Tunable, Flexible and Efficient Optimization of Control Pulses for Practical Superconducting Qubits - Supplementary Material

Shai Machnes,[1,2,*] Elie Assémat,[1] David Tannor,[2] and Frank Wilhelm[1]

[1] *Theoretical Physics, Saarland University, 66123 Saarbrücken, Germany*

[2] *Weizmann Institute of Science, 76100 Rehovot*

APPENDIX A - GRADIENT-FREE VS. GRADIENT-BASED SEARCH: NELDER-MEAD VS. GOAT USING L-BFGS

In situations when both gradient-free and gradient-based optimization may be used (e.g. the typical QOC open-loop optimization, with which we are dealing in this work), the critical criteria for deciding whether to make use of one or the other is the efficiency with which one may compute the gradient of the goal function. It is well known that when the gradient can be computed quickly, gradient-based methods, such as L-BFGS, can be orders-of-magnitude quicker than gradient-free methods [1].

The most commonly used gradient-free optimizer is Nelder-Mead [2]. We shall therefore compare it to GOAT, which, in this context, can be viewed as a method of computing the goal function gradient, coupled to a standard gradient-based search, L-BFGS [3].

To demonstrate this superiority of GOAT using L-BFGS (GOAT being the method of computing the gradient and L-BFGS the gradient-driven search algorithm) vs. Nelder-Mead, we consider a toy model of a 2 qubit xmon system [4] (which me model as qutrits), and the resonator is sufficiently detuned to be neglected. The Hamiltonian is

$$H_0 = g_{12}(b_1^\dagger b_2 + b_1 b_2^\dagger) + \sum_{k=1}^{2}(\omega_k - \frac{\delta_k}{2})b_k^\dagger b_k + \frac{\delta_k}{2} b_k^\dagger b_k b_k^\dagger b_k \tag{1}$$

$$H_l^c = b_l^\dagger b_l \tag{2}$$

where ω_k is the frequency of the Xmon k and δ_k is its anharmonicity, g_{12} is the coupling between two neighbouring qutrits. The parametrization used is a 500-slice PWC and the gate generated is a CNOT. Numeric values are $\omega_1 = 5.23 \times 2\pi$ GHz, $\omega_2 = 5.78 \times 2\pi$ GHz, $\delta_1 = -0.220 \times 2\pi$ GHz, $\delta_2 = -0.210 \times 2\pi$ GHz, $g_{12} = 0.030 \times 2\pi$ GHz.

Examining Fig. 1 it is clear the Nelder-Mead is not competitive with GOAT for open-loop optimization, as GOAT is capable of efficiently computing and utilizing the gradient of the goal value with respect to the controls, allowing the use of L-BFGS to efficiently search the parameter space. Nelder-Mead, however, struggles when the number of parameters grows. It is important to stress that the results shown in Fig. 1 are universal, with little dependence on system specifics. This is a demonstration of the general attributes of gradient-driven vs. gradient-free search algorithms, when an efficient ay of computing the gradient is available - a fact which is well-known in the general optimization community. A more detailed discussion of the subject is outside the scope of this work.

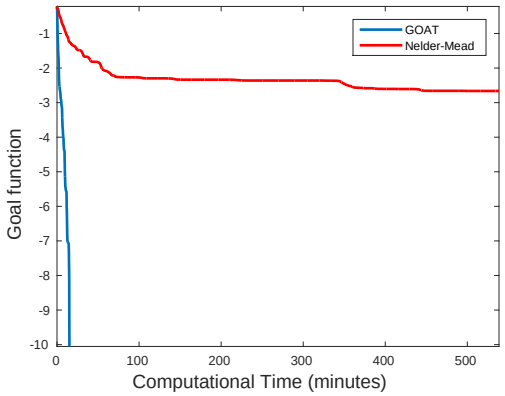

FIG. 1. Performance of GOAT vs. Nelder-Mead search, as applied to a CNOT gate generation on Xmon system [4]. Details of the system appear in eq. 2. It is evident that GOAT optimization is faster than Nelder-Mead by several orders of magnitude. These results are not surprising, as it is well understood in the computer science community that if one can efficiently compute the gradients of the goal function with respect to the parametrization (as GOAT can), it is far preferable to conduct a gradient-driven search. To contrast, in closed-loop pulse calibration the gradient is not accessible, in which case gradient-free methods are the only available option.

