. Lee, R. T. McGibbon, R. Feldbauer, S. Lewis, S. Tygier, S. Sievert, S. Vigna, S. Peterson, S. More, T. Pudlik, T. Oshima, T. J. Pingel, T. P. Robitaille, T. Spura, T. R. Jones, T. Cera, T. Leslie, T. Zito, T. Krauss, U. Upadhyay, Y. O. Halchenko, and Y. Vázquez-Baeza, Nature Methods **17**, 261 (2020).

[23] If the point lies on a face of the simplex, the only nonzero barycentric coordinates will be those corresponding to the vertices that define that face. This property ensures that interpolations are consistent along face boundaries of adjacent simplices. Along with the fact that barycentric coordinates vary continuously within a simplex, this means that the interpolation space is continuous throughout the simplicial mesh.

[24] J. Zhang, J. Vala, S. Sastry, and K. B. Whaley, Physical Review A **67**, 042313 (2003).

[25] G. E. Crooks, "Gates, States, and Circuits," (2022).

[26] See Supplemental Material in following pages.

[27] J. Bylander, S. Gustavsson, F. Yan, F. Yoshihara, K. Harrabi, G. Fitch, D. G. Cory, Y. Nakamura, J.-S. Tsai, and W. D. Oliver, Nature Physics **7**, 565 (2011).

[28] K. W. Chan, W. Huang, C. H. Yang, J. C. C. Hwang, B. Hensen, T. Tanttu, F. E. Hudson, K. M. Itoh, A. Laucht, A. Morello, and A. S. Dzurak, Physical Review Applied **10**, 044017 (2018).

[29] P. Klimov, J. Kelly, Z. Chen, M. Neeley, A. Megrant, B. Burkett, R. Barends, K. Arya, B. Chiaro, Y. Chen, A. Dunsworth, A. Fowler, B. Foxen, C. Gidney, M. Giustina, R. Graff, T. Huang, E. Jeffrey, E. Lucero, J. Mutus, O. Naaman, C. Neill, C. Quintana, P. Roushan, D. Sank, A. Vainsencher, J. Wenner, T. White, S. Boixo, R. Babbush, V. Smelyanskiy, H. Neven, and J. Martinis, Physical Review Letters **121**, 090502 (2018).

[30] J. Chadwick, "pulse-interpolation," (2023).

Supplemental Material: Efficient control pulses for continuous quantum gate families through coordinated re-optimization

Jason Chadwick* and Frederic T. Chong†
Department of Computer Science, University of Chicago, Chicago, IL 60637, USA
(Dated: February 3, 2023)

All example results used in the Main Text as well in this Supplemental Material are obtained using Python code that is available on Github [1]. A valid Q-CTRL license is needed to fully reproduce the results from scratch.

I. OTHER APPROACHES TO REFERENCE OPTIMIZATION AND INTERPOLATION

In this paper, we demonstrate our procedure using the neighbor-average Tikhonov regularization method and simple piecewise-linear interpolation. However, the general idea of creating reference pulses to interpolate between could have many possible implementations. Below, we list several alternative ways of realizing this methodology.

- *Nonuniform reference point distribution.* Reference points need not be distributed on a rectangular grid; the Delaunay triangulation method will work just as well with a nonuniform distribution of points. It may be effective to use more densely-packed reference points in certain parts of parameter space to give better interpolations, or less dense points in other areas to improve calibration efficiency.

- *Different guess methods for Tikhonov re-optimization.* We used the simple neighbor-average method to generate the initial guesses for our Tikhonov regularization re-optimization approach. However, our general approach could also be applied using guesses generated in some other way. For example, each new reference point guess could be a weighted sum of all reference pulses, with weights determined by some notion of distance between the operations.

- *Selective re-optimization.* Some reference points likely do not need to be re-optimized if they already have a high degree of similarity to neighboring reference pulses; computation time could be easily improved by only re-optimizing the points that differ from nearby pulses by more than some amount.

