Analytic Filter-Function Derivatives for Quantum Optimal Control

Isabel Nha Minh Le,* Julian D. Teske, Tobias Hangleiter, Pascal Cerfontaine, and Hendrik Bluhm†

JARA-FIT Institute for Quantum Information, Forschungszentrum Jülich GmbH and RWTH Aachen University, 52074 Aachen, Germany

(Received 18 April 2021; revised 27 November 2021; accepted 17 December 2021; published 2 February 2022)

Autocorrelated noise appears in many solid-state qubit systems and hence needs to be taken into account when developing gate operations for quantum information processing. However, explicitly simulating this kind of noise is often less efficient than approximate methods. Here, we focus on the filter function formalism, which allows the computation of gate fidelities in the presence of autocorrelated classical noise. Hence, this formalism can be combined with optimal control algorithms to design control pulses, which optimally implement quantum gates. To enable the use of gradient-based algorithms with fast convergence, we present analytically derived filter function gradients with respect to control pulse amplitudes, and analyze the computational complexity of our results. When comparing pulse optimization using our derivatives to a gradient-free approach, we find that the gradient-based method is roughly 2 orders of magnitude faster for our test cases. We also provide a modular computational implementation compatible with quantum optimal control packages.

DOI: 10.1103/PhysRevApplied.17.024006

I. INTRODUCTION

Noise leading to the loss of quantum information remains a major challenge in the current development of quantum computers [1–4]. Minimizing decoherence, while still leaving the system accessible for quantum control, stands in the center of quantum optimal control [5–7]. While quasistatic noise can be well addressed in pulse optimization approaches, treating so-called colored noise characterized, e.g., by $1/f^{\alpha}$ spectral noise densities remains a difficult task [8–11]. However, colored noise poses a major noise contribution in many candidate systems for quantum information processing. As such, leading solid-state qubit implementations are subject to colored flux or charge noise [12–18].

The filter function formalism is a suitable and experimentally verified tool to fully describe a quantum system under wide-sense stationary classical noise with arbitrary auto- and cross-correlations [19–22]. A so-called filter function quantifies the noise susceptibility of a quantum channel as a function of noise frequency. Specifically, the average gate infidelity [23,24], a useful metric for the accuracy of quantum operations, is accessible via filter functions. Hence, this formalism can be used to design cost functions for the optimization of control pulses in the presence of correlated noise.

In previous works, colored noise has been taken into account by combining Monte Carlo simulations with Nelder-Mead optimization [25,26]. In addition, filter function gradients have been used for quantum optimal control using gradient-based algorithms. Such algorithms typically require fewer iterations, but the overall performance gain depends on the cost of evaluating gradients. While, for many problems, this cost turns out prohibitively large, it was shown that unitary quantum dynamics allow gradients to be computed rather efficiently [27]. Filter function gradients were calculated either by autodifferentiation [28] or finite differences [29]. Here, we develop analytical filter function gradients, study their computational complexity, and benchmark their application for pulse optimization. This not only allows for a numerically robust and computationally efficient implementation, but also provides insight if and when efficiency gains are expected compared to other methods. Additionally, no specialized software packages, e.g., for automatic differentiation, are required.

This paper is organized as follows. In Sec. II, we introduce the theoretical concepts of the filter function formalism for an easily accessible, but still nontrivial case of a single pulse of one control operator and one noise source. Subsequently, we show how to obtain the derivatives of the filter function for this case in Sec. III (a full but more complex derivation is presented in the Appendix B). We continue to introduce the numerical implementation of the derivatives in Sec. IV, and present an analysis of its computational complexity in Sec. V. In Sec. VI, we apply the newly implemented derivatives to pulse optimization

*isabel.le@rwth-aachen.de
†bluhm@physik.rwth-aachen.de

and compare the results with a gradient-free optimization approach. We conclude in Sec. VII by summarizing our results and giving an outlook.

For the purpose of conciseness, we use the following notation. We denote operators and their matrix representations with italic font, e.g., P, and reserve calligraphic font for quantum operations and their representations, e.g., $\mathcal{P}$. Additionally, we denote the control matrix, which we introduce in the following section, with $\tilde{\mathcal{B}}$, due to its resemblance with the Liouville representation of quantum operations. Operators in the interaction picture are written with an overset tilde, e.g., $\tilde{P} = U^\dagger P U$ with the toggling-frame operator U. Furthermore, we use an overset bar for the matrix representation of operators transformed into the eigenbasis of a Hamiltonian, e.g., $\bar{P} = V^\dagger P V$ with the corresponding unitary matrix of eigenvectors V. A general matrix is denoted with double-stroke font, e.g., $\mathbb{X}$, while its elementwise notation is labeled as $[\mathbb{X}]_{pq}$. Lastly, we denote the identity matrix by $\mathbb{1}$, and set $\hbar \equiv 1$ throughout this work.

II. FILTER FUNCTION FORMALISM FOR A SINGLE PULSE

Before computing the derivatives, we first review the filter function formalism [19–21,24] for the simple, but nontrivial case of a single pulse with one control operator with variable amplitude and one noise source. To this end, we break down the quantum system's Hamiltonian into control and noise contributions, and derive the ensemble average entanglement infidelity as well as the filter function. The latter is a central quantity of the formalism and describes the quantum gate's susceptibility to noise as a function of noise frequency. Since we focus on the infidelity, we summarize the derivation given by Green *et al.* [24], even though more general approaches exist [20,21].