APPENDIX B - AVOIDING PIECEWISE CONSTANT APPROXIMATION OF CONTROL FIELDS

Approximating smooth drive shapes by piecewise-constant (PWC) functions introduces inaccuracies into the simulated dynamics [5, 6]. This approximation is acceptable if one is seeking low or medium accuracy (infidelity of 10^{-3} or below). However, when targeting high fidelities, this approach becomes untenable. To illustrate the point, we simulated the dynamics of a system with a 3-spin random drift Hamiltonian, driven by a second random control Hamiltonian. The control field is parameterized by 11 bandwidth-limited Fourier components, so as to generate a low-frequency, smooth control field. See Fig. 2 (top).

We compare the unitary propagator generated by the TDSE $\partial_t U(t) = (H_0 + c(t) H_c) U(t)$, as computed by a high-accuracy Runge-Kutta 4/5 ODE solver, to integrations of PWC approximations of the control field $c(t)$ at various resolutions.

In all PWC approximations, the constant Hamiltonian assigned to each slice is taken by sampling $c(t)$ at the middle of each slice. The error, as a function of the number of pieces used to approximate the smooth drive, is presented in Fig. 2 (bottom).

We observe that, for example, more than $\mathcal{O}\left(10^4\right)$ time steps are necessary to achieve an error below 10^{-5}. At such a high step-count, it is numerically advantageous for forgo the PWC approximation and instead fall back on the more accurate Runge-Kutta integrators. Moreover, regardless of questions of numerical expediency in integration, such a representation of the control signal would imply that optimization is carried out in a very high dimensional search space, leading to convergence problems. Even for more sophisticated versions of GRAPE [7], the exceedingly large Jacobian required to move from the PWC gradient to a more compact parametrization makes the algorithm problematic, as pointed out in [8]. Or if the latter approach is not taken, clearly such a huge number of parameters is not calibratable.

APPENDIX C - CONTROL PARAMETRIZATION

The following is a detailed description of the parametrization chain, including both discrete parameter transformations and control function wrappers, used in the flux-tunable coupler [9] iSWAP optimization described in the main text. We recall the Hamiltonian

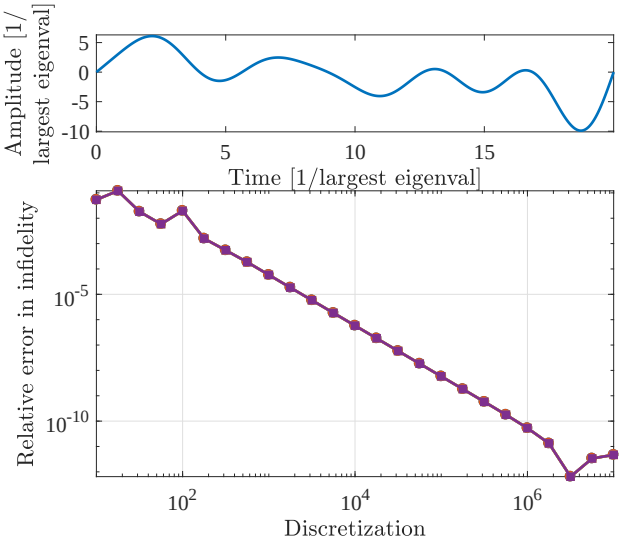

FIG. 2. Accuracy implications of the piecewise-constant approximation when simulating the dynamics of a 3-spin system with random drift and control Hamiltonians and a smooth control function. PWC samples the control field at middle of equal-width time slices. Top: The smooth drive shape, with 1/time and amplitude specified in the units of the largest magnitude eigenvalue. Bottom: Gate error, comparing the Runge Kutta propagation and the PWC approximation, as a function of the number of time slices used to represent the drive shape, $1-\mathrm{Re}\left(\mathrm{Tr}\left(U_{\mathrm{RK}}^{\dagger}U_{\mathrm{PWC}}\right)\right)$. Evidently, in order to achieve high accuracy using the PWC approximation an extraordinarily large number of slices is required.

$$\begin{aligned} H\left(t\right) &= \textstyle\sum_i \omega_i a_i^{\dagger}a_i + \omega_{TB}(t)b^{\dagger}b + g_i(a_i^{\dagger}b + b^{\dagger}a_i) - \alpha\left|2\right\rangle\left\langle 2\right|_i \\ \omega_{TB}\left(t\right) &= \omega_{\mathrm{TB},0}\sqrt{\left|\cos(\pi\Phi/\Phi_0)\right|} \\ \Phi\left(t\right) &= \Theta + \delta\left(t\right)\cos(\omega_{\Phi}t) \end{aligned} \tag{3}$$

Optimization is performed over 18 unconstrained parameters which ultimately describe $\delta\left(t\right)$ using 6 additive frequency components, each with 3 parameters (amplitude, frequency and phase).