- *Higher-order interpolation fits.* Regardless of optimization method, different methods can be used for the interpolation step of the procedure. In this paper, we show results using a simple piecewise-linear interpolation. Alternatively, interpolation could be done using a higher-order model such as a spline or other function.

- *Minimal-curvature optimization and interpolation.* Instead of running individual optimizations at each reference point, all reference pulses could be tuned at once in one large optimization. We imagine a cost function consisting of the standard infidelity term, averaged over all reference points, as well as a new term that is proportional to the total curvature (or some other metric) of an interpolation function fitted to the reference point pulses. The interpolating function would then be built into the optimization directly, potentially yielding more accurate results.

We do not perform these other interpolation methods for this paper due to the high effectiveness of the simpler linear model and lack of a clearly-motivated choice of a more intricate method. However, our general framework is flexible and will work with any optimization or interpolation method, and we anticipate that more complex methods could potentially yield better results (depending on the system and gate family of interest).

II. EXTENSION: VARIABLE PULSE DURATION

In this work, we assumed that all pulses have the same fixed duration, regardless of the target quantum operation. In reality, it is much easier to perform a gate that is close to the identity than a more complex operation such as SWAP. We expect that these easier gates could be accomplished with a shorter duration within the same pulse amplitude bounds, compared to the more complex gates in the family.

Depending on the chosen pulse description, it can be possible to include the duration of the pulse directly as an optimizable quantity. The duration is then an additional element in the pulse vector and can be included in the interpolation.

Alternatively, when constrained to a specific duration as in this work, we expect that some easier operations

* jchadwick@uchicago.edu
† chong@cs.uchicago.edu

TABLE S1. Weyl chamber interpolation infidelities after three rounds of neighbor-average re-optimization, averaged over 819 test points. Results for smaller numbers of rounds are shown in Figure 3 of the Main Text.

Reference point granularity	Average infidelity	Maximum (worst) infidelity	Total iterations
1/2	$5.83(2.63) \times 10^{-3}$	2.12×10^{-2}	867
1/4	$4.88(5.53) \times 10^{-4}$	2.81×10^{-3}	2055
1/6	$1.47(1.96) \times 10^{-4}$	1.23×10^{-3}	4005
1/8	$6.86(9.59) \times 10^{-5}$	4.91×10^{-4}	6672

TABLE S2. Point granularities used in examples and corresponding number of points.

Point granularity	Number of points in Weyl chamber	Number of points in $[0,1]^3$ box
1/2	5	27
1/4	14	125
1/6	30	343
1/8	55	729
1/12	140	2197
1/16	285	4913
1/24	819	15625

need not make use of the full allowed amplitude of the pulse. In this case, for some Hamiltonians, it is possible to upscale the amplitude and downscale the duration equivalently to obtain a shorter pulse with the same effect. A new cost function term could be added to the optimizations to penalize the maximum value of the control field to improve the amount of downscaling possible.

III. GENERAL METHODOLOGY: COORDINATED RE-OPTIMIZATION

The methodology for applying our general approach (coordinated re-optimization of individual reference pulses) to an arbitrary family of quantum operations is as follows:

1. *Setup.* If using a model-based optimizer, obtain a model of the device, such as a Hamiltonian. Choose a pulse description $\vec{\alpha}$ (a finite set of variables used to construct each pulse) and a pulse optimization algorithm. Define the parameters of the gate family and determine the space to interpolate within. Sample a number of parameter points $\{p_i\}$ from this space and obtain the corresponding quantum operations.

2. *Initial optimization.* Use the optimizer to generate initial reference pulses $\vec{\alpha}_i$ for each reference point.

3. *Re-optimization.* For each reference point p_i:

   (a) Calculate some new target pulse $\vec{\alpha}_{0,i}$ based on the set of existing reference pulses.

   (b) Re-optimize the reference pulse with this target pulse as the initial guess. Use Tikhonov regularization in the cost function (or some other method) to encourage the final pulse to be close to the target pulse.

   Repeat this step as needed.