We start by considering a quantum system, whose total Hamiltonian $H(t)$ during $t \in [0, \tau]$ consists of two parts: a control Hamiltonian $H_c(t)$ consisting of a time-independent contribution H_0 and adjustable parameters to achieve the intended quantum operation, and a noise Hamiltonian $H_n(t)$ perturbing $H_c(t)$,

$$\begin{aligned} H(t) &= H_c(t) + H_n(t) \\ &= [H_0 + u(t)A] + b(t)B. \end{aligned} \tag{1}$$

Here, $u(t)$ is a single time-dependent control amplitude with the corresponding control operator A. Noise enters via the random variable $b(t)$ and the corresponding Hermitian noise operator B. The time evolution generated by the total Hamiltonian is described by the unitary operator $U(t) = \exp(-i\int_0^t H(t')dt')$. It is possible to rewrite the total propagator by factoring it into two parts as $U(t) = U_c(t)\tilde{U}(t)$, where $U_c(t)$ contains the control effects, and the unitary operator $\tilde{U}(t)$ captures the effect of a single noise realization. While the control propagator $U_c(t)$ fulfills the noise-free Schrödinger equation $i\partial U_c(t)/\partial t = H_c(t)U_c(t)$, it can be shown [30] that $\tilde{U}(t)$ fulfills the equation of motion $i\partial\tilde{U}(t)/\partial t = \tilde{H}_n(t)\tilde{U}(t)$, with $\tilde{H}_n(t) = U_c^\dagger(t)H_n(t)U_c(t)$ the noise Hamiltonian transformed into the interaction picture of the control Hamiltonian. The generator of $\tilde{U}(t=\tau) \equiv \tilde{U}$ can then be considered as a time-independent effective Hamiltonian H_{eff}, such that

$$\tilde{U} = \exp(-iH_{\text{eff}}\tau). \tag{2}$$

Using the Pauli basis, H_{eff} can be written in terms of an error vector $\vec{\beta}$ as $H_{\text{eff}} = \vec{\beta}\cdot\vec{\sigma}$. Equation (2) can be expanded by using the Magnus expansion [31,32], such that the exponent is given by $H_{\text{eff}} = \sum_{\mu=1}^{\infty}\vec{\beta}_\mu\cdot\vec{\sigma}$ with the first-order term

$$H_{\text{eff},1} = \vec{\beta}_1\cdot\vec{\sigma} = \frac{1}{\tau}\int_0^\tau dt\tilde{H}_n(t). \tag{3}$$

For small noise strength, i.e., sufficiently small $\vec{\beta}_i$, higher orders provide diminishing contributions [21,24].

A suitable measure for the quantum gate accuracy is the ensemble average entanglement infidelity [23], to which we refer to as simply the infidelity $\mathcal{I}$. The infidelity is linked to $\vec{\beta}$ via

$$\mathcal{I} \equiv \langle\mathcal{I}(\tau)\rangle = \tfrac{1}{2}[1 - \langle\cos(2|\vec{\beta}|)\rangle]. \tag{4}$$

By Taylor expanding the cosine term in Eq. (4), the infidelity can be approximated by $\mathcal{I} = \frac{1}{2}\langle|\vec{\beta}|^2\rangle$ for small noise $|\vec{\beta}| \ll 1$. Evaluating this term only requires the square of the error vector $|\vec{\beta}|^2 = (\sum_k \beta_k^2)$. Inserting the first-order term of the Magnus expansion given by Eq. (3) leads to

$$\langle\beta_{1,k}^2\rangle = \int_0^\tau dt_1\int_0^\tau dt_2[\langle b(t_1)b(t_2)\rangle\tilde{\mathcal{B}}_k(t_1)\tilde{\mathcal{B}}_k(t_2)], \tag{5}$$

where $\langle b(t_1)b(t_2)\rangle$ is the noise amplitude's autocorrelation function, and $\tilde{\mathcal{B}}_k(t)$ denotes the control matrix elements in the time domain defined by

$$\tilde{\mathcal{B}}_k(t) = \text{tr}[U_c^\dagger(t)BU_c(t)\sigma_k], \tag{6}$$

where we now consider the matrix representation of the noise operator B. Note that in our case of a single noise contribution, the control matrix reduces to a control vector.

Under the assumption of wide-sense stationary classical noise, the spectral noise density $S(\omega)$ can be defined as the Fourier transform of $\langle b(t_1)b(t_2)\rangle$. Assuming the noise to be Gaussian, which is reasonable in many cases [33], $S(\omega)$ gives a full characterization of the noise. Inserting $S(\omega)$

into Eq. (5) and shifting the Fourier transformation to the control matrix in the frequency domain defined by

$$\tilde{\mathcal{B}}(\omega) = \int_0^\tau dt \tilde{\mathcal{B}}(t) e^{i\omega t} \tag{7}$$

results in $\langle \beta_{1,k}^2 \rangle = \int (d\omega/2\pi) \tilde{\mathcal{B}}_k^*(\omega) S(\omega) \tilde{\mathcal{B}}_k(\omega)$. By summing over the Cartesian coordinates, we obtain

$$\begin{aligned} \mathcal{I} &= \frac{1}{2} \int_{-\infty}^{\infty} \frac{d\omega}{2\pi} \Big(\sum_k |\tilde{\mathcal{B}}_k(\omega)|^2 \Big) S(\omega) \\ &= \frac{1}{2} \int_{-\infty}^{\infty} \frac{d\omega}{2\pi} F(\omega) S(\omega) \end{aligned} \tag{8}$$

with the filter function

$$F(\omega) = \sum_k |\tilde{\mathcal{B}}_k(\omega)|^2. \tag{9}$$

This expression gives a full description of how the noise contribution affects the given quantum channel described by $\tilde{\mathcal{B}}(\omega)$.

III. FILTER FUNCTION DERIVATIVES FOR A SINGLE PULSE

In the following, the filter function gradient with respect to the control amplitude is derived for the previously presented simple case with only a single control variable and noise contribution. To this end, we approximate $u(t)$ for $t \in [0, \tau]$ to be a sequence of $n_{\Delta t}$ piecewise-constant control amplitudes $\{u_1, u_2, \ldots, u_{\Delta t}\}$ and restrict our derivation to the case of a single constant control amplitude $u \in \mathbb{R}$.