The raw optimized parameters are:

$$\begin{array}{rrrrrr} a_{1,1} &=& -0.441194 & a_{1,2} &=& 0.412071 & a_{1,3} &=& -0.126278 \\ a_{2,1} &=& -1.57074 & a_{2,2} &=& 0.532019 & a_{2,3} &=& 0.134298 \\ a_{3,1} &=& 0.43699 & a_{3,2} &=& 0.603436 & a_{3,3} &=& 0.461675 \\ a_{4,1} &=& -1.04331 & a_{4,2} &=& 0.490129 & a_{4,3} &=& -0.361516 \\ a_{5,1} &=& -1.16992 & a_{5,2} &=& 0.454957 & a_{5,3} &=& 0.91013 \\ a_{6,1} &=& 0.81492 & a_{6,2} &=& 0.489283 & a_{6,3} &=& 0.128118 \end{array} \tag{4}$$

The parameters are then linearly rescaled:

$$\begin{array}{rrrrr} b_{*,1} &=& L\left(a_{*,1},-1,1\right. & -0.3, & 0.3\ \left.\right) \\ b_{*,2} &=& L\left(a_{*,2},-1,1\right. & -2.0944{\times}10^{9}, & 2.0944{\times}10^{9}\ \left.\right) \\ b_{*,3} &=& L\left(a_{*,3},-1,1\right. & -3141.59, & 3141.59\ \left.\right) \end{array} \tag{5}$$

where

$$L\left(x,a,b,a_2,b_2\right) := \tfrac{b_2-a_2}{b-a}\left(x-\tfrac{b+a}{2}\right)+\tfrac{b_2+a_2}{2}, \tag{6}$$

resulting in

$$\begin{array}{lll} b_{1,1} = -0.132358 & b_{1,2} = 8.6304\times10^{8} & b_{1,3} = -396.713 \\ b_{2,1} = -0.471222 & b_{2,2} = 1.11426\times10^{9} & b_{2,3} = 421.909 \\ b_{3,1} = 0.131097 & b_{3,2} = 1.26383\times10^{9} & b_{3,3} = 1450.39 \\ b_{4,1} = -0.312994 & b_{4,2} = 1.02652\times10^{9} & b_{4,3} = -1135.74 \\ b_{5,1} = -0.350977 & b_{5,2} = 9.5286\times10^{8} & b_{5,3} = 2859.26 \\ b_{6,1} = 0.244476 & b_{6,2} = 1.02475\times10^{9} & b_{6,3} = 402.495 \end{array} \tag{7}$$

Next, the parameters are bound,

$$\begin{array}{l} c_{*,1} = \mathcal{C}\left(b_{*,1},\, -0.3, 0.3\right) \\ c_{*,2} = \mathcal{C}\left(b_{*,2},\, -2.0944\times10^{9}, 2.0944\times10^{9}\right) \\ c_{*,3} = b_{*,3}(\text{unbounded}) \end{array} \tag{8}$$

where the bounding is performed via a scaled sine function:

$$\mathcal{C}\left(x,a,b\right) := \tfrac{b-a}{2}\,\sin\left(\left(x-\tfrac{b+a}{2}\right)/\tfrac{b-a}{2}\right)+\tfrac{b+a}{2}. \tag{9}$$

The resultant parameter are

$$\begin{array}{lll} c_{1,1} = -0.128106 & c_{1,2} = 8.38822\times10^{8} & c_{1,3} = -396.713 \\ c_{2,1} = -0.3 & c_{2,2} = 1.06243\times10^{9} & c_{2,3} = 421.909 \\ c_{3,1} = 0.126964 & c_{3,2} = 1.18852\times10^{9} & c_{3,3} = 1450.39 \\ c_{4,1} = -0.259223 & c_{4,2} = 9.85915\times10^{8} & c_{4,3} = -1135.74 \\ c_{5,1} = -0.276216 & c_{5,2} = 9.20327\times10^{8} & c_{5,3} = 2859.26 \\ c_{6,1} = 0.218301 & c_{6,2} = 9.84352\times10^{8} & c_{6,3} = 402.495 \end{array} \tag{10}$$