4. *Interpolation.* Choose an interpolation function $f : (\widetilde{p}, \{p_i, \vec{\alpha}_i\}) \to \vec{\alpha}_{\widetilde{p}}$ that calculates interpolated pulse $\vec{\alpha}_{\widetilde{p}}$ at parameter-space point $\widetilde{p}$ given the set of optimized reference points and pulses $\{p_i, \vec{\alpha}_i\}$.

IV. DETAILS ON WEYL CHAMBER EXAMPLES FROM MAIN TEXT

For all examples, we fix the pulse duration to π and parameterize each of the 5 control functions $f(t)$ as a piecewise-constant function of 6 segments, yielding 30 total optimizable parameters. We set the Tikhonov regularization weight $\lambda = 10^{-2}$. The pulse optimizer is run for a maximum of 50 iterations for each individual optimization (or re-optimization), although it often terminates early upon reaching convergence, especially for later rounds of re-optimization where the pulse shapes do not change as drastically. We found this to be sufficient for convergence over multiple re-optimization rounds, and it also avoids unnecessary computation during early rounds of optimization (where the pulse shapes need not be finalized). We use the default Q-CTRL convergence criteria [2]. We use various point granularities to uniformly cover the parameter space of interest; the number of points generated by grids of various granularities (for both the Weyl chamber and the larger parameter-space box used to compare to [3]) are shown in Table S2.

For the example used in Figures 1 and 2 in the Main Text, we use a reference point granularity of 1/4 and a test point granularity of 1/16. We initialize the reference points with naive pulse optimizations, run three consecutive rounds of neighbor-average re-optimization, and use the initial and final reference pulses to extract the data shown in the two figures.

To demonstrate the relationship between computational cost and average infidelity (Figure 3 in the Main Text), we use reference point granularities ranging from 1/2 to 1/8 and always use test point granularity of 1/24. We run three rounds of re-optimization for each reference point granularity. After initialization and after each

TABLE S3. Interpolation infidelities for the Cartan coordinate $[0,1]^3$ box after varying rounds of re-optimization.

Number of re-optimization rounds	Average infidelity	Maximum (worst) infidelity	Cumulative total iterations
0	$3.27(16.2) \times 10^{-2}$	1.00×10^{0}	17213
1	$6.62(16.1) \times 10^{-4}$	1.95×10^{-2}	26631
2	$3.55(6.72) \times 10^{-4}$	6.54×10^{-3}	34407
3	$2.54(4.33) \times 10^{-4}$	3.91×10^{-3}	41289

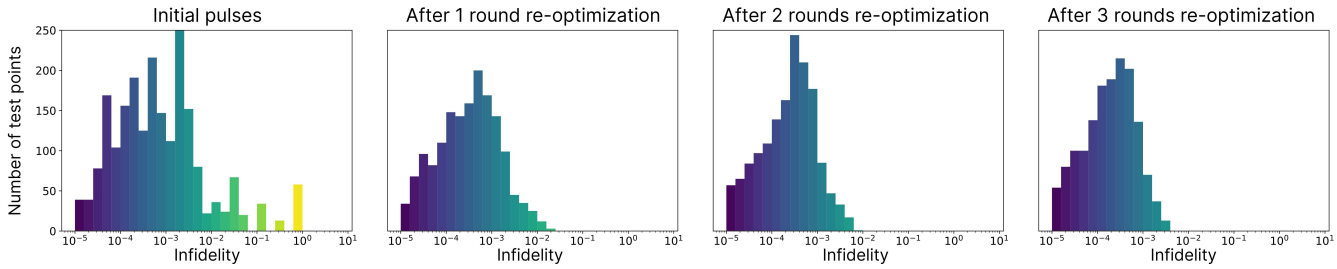

FIG. S1. Infidelities at 2197 test points within the $[0,1]^3$ Cartan coordinate box for 0 to 3 rounds of neighbor-average re-optimization. Average and worst-case infidelities are shown in Table S3.

round of re-optimization, we record the average and maximum infidelities achieved over all test points. These values are displayed in Figure 3 of the Main Text. Table S1 shows specific values for the average and worst-case infidelities achieved after the final round of re-optimization. Increasing reference point granularity appears to significantly improve interpolation quality, but at the cost of more pulse optimizer iterations.