Differentiating Eq. (8) requires the filter function derivative with respect to the control amplitude u,

$$\frac{\partial F(\omega)}{\partial u} = 2\Re \Big(\sum_k \tilde{\mathcal{B}}_k^*(\omega) \frac{\partial \tilde{\mathcal{B}}_k(\omega)}{\partial u} \Big). \tag{10}$$

By applying the product rule, the gradient of the control matrix in frequency space, $\partial \tilde{\mathcal{B}}_k(\omega)/\partial u$, is given by the Fourier transform of

$$\frac{\partial \tilde{\mathcal{B}}_k(t)}{\partial u} = \mathrm{tr} \Big(\frac{\partial U_c^\dagger(t)}{\partial u} B U_c(t) \sigma_k + U_c^\dagger(t) B \frac{\partial U_c(t)}{\partial u} \sigma_k \Big). \tag{11}$$

In order to deduce the gradient of the control propagator, $\partial U_c(t)/\partial u$, we consider a small perturbation δu on u and add a corresponding term to the control Hamiltonian, $H_c + H_\delta = (u + \delta u)A$. Let $U_c(t)$ be the control propagator of the unperturbed control Hamiltonian H_c. The perturbation Hamiltonian in the interaction picture defined by H_c is then given by $\tilde{H}_\delta(t) = U_c^\dagger(t)(\delta u A) U_c(t)$. The corresponding Schrödinger equation

$$i \partial \tilde{U}_\delta(t)/\partial u = \tilde{H}_\delta(t) \tilde{U}_\delta(t) \tag{12}$$

defines the perturbation propagator $\tilde{U}_\delta(t)$. A solution of Eq. (12) can be approximated as an exponential operator by using the Magnus expansion up to first order similarly to Eqs. (2) and (3), such that

$$\tilde{U}_\delta(t) = \exp \Big[-i \Big(\sum_{i=1}^{\infty} \Omega_i \Big) \Big], \tag{13a}$$

$$\Omega_1 = \int_0^t dt' \tilde{H}_\delta(t'). \tag{13b}$$

We introduce the basis change matrix V consisting of the eigenvectors of the matrix representation of the control Hamiltonian H_c, which transforms into the eigenbasis of the control Hamiltonian, and indicate an operator in matrix representation P under such a transformation as $\bar{P} = V^\dagger P V$. Since $V V^\dagger = V^\dagger V = \mathbb{1}$, inserting the identity into the exponent given by Eq. (13b) leads to

$$\Omega_1 = V \Big(\int_0^t dt' \bar{U}_c^\dagger(t') \bar{\tilde{H}}_\delta \bar{U}_c(t') \Big) V^\dagger = V \mathbb{K}(t) V^\dagger. \tag{14}$$

Naturally, the control propagator in the eigenbasis of the control Hamiltonian $\bar{U}_c$ is diagonal. Specifically, if we consider the sorted set of eigenvalues of the control Hamiltonian $\{\omega_i\}$ in the same order as the eigenvectors in V, it is given by $\bar{U}_c = \mathrm{diag}(e^{i\omega_i t})$. For the further calculation, we introduce the elementwise notation $[\mathbb{X}]_{pq}$ for the matrix $\mathbb{X}$. The integral $\mathbb{K}$ in Eq. (14) can then be evaluated as

$$\begin{aligned} [\mathbb{K}(t)]_{pq} &= [\bar{\tilde{H}}_\delta]_{pq} \int_0^t dt' \exp[i(\omega_p - \omega_q)t'] \\ &= [\bar{\tilde{H}}_\delta]_{pq} \Big[\delta_{pq} t + (1 - \delta_{pq}) \frac{e^{i(\omega_p - \omega_q)t} - 1}{i(\omega_p - \omega_q)} \Big] \\ &= [\bar{\tilde{H}}_\delta]_{pq} [\mathbb{M}(t)]_{pq}, \end{aligned} \tag{15}$$

where δ_{pq} is the Kronecker delta. We now note that the total propagator of $H_c + H_\delta$ is given by $U(t) = U_c(t) \tilde{U}_\delta(t)$, and that $\tilde{U}_\delta(t) \,|_{\delta u = 0} = \mathbb{1}$. Using this, we can rewrite the control propagator derivative as

$$\frac{\partial U_c(t)}{\partial u} = \frac{\partial U(t)}{\partial (\delta u)} \Big|_{\delta u = 0} \tag{16}$$

In order to calculate $[\partial U(t)/\partial(\delta u)]|_{\delta u = 0}$, we differentiate Eq. (13a) to first order. To this end, we need to take the

derivative of the exponent given by Eq. (14). Taking into account Eq. (15) and recognizing that

$$\frac{\partial \bar{\tilde{H}}_\delta}{\partial(\delta u)} = \frac{\partial}{\partial(\delta u)}[V^\dagger(\delta u A)V^\dagger] = \bar{A}, \tag{17}$$

the derivative of the exponent is given by

$$\frac{\partial \Omega_1}{\partial(\delta u)} = V[\mathbb{M}(t) \circ \bar{A}]V^\dagger, \tag{18}$$

where "$\circ$" symbolizes elementwise multiplication. Considering Eqs. (13)–(18), the derivative of the control propagator with respect to the constant control amplitude can be calculated as

$$\frac{\partial U_c(t)}{\partial u} = -iU_c(t)V[\mathbb{M}(t) \circ \bar{A}]V^\dagger. \tag{19}$$

By inserting Eq. (19) into Eq. (11) and the latter equation into Eq. (10), we have derived a complete analytic form of the filter function gradient for the considered case of a single pulse.