These rescaled and bounded parameters are then used by the Fourier parametrization

$$d\left(t\right) = \sum_{k} c_{k,1}\,\sin\left(c_{k,2}\,t + c_{k,3}\right) \tag{11}$$

Plugging in the transformed optimized parameter values we arrive at

$$\begin{array}{rcl} c\left(t\right) = & -0.128106 & \sin\left(\ 1.33503\times10^{8}\ 2\pi\,t\ -0.277691\,\pi\right) \\ & -0.3 & \sin\left(\ 1.69091\times10^{8}\ 2\pi\,t\ +0.297666\,\pi\right) \\ & +0.126964 & \sin\left(\ 1.89158\times10^{8}\ 2\pi\,t\ -0.325411\,\pi\right) \\ & -0.259223 & \sin\left(\ 1.56913\times10^{8}\ 2\pi\,t\ +0.483735\,\pi\right) \\ & -0.276216 & \sin\left(\ 1.46475\times10^{8}\ 2\pi\,t\ +0.130343\,\pi\right) \\ & +0.218301 & \sin\left(\ 1.56664\times10^{8}\ 2\pi\,t\ +0.118224\,\pi\right) \end{array} \tag{12}$$

Next, we bind the total additive amplitude and concurrently ensure the controls smoothly go to zero at the beginning and end of the pulse. The smooth edges are enforced by a product of an ascending sigmoid and time-shifted descending sigmoid, and the amplitude bound is enforced by an additional scaled sigmoid. The explicit function is:

$$\delta\left(t\right) = \mathcal{A}\left(d\left(t\right), t/T_{\text{final}},\, -0.3, 0.3,\, 40, 0.075\right) \tag{13}$$

where

$$\mathcal{A}\left(f\left(t\right),\tau,\, a,b,\, g,\Delta_{\tau}\right) \;:=\; \tilde{\mathcal{S}}_{\uparrow}\left(\tau-\Delta_{\tau},g\right)\tilde{\mathcal{S}}_{\downarrow}\left(\tau-\left(1-\Delta_{\tau}\right),g\right)\mathcal{S}_{\uparrow}\left(-\tfrac{b-a}{2},\tfrac{b-a}{2},\, f\left(t\right)\right) \tag{14}$$

Here $\tau \in [0\ldots1]$ is the scaled time, a and b are the minimal and maximal allowed values, g is the ascent/descent gradient of the window function and Δ_{τ} is the width of the ascent/descent. The first sigmoid ensures zero control at $t=0$; the second sigmoid ensures zero control at $t=T_{\text{final}}$; the third bounds the amplitude to the $[a\ldots b]$ range.

The descending [ascending] sigmoid function $\tilde{\mathcal{S}}_{\downarrow}$ [$\tilde{\mathcal{S}}_{\uparrow}$], with value $\frac{1}{2}$ at $x = 0$ and gradient g_0 [$-g_0$], are defined by

$$\begin{array}{rcl} \tilde{\mathcal{S}}_{\downarrow}(x, g_0) & := & 1/\left(1 + \exp\left((4 \times g_0) \times x\right)\right) \\ \tilde{\mathcal{S}}_{\uparrow}(x, g_0) & := & 1 - \tilde{\mathcal{S}}_{\downarrow}(x, g_0). \end{array} \tag{15}$$

The scaled ascending sigmoid with gradient 1 mid-range is

$$\mathcal{S}_{\uparrow}(a, b, x) := \left(2\tilde{\mathcal{S}}_{\uparrow}\left(\left(y - \tfrac{b+a}{2}\right)/\tfrac{b-a}{2}, \tfrac{1}{2}\right) - 1\right)\tfrac{b-a}{2} + \tfrac{b+a}{2} \tag{16}$$

The bounded $\delta(t)$ of eq. 13 serves as input to the arbitrary waveform generator, where a carrier and bias are added

$$\Phi(t) = \mathcal{G}\left(\delta(t), t, -0.108, 5.34448{\times}10^9\right) \tag{17}$$

where

$$\mathcal{G}(f(t), t, \Theta, \omega_\Phi) := \Theta + \cos(\omega_\Phi t) f(t). \tag{18}$$

Finally, the resultant flux takes the following form as the pre-factor of the control Hamiltonian $b^\dagger b$,

$$\omega_{\mathrm{TB}}(t) = \mathcal{W}\left(\Phi(t), t, 4.67783{\times}10^{10}\right) \tag{19}$$

where

$$\mathcal{W}(\Phi(t), t, \omega_{\mathrm{TB}_0}) := \omega_{\mathrm{TB}_0}\sqrt{|\cos(\pi\Phi(t))|} \tag{20}$$

The gradient of the above function compositions is used by the optimization algorithm.