V. COMPARISON TO EXISTING METHODS

In this section, we describe our methodology for comparing the approach presented in this work with that of [3], which to the best of our knowledge is the only other work that addresses the problem of pulse generation for multi-parameter continuous sets of gates. It is critical to minimize the computational overhead of this problem to enable a future experimental implementation.

A. Setup

We specifically compare 3-parameter interpolation for the 2-qubit gate family described by

$$U = \exp\left(-i\frac{\pi}{2} \sum_{j=x,y,z} t_j \sigma_j^{(1)} \sigma_j^{(2)} \right) \tag{1}$$

where t_x, t_y, t_z are the Cartan coordinates of U. This is the same unitary as used in the Main Text for the Weyl chamber example, but for a fair comparison with the results of Ref. [3], we extend the parameter bounds to $t_x, t_y, t_z \in [0,1]$. This occupies a volume 24 times larger than the Weyl chamber in parameter space. However, we note that the Weyl chamber alone can already represent any two-qubit operation (up to single-qubit operations), so this extra volume in parameter space only serves to increase computational complexity without adding any practical benefit.

We discretize each of the five control pulses into 6 piecewise-constant segments of equal duration. As in the main paper, we use the Hamiltonian

$$H(t) = f_{xx}^{\vec{\alpha}}(t)\sigma_x^{(1)}\sigma_x^{(2)} + \sum_{j=2}^{2} f_{jy}^{\vec{\alpha}}(t)\sigma_y^{(i)} + f_{jz}^{\vec{\alpha}}(t)\sigma_z^{(i)} \tag{2}$$

and set the pulse duration to π.

We pick reference points on a grid in parameter space with spacing 1/6. We find that the optimization and interpolation methods discussed in the Main Text provide good results with little need for fine-tuning.

B. Results

Interpolation results are shown in Figure S1 for various numbers of re-optimization rounds. Infidelities are evaluated on a grid in parameter space with granularity 1/12. Large regions of parameter space appear to already interpolate reasonably well; we attribute this to the constraints we have imposed on the pulses, namely the small number of optimizable parameters (each pulse consisting of only 6 piecewise-constant segments) and the initial Tikhonov regularization that favors low-amplitude pulses.

Using a neural network, Ref. [3] reports average pulse infidelity of $4 \pm 4 \times 10^{-4}$ while using 51,200 system

evolutions to train the network. As shown in Table S3, our method reaches an average pulse infidelity of $3.55 \pm 6.72 \times 10^{-4}$ in 34,407 total iterations, which can further be refined to $2.54 \pm 4.33 \times 10^{-4}$ in 41,289 total iterations. Each iteration corresponds to a system evolution.

This demonstrates that our method can obtain similar or better results than previous methods using less computation, with the additional benefits (as discussed in the Main Text) of improved explainability and modularity.

We expect that computation time could be further reduced in several simple ways, if desired. For example, the cost value convergence threshold of the pulse optimizer could be reduced, since the reference pulses are re-optimized several times and intermediate results are thus not required to be perfect. Additionally, as mentioned in the first section of this Supplemental Material, some reference points may not need to be re-optimized as many times as others.

[1] J. Chadwick, pulse-interpolation (2023).
[2] H. Ball, M. J. Biercuk, A. R. R. Carvalho, J. Chen, M. Hush, L. A. De Castro, L. Li, P. J. Liebermann, H. J. Slatyer, C. Edmunds, V. Frey, C. Hempel, and A. Milne, Software tools for quantum control: improving quantum computer performance through noise and error suppression, Quantum Science and Technology **6**, 044011 (2021).
[3] F. Sauvage and F. Mintert, Optimal Control of Families of Quantum Gates, Physical Review Letters **129**, 050507 (2022).