In general, a pulse sequence can consist of several time steps, control contributions, and noise sources. Under the assumption of piecewise-constant control amplitudes, the total control effect on the quantum system is a composition of the single pulse effects. The total control matrix is then given by [20,21]

$$\tilde{\mathcal{B}}(\omega) = \sum_{g=1}^{n_{\Delta t}} e^{i\omega t_{g-1}} \tilde{\mathcal{B}}^{(g)}(\omega)\mathcal{Q}^{(g-1)}, \tag{20}$$

where $\tilde{\mathcal{B}}^{(g)}(\omega)$ are the single pulse control matrices given by Eqs. (6) and (7), and $\mathcal{Q}^{(g-1)}$ are the cumulative control propagators of the individual single pulses, i.e., the control propagators in Liouville representation multiplied with each other up to each time step. A more detailed explanation is given in Appendix A. Consequently, when calculating filter function gradients for a general pulse sequence, products of $\tilde{\mathcal{B}}^{(g)}(\omega)\mathcal{Q}^{(g-1)}$ in the total control matrix need to be taken into account in Eq. (10). To this end, partial derivatives of the cumulative control propagators of the individual pulses have to be deduced additionally. A full calculation of the filter function gradient for a generic pulse sequence is given in Appendix B.

IV. SOFTWARE IMPLEMENTATION

We implemented the filter function gradients derived in the previous section as part of the open-source software framework `filter_functions` [21,34]. This PYTHON package facilitates the efficient numerical calculation of generalized filter functions and derived quantities, such as the infidelity, for given pulse sequences. A pulse sequence is represented by the `PulseSequence` class, from which properties like the filter function or infidelity can directly be computed. The newly implemented module `gradient` enables the calculation of filter function and infidelity derivatives for systems that are subject to classical wide-sense stationary noise, which can be characterized by an arbitrary spectral noise density. The gradients are taken with respect to stepwise constant control amplitudes. The function `infidelity_derivative()` is placed at the user's disposal for direct calculation of the infidelity derivatives for a given pulse sequence. This function can be used directly for quantum optimal control, i.e., to optimize gate fidelities. An illustration of the structure of the implementation can be found in Appendix C, and a verification of the implemented analytical gradients is available in Ref. [34]. Furthermore, we ensured the implementation's compatibility with quantum optimal control packages, such as `qopt` [35,36].

V. COMPUTATIONAL COMPLEXITY

In principle, our software implementation enables the calculation of filter function and infidelity gradients without any constraints on the quantum system's dimension d or on the number of control and noise operators n_c and n_α. Furthermore, a pulse sequence can contain any number of time steps $n_{\Delta t}$ and the number of frequency samples n_ω describing the noise spectral density is completely variable. Naturally, the computational complexity in computing the infidelity derivatives depends on the chosen set of parameters. Theoretical investigations of the implementation lead to the expected scaling behavior summarized in Table I for dominant terms.

We analyze the actual run-time behavior by running the implemented software module with various pulses. To this end, we increase one of the parameters n_α, n_c, n_ω, $n_{\Delta t}$, and d, while fixing the remaining quantities and generating the control amplitudes randomly for each pulse. We then obtain the scaling behavior of the implementation by means of asymptotic fits. A graphical illustration can be seen in Fig. 1 and in Appendix C. Table I contrasts the actual run-time results with their theoretical expectations.

TABLE I. Summary computational complexity.

Parameter	Expectation	Run time
Number of frequency samples n_ω	$\mathcal{O}(n_\omega)$	$\mathcal{O}(n_\omega)$
Number of control operators n_c	$\mathcal{O}(n_c)$	$\mathcal{O}(n_c)$
Number of noise operators n_α	$\mathcal{O}(n_\alpha)$	$\mathcal{O}(n_\alpha)$
Number of time steps $n_{\Delta t}$	$\mathcal{O}(n_{\Delta t}^2)$	$\mathcal{O}(n_{\Delta t}^{1.92})$
Dimension d	$\mathcal{O}(d^{b+4})$[a]	$\mathcal{O}(d^{4.35})$

[a] b arises from the multiplication of two $n \times n$ matrices, which scales polynomially with n^b. For a naive algorithm, $b = 3$ [37] and, for the Coppersmith-Winograd algorithm, $b = 2.376$ [38].

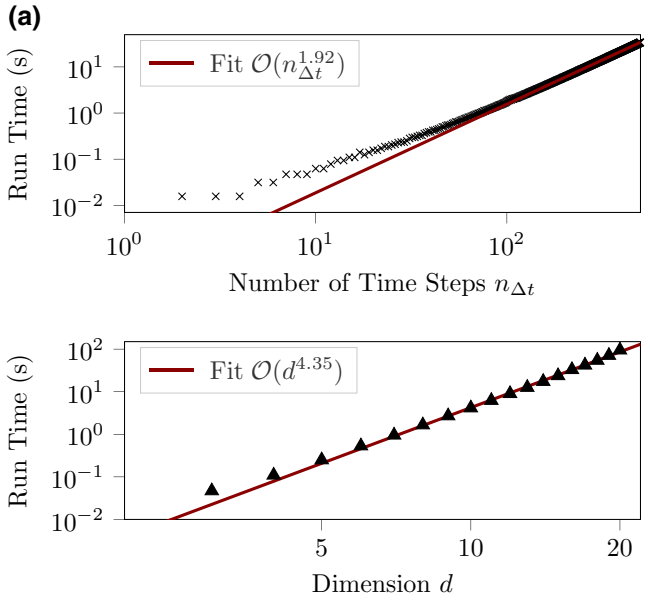

FIG. 1. Run-time behavior for computing the infidelity gradients on a standard desktop computer [39] for (a) various numbers of time steps $n_{\Delta t}$ and (b) various dimensions d. Panel (a) shows a random pulse sequence with dimension $d = 2$, $n_\omega = 200$ frequency samples, $n_c = n_\alpha = 2$ control and noise operators, and $n_d = 1$ drift direction. A fit to the data yields the predicted quadratic scaling behavior. Panel (b) shows the median run time of 50 randomly generated pulse sequences of $n_{\Delta t} = 3$ time steps. The parameters chosen for each random pulse sequence are $n_\omega = 200$ frequency samples, $n_c = n_\alpha = 2$ control and noise operators, and $n_d = 1$ drift direction. A fit to the data yields a polynomial scaling behavior.