Should one wishes, it is possible to accurately simulate the effects of an Arbitrary Waveform Generators (AWG) on the generated signal by

- Sampling the smooth control ansatz at specific times (1.2GHz is a commonly used sampling rate).
- Generating PWC output from the AWG.
- Filtering the PWC output with a low-pass filter, which is inherent to all AWGs (0.3Ghz is again a commonly used bandwidth).

Adapting GOAT algorithm to the above model would require

- Generating the Jacobians required for the chain rule describing the discrete sampling $\partial_{\bar{\alpha}} c_k(\bar{\alpha}, t)$.
- The transfer functions will be modeled by combining the equations of motion of the filter's output and its internal state (e.g. the equations of motion for charge and current of an RLC bandpass-filter) with the propagator equation of motion, $\partial_t U(\bar{\alpha}, t)$. If one derives the entire set with respect to $\bar{\alpha}$, one arrives at an expanded version of eq. (7) of the main text; see also [10].

However, the above highly precise modeling of an AWG is not necessary in this case, as the produced pulses are already bandwidth limited to the capabilities of typical AWGs.

APPENDIX D - LOCAL PHASE INSENSITIVE GOAL FUNCTION

Local (single qubit) Z rotations can be accounted for in software by noting the logical rotation of the $x-y$ plane. Therefore, we aim to generate the iSWAP gate modulo Z rotations. The following goal function is insensitive to such operations

$$g = 1 - \frac{1}{d}\mathrm{Tr}\left(|U_{goal}^{\dagger}U(t_f)|^2\right). \tag{21}$$

The gradient is

$$\partial_\alpha g = -\frac{2}{d}\mathrm{Tr}\left(|U_{goal}|\mathrm{Re}(\partial_\alpha U(t_f).*U(t_f)^*)\right) \tag{22}$$

where $.*$ denotes element-wise multiplication. Notice that both the goal function and the gradient are defined element-wise.

This approach can be generalized to any target gate, and will be covered in more detail in forthcoming work.

* shai.machnes@gmail.com

[1] J. Nocedal and S. J. Wright, *Numerical Optimization*, 2nd ed. (Springer, 2006).
[2] J. A. Nelder and R. Mead, The Computer Journal **7**, 308 (1965), /oup/backfile/Content_public/Journal/comjnl/7/4/10.1093/comjnl/7.
[3] J. Nocedal, Mathematics of computation **35**, 773 (1980).
[4] R. Barends, J. Kelly, A. Megrant, D. Sank, E. Jeffrey, Y. Chen, Y. Yin, B. Chiaro, J. Mutus, C. Neill, P. O'Malley, P. Roushan, J. Wenner, T. C. White, A. N. Cleland, and J. M. Martinis, Phys. Rev. Lett. **111**, 080502 (2013).
[5] C. Leforestier, R. Bisseling, C. Cerjan, M. Feit, R. Friesner, A. Guldberg, A. Hammerich, G. Jolicard, W. Karrlein, H.-D. Meyer, N. Lipkin, O. Roncero, and R. Kosloff, Journal of Computational Physics **94**, 59 (1991).
[6] D. Lauvergnat, S. Blasco, X. Chapuisat, and A. Nauts, The Journal of Chemical Physics **126**, 204103 (2007), http://dx.doi.org/10.1063/1.2735315.
[7] T. E. Skinner and N. I. Gershenzon, J. Magn. Reson. **204**, 248 (2010).
[8] E. Zahedinejad, J. Ghosh, and B. C. Sanders, Phys. Rev. Lett. **114**, 200502 (2015).
[9] D. C. McKay, S. Filipp, A. Mezzacapo, E. Magesan, J. M. Chow, and J. M. Gambetta, Phys. Rev. Applied **6**, 064007 (2016).
[10] I. Hincks, C. Granade, T. Borneman, and D. Cory, Physical Review Applied **4**, 024012 (2015).