The plots confirm the expected linear dependence on n_ω, n_c, and n_α. For various $n_{\Delta t}$ and d, the run-time data are shown in Fig. 1. A power-law fit on the data for various $n_{\Delta t}$ stands in agreement with the expectation of quadratic scaling. Concerning the d dependency, a clear polynomial behavior is visible. Because of memory limitation, we tested dimensions restricted to $d \leq 20$. Within this restriction, lower-order terms in d dominate the scaling behavior, such that this cannot be considered as an asymptotic regime. Therefore, the underestimated exponent for the d dependency does not contradict the theoretical expectation.

The complexity of our implementation can be compared to an alternative implementation using automatic differentiation. A benchmark of an implementation with gradients calculated using JAX [40] can be found in Appendix D. Within the analysis, we demonstrate that JAX computes derivatives faster, but has the same scaling behavior as our implementation. This indicates that the acceleration has to be attributed rather to the integrated just-in-time compilation than the automatic differentiation algorithm.

VI. APPLICATION

As mentioned previously, filter function derivatives lend themselves to gradient-based pulse optimization. In the following, we motivate the use of such gradient-based methods by comparing the numerical optimization of control pulses using filter function gradients to a gradient-free approach.

To this end, we use the filter function derivatives in conjunction with `scipy`'s L-BFGS-B algorithm [41,42] and contrast this strategy with a gradient-free constrained Nelder-Mead method [43,44] in terms of run-time and error rates of the optimized pulses. We conduct both optimization approaches within the quantum optimal control package `qopt` [35,36]. To facilitate a fair comparison, we disabled the internal multithreading of the optimization algorithms in PYTHON.

We apply each technique to a generic four-level quantum system corresponding to the optimization of two-qubit gates. For this purpose, we use a control Hamiltonian given by

$$H_c = \sum_{ij} u_{ij,g} \times \sigma_i \otimes \sigma_j , \tag{21}$$

where $u_{ij,g}$ signifies the piecewise-constant control amplitudes at the discrete time step $g \in (1, 2, \ldots, n_{\Delta t})$ of uniform length Δt. For simplicity, we assume the system to be exposed to exactly one noise source, such that the noise Hamiltonian is given by $H_n(t) = b(t) \times \sigma_0 \otimes \sigma_x$. In the latter, $b(t)$ denotes a noise amplitude corresponding to a pink noise spectral density of the form $S(f) = S_0/f$ with frequency f and constant S_0.

We choose the cost function of the pulse optimization to be the total infidelity $\mathcal{I}$, which is the sum of systematic and noise-induced deviations from the target gate. The former ones are quantified by the standard entanglement infidelity and the latter ones are calculated as in Eq. (8). The optimization problem then lies in the minimization of the infidelity $\mathcal{I}$ by finding the optimal control amplitudes $u_{ij,g}$.

As convergence criterion, we choose that $\mathcal{I}$ improves by less than 10^{-7} within one iteration of the optimization algorithm, which does not favor either algorithm (see Fig. 7 in Appendix E). The performance is depicted in Fig. 2, where we observe that the L-BFGS-B algorithm converges faster and scales better with the number of time steps $n_{\Delta t}$ and the number of control operators n_c. Both algorithms find locally optimal pulses with similar fidelities (see Fig. 6 in Appendix E).

In Appendix B, we discuss the convergence of the two optimization algorithms in greater detail, demonstrating that the Nelder-Mead algorithm offers better global performance, while the L-BFGS-B algorithm is more prone to get stuck in local minima. A further discussion of the benefits of filter functions compared to Monte Carlo methods

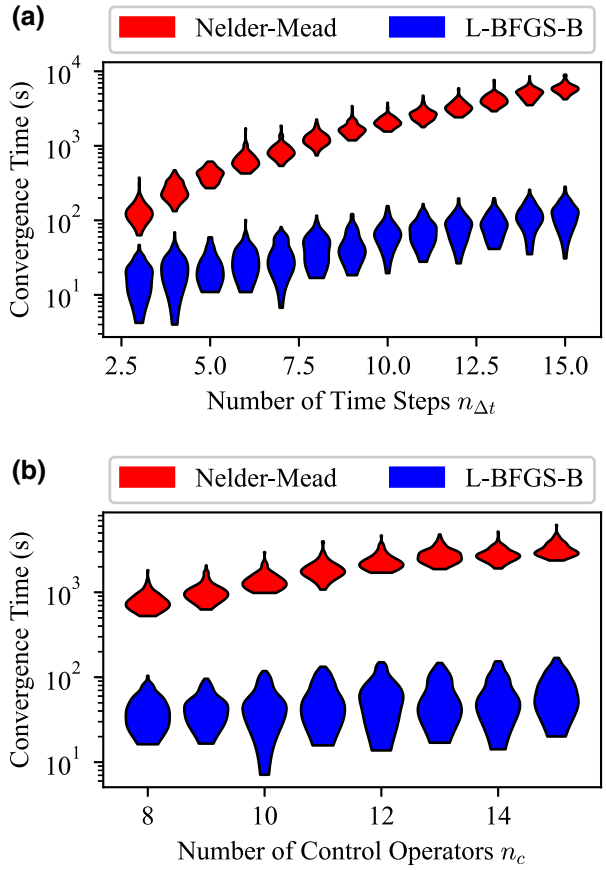

FIG. 2. Convergence time of the analytical derivatives in conjunction with the L-BFGS-B algorithm compared to Nelder-Mead. The distribution of convergence times is depicted by the width of the violin plots, where each violin plot shows the distribution of 100 runs. The convergence time is (a) displayed as a function of the number of time steps, while the number of control operators is constant $n_c = 8$, and (b) as a function of number of control operators for a constant number of time steps $n_{\Delta t} = 6$. In both cases the L-BFGS-B algorithm converges faster to optimal pulses with about the same infidelity as the Nelder Mead algorithm and scales better with an increasing number of degrees of freedom.

and the description of open quantum systems by master equations can be found in Ref. [36].

VII. CONCLUSION AND OUTLOOK

In this paper, we present analytically derived filter function gradients and their numerical implementation. By doing so, we make the gradients easily accessible for various pulse optimization algorithms. Furthermore, we conduct an analysis of the computational complexity for obtaining the filter function derivatives by our implementation. We verify the theoretical prediction by comparing it to the actual run-time scaling behavior. Finally, we apply our filter function derivatives to gradient-based pulse optimization of generic two-qubit gates and contrast this approach with a gradient-free optimization method. While both strategies result in optimized pulses of similar fidelities, we show that the gradient-based optimization requires up to 2 orders of magnitude less time to converge than the gradient-free optimization for our test case.

In addition to pulse optimization for implementation of quantum gates, various other applications for analytic filter function derivatives exist. Since the filter function formalism can be used to describe quantum algorithms in terms of pulse sequences [20,21], filter function gradients can not only be used to optimize quantum gates, but also the implementation of quantum algorithms. Furthermore, pulse optimization can also aid noise spectroscopy [45], where carefully designed control pulses are used to obtain an accurate description of the present spectral noise density. As another application, filter function derivatives could be used to assess the impact of quasistatic noise or calibration errors on the high-frequency noise properties of a given pulse. Lastly, if the noise environment changes, e.g., due to dynamically changing operational parameters of on-chip control electronics, our gradients can be used to quickly recalibrate qubits in a quantum processor.

Based on our results, analytic filter function derivatives can facilitate quantum optimal control for various proposed qubit implementations that are subject to arbitrary classical auto- and cross-correlated noise. Thus, these derivatives can help to assess the potential performance of candidate hardware platforms for quantum information processing [46–48].

ACKNOWLEDGMENTS

This work is supported by the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation program (Grant Agreement No. 679342).

All correspondence should be addressed to Hendrik Bluhm.

APPENDIX A: FILTER FUNCTION FORMALISM FOR A GENERAL PULSE SEQUENCE

In Sec. II of the main text, the filter function formalism has been presented for the special case of a single control contribution and a single noise source. The following section recaptures some modifications needed for the calculation of filter function derivatives of a general pulse sequence consisting of various control and noise contributions.

Consider now the general description of a quantum system, whose total Hamiltonian $H(t)$ during $t \in [0, \tau]$ consists of two parts: a control Hamiltonian $H_c(t)$ consisting of a time-independent contribution H_0 and multiple adjustable parameters to achieve the intended quantum operation, and a noise Hamiltonian $H_n(t)$ perturbing $H_c(t)$,

$$H(t) = H_c(t) + H_n(t), \tag{A1a}$$

$$H_c(t) = H_0 + \sum_k u_k(t) A_k, \tag{A1b}$$

$$H_n(t) = \sum_\alpha s_\alpha(t) b_\alpha(t) B_\alpha. \tag{A1c}$$

Within $H_c(t)$, $u_k(t)$ is the adjustable control strength of the control operator A_k at time t. Likewise, in $H_n(t)$, $b_\alpha(t)$ is the randomly distributed amplitude of the Hermitian noise operator B_α and $s_\alpha(t)$ captures the system's sensitivity to the corresponding noise source and might be dependent on the control amplitude.

Following the approach given in Sec. II, the control matrix can be derived. The control propagator $U_c(t)$ again contains the control effects and fulfills the noise-free Schrödinger equation $i\partial U_c(t)/\partial t = H_c(t) U_c(t)$. In the main text, the set of Pauli operators $\{\mathbb{1}, \sigma_x, \sigma_y, \sigma_z\}$ was chosen as an operator basis. In general, the operators can be expressed in any orthonormal operator basis $\{C_0, C_1, \ldots, C_{d^2-1}\} \in \mathbb{C}^{d\times d}$ with respect to the Hilbert-Schmidt product $\langle C_i, C_j \rangle := \mathrm{tr}(C_i^\dagger C_j) = \mathbb{1}\delta_{ij}$. Taking this generalization into account, and considering the noise sensitivity, the control matrix in time domain $\tilde{\mathcal{B}}(t)$ needs to be modified. It is then defined as

$$\tilde{\mathcal{B}}(t)_{\alpha k} = s_\alpha(t)\,\mathrm{tr}[U_c^\dagger(t) B_\alpha U_c(t) C_k]. \tag{A2}$$

In the case of piecewise-constant control, a control sequence consists of $n_{\Delta t}$ constant single pulses. Let us denote $[t_{g-1}, t_g]$ as the time interval corresponding to the gth single pulse for $g \in \{1, \ldots, n_{\Delta t}\}$. The control sequence can then be described with the help of the single pulse propagators $U_c(t_g, t_{g-1})$. Consequently, the propagator cumulated up to the gth time step is given by $Q_g = \prod_{l=g}^{1} U_c(t_l, t_{l-1})$ and its so-called Liouville representation is defined by $\mathcal{Q}_{jk}{}^{(g)} = \mathrm{tr}(Q_g^\dagger C_j Q_g C_k)$. Next, we denote the duration of the gth single pulse with $\Delta t_g = t_g - t_{g-1}$, and the single pulse control matrix in frequency space at time step g with

$$\tilde{\mathcal{B}}^{(g)}(\omega) = \int_0^{\Delta t_g} dt e^{i\omega t} \tilde{\mathcal{B}}^{(g)}(t). \tag{A3}$$

The total control matrix of the pulse sequence can then be directly determined by

$$\tilde{\mathcal{B}}(\omega) = \sum_{g=1}^{n_{\Delta t}} e^{i\omega t_{g-1}} \tilde{\mathcal{B}}^{(g)}(\omega) \mathcal{Q}^{(g-1)}. \tag{A4}$$

In the above, $\mathcal{Q}^{(g-1)}$ is the control propagator of the individual single pulses cumulated up to time step $g-1$. The temporal positions of the single pulses enter the expression due to the Fourier transform via the phase factor $e^{i\omega t_{g-1}}$. Therefore, it is possible to obtain the total control matrix of a generic pulse sequence by summing up each single pulse control matrix multiplied with the cumulated control propagators of the priorly executed single pulses. Using Eq. (A4), the filter function F_α for a noise contribution α can be obtained by Eq. (9). If we extend the calculation to generic dimensions d, the infidelity $\mathcal{I}_\alpha$ for a noise contribution α is given by

$$\mathcal{I}_\alpha = \frac{1}{d} \int_{-\infty}^{\infty} \frac{d\omega}{2\pi} F_\alpha(\omega) S_\alpha(\omega). \tag{A5}$$

APPENDIX B: FILTER FUNCTION DERIVATIVES FOR A GENERAL PULSE SEQUENCE

When considering a pulse sequence with $n_{\Delta t} > 1$ time steps, the correlation terms in the total control matrix given by Eq. (20) need to be taken into account. In the following, we derive the filter function gradient for the general case of a pulse sequence under the assumption of piecewise-constant control.

We write $u_h(t_{g'})$ for the control amplitude in direction h at a fixed time $t_{g'}$ and note that $u_h(t) = u_h(t_g)$ for $t \in [t_{g-1}, t_g]$. Applying the product rule on Eq. (A4) results in

$$\frac{\partial \tilde{\mathcal{B}}_{\alpha k}(\omega)}{\partial u_h(t_{g'})} = \sum_{g=1}^{n_{\Delta t}} \sum_{j=1}^{d^2} e^{i\omega t_{g-1}} \times \left(\frac{\partial \tilde{\mathcal{B}}_{aj}^{(g)}(\omega)}{\partial u_h(t_{g'})} \mathcal{Q}_{jk}^{(g-1)} + \tilde{\mathcal{B}}_{\alpha j}{}^{(g)}(\omega) \frac{\partial \mathcal{Q}_{jk}^{g-1}}{\partial u_h(t_{g'})} \right). \tag{B1}$$

This expression depends on four quantities: the control propagators in Liouville representation $\mathcal{Q}_{jk}{}^{(g-1)}$, and their partial derivatives $\partial \mathcal{Q}_{jk}{}^{(g-1)}/\partial u_h(t_{g'})$; and the control matrices $\tilde{\mathcal{B}}_{\alpha j}{}^{(g)}(\omega)$, and their partial derivatives $\partial \tilde{\mathcal{B}}_{\alpha j}{}^{(g)}(\omega)/\partial u_h(t_{g'})$. To break the calculation into comprehensive parts, we dedicate each of the mentioned quantities one of the following subsections.

1. Control matrix at time step g

For a fixed time step g, the single control matrix $\tilde{\mathcal{B}}^{(g)}(\omega)$ is defined in Eq. (A3). In the following, we evaluate the integral formula analytically. To this end, we introduce the basis change matrix $V^{(g)}$ with its columns being the eigenvectors of the control Hamiltonian at time step g, $H_c{}^{(g)}$. An operator P in matrix representation transformed into the eigenbasis of $H_c{}^{(g)}$ is then denoted by $\bar{P}^{(g)} = V^{(g)\dagger} P V^{(g)}$.

Let $\{\omega_i{}^{(g)}\}_{1\le i\le d}$ be the set of eigenvalues of $H_c{}^{(g)}$ in the same order as the eigenvectors in $V^{(g)}$. The control propagator during time step g transformed into the eigenbasis of

$H_c{}^{(g)}$ is naturally a diagonal matrix

$$D_{ij}{}^{(g)}(t) \equiv [\bar{U}_c{}^{(g)}(t, t_{g-1})]_{ij} = \delta_{ij} \exp[-i\omega_i{}^{(g)}(t - t_{g-1})] \tag{B2}$$

with $t \in [t_{g-1}, t_g]$. Because of piecewise-constant control, the sensitivity during the gth time step is constant, i.e., $s_\alpha(t) = s_\alpha{}^{(g)}$. Inserting the identity $\mathbb{1} = V^{(g)\dagger}V^{(g)} = V^{(g)}V^{(g)\dagger}$ into Eq. (A2), and taking into account the cyclic properties of the trace results in

$$\begin{aligned}\tilde{\mathcal{B}}^{(g)}(t)_{\alpha j} &= s_\alpha{}^{(g)} \operatorname{tr}(\bar{U}_c{}^{(g)\dagger}\bar{B}_\alpha \bar{U}_c{}^{(g)}\bar{C}_j) \\ &= s_\alpha{}^{(g)} \operatorname{tr}\left(\sum_{ijkl} D_{ij}{}^{(g)\dagger}(t)\bar{B}_{\alpha,jk}D_{kl}{}^{(g)}(t)\bar{C}_{j,lm}\right) \\ &= s_\alpha{}^{(g)} \sum_{ik}\{\exp[i(\omega_i{}^{(g)} - \omega_k{}^{(g)})t]\bar{B}_{\alpha,ik}\bar{C}_{j,ki}\}.\end{aligned} \tag{B3}$$

Now we can carry out the transformation of $\tilde{\mathcal{B}}^{(g)}(t)$ into the frequency domain given by Eq. (A3). Since B_α and C_j are not time dependent, evaluating the time integral over time-dependent factors at time step g yields

$$\begin{aligned}\mathbb{O}_{ik}{}^{(g)}(\omega) &= \int_0^{\Delta t} dt \exp[i(\omega - \omega_i{}^{(g)} - \omega_k{}^{(g)})t] \\ &= \frac{\exp[i(\omega + \omega_i{}^{(g)} - \omega_k{}^{(g)})\Delta t_g] - 1}{i(\omega + \omega_i{}^{(g)} - \omega_k{}^{(g)})}.\end{aligned} \tag{B4}$$

By inserting this expression into Eq. (A3) and denoting elementwise multiplication by "$\circ$," the control matrix in frequency space can be evaluated to

$$\begin{aligned}\tilde{\mathcal{B}}_{\alpha j}{}^{(g)}(\omega) &= s_\alpha{}^{(g)} \sum_{ik} \bar{B}_{\alpha,ik}\bar{C}_{j,ki}\mathbb{O}_{ik}(\omega) \\ &= s_\alpha{}^{(g)} \operatorname{tr}\{[\bar{B}_\alpha \circ \mathbb{O}^{(g)}(\omega)]\bar{C}_j\}.\end{aligned} \tag{B5}$$

2. Derivative of the control propagator

In the main text we derived a closed formula of the control propagator gradient in the case of a single pulse. The resulting expression in Eq. (19) is valid for each control propagator gradient within a certain time step g. More specifically, the derivative of the control propagator $U_c(t, t_{g-1})$ within time step g is given as

$$\frac{\partial U_c(t, t_{g-1})}{\partial u_h(t_g)} = -iU_c(t, t_{g-1})V^{(g)}[\mathbb{M}^{(g)}(t) \circ \bar{A}_h]V^{(g)\dagger}. \tag{B6}$$

Differences are that, in the general case, various control contributions h are considered, and that $V^{(g)}$ and $\mathbb{M}^{(g)}(t)$ denote each quantity for the specific considered time step g.

3. Derivative of the propagator in Liouville representation

The main difference between the cases of a single pulse and a pulse sequence is that we need to take the derivative of the cumulated control propagator in Liouville representation $\partial \mathcal{Q}_{jk}{}^{(g-1)}/\partial u_h(t_{g'})$ at each time step into account. To this end, we keep in mind that $Q_g = \prod_{l=g}^{1} U_c(t_l, t_{l-1}) = U_c(t_g, 0)$ is the cumulative propagator up to time step g and that its Liouville representation is given elementwise as $\mathcal{Q}_{jk}{}^{(g)} = \operatorname{tr}(Q_g^\dagger C_j Q_g C_k)$. For the further calculation, we first need the derivative of Q_g, which can be evaluated as

$$\begin{aligned}\frac{\partial Q_g}{\partial u_h(t_{g'})} &= \frac{\partial U_c(t_{g-1}, 0)}{\partial u_h(t_{g'})} \\ &= \Theta(g-1, g')U_c(t_{g-1}, t_{g'}) \\ &\quad \times \frac{\partial U_c(t_{g'}, t_{g'-1})}{\partial u_h(t_{g'})}U_c(t_{g'-1}, 0),\end{aligned} \tag{B7a}$$

$$\Theta(g-1, g') = \begin{cases} 1, & g' < g-1, \\ 0, & \text{otherwise}, \end{cases} \tag{B7b}$$

where Eq. (B7b) incorporates the fact that cumulative propagators up to a time step g are independent of the control at a later point in time $g' > g$ and where $\partial U_c(t_{g'}, t_{g'-1})/\partial u_h(t_{g'})$ is given by Eq. (B6). By applying the product rule, we calculate the derivative of $\mathcal{Q}^{(g-1)}$ element-wise to

$$\begin{aligned}&\frac{\partial \mathcal{Q}_{jk}{}^{(g-1)}}{\partial u_h(t_{g'})} \\ &\quad = \Theta(g-1, g') \operatorname{tr}\left(\frac{\partial Q_{g-1}^\dagger}{\partial u_h(t_{g'})} C_j Q_{g-1} C_k \right. \\ &\qquad \left. + Q_{g-1}^\dagger C_j \frac{\partial Q_{g-1}}{\partial u_h(t_{g'})} C_k\right).\end{aligned} \tag{B8}$$

4. Derivative of the control matrix

The last quantity that remains to be computed are the derivatives of single control matrices. In order to calculate the derivative of the control matrix at time step g in the frequency domain $\partial \tilde{\mathcal{B}}^{(g)}(\omega)/\partial u_h(t_{g'})$, we first obtain a formula for the gradient in time space and subsequently transform the result into frequency space. To this end, we consider $t \in [t_{g-1}, t_g]$ and again label $\Delta t = t - t_{g-1}$. If $s_\alpha(t)$ depends on the control amplitudes, an additive term including the noise sensitivity derivative needs to be taken into account. Here, we assume to know the analytic dependency of $s_\alpha(t)$ on the control amplitudes and, therefore, concentrate on the remaining term by choosing $s_\alpha(t)$ to be independent of the control amplitudes. Using the product rule and cyclic properties of the trace, and keeping in mind that the noise and control operators are independent of the

control strength, leads to

$$\frac{\partial \tilde{\mathcal{B}}_{aj}{}^{(g)}(\Delta t)}{\partial u_h(t_{g'})} = i\delta_{gg'} s_\alpha{}^{(g)} \operatorname{tr}\big(U_c^\dagger(t,t_{g'-1}) B_\alpha U_c(t,t_{g'-1})[C_j, V^{(g)} \times [\mathbb{M}^{(g)}(t)\circ \bar{A}_h]V^{(g)\dagger}]\big), \tag{B9}$$

where we write $[X, Y]$ for the commutator of X and Y.

Similarly to the approach in Sec. 1, we transform the control propagator into the eigenspace of $H_c{}^{(g)}$ and make use of its diagonal form given in Eq. (B2). Under consideration of the cyclic properties of a trace, and writing $\Delta\omega_{nm} = \omega_n{}^{(g)} - \omega_m{}^{(g)}$, the calculation can be further carried out as

$$\frac{\partial \tilde{\mathcal{B}}_{\alpha j}{}^{(g)}(\Delta t)}{\partial u_h(t_{g'})} = i\delta_{gg'} s_\alpha{}^{(g)} \sum_{mn} e^{i\Delta\omega_{nm}\Delta t} \times \bar{B}_{\alpha,nm}[\bar{C}_j, \mathbb{M}^{(g)}(t)\circ \bar{A}_h]_{mn}. \tag{B10}$$

Transforming the latter into the frequency domain requires an integration over time. Since $s_\alpha{}^{(g)}$ and $\mathcal{B}_\alpha$ are time independent within the regarded time step, this integration lies in

$$\begin{aligned}\mathbb{K}_{pq}{}^{(mn)} &= \int_0^{\Delta t_g} dt \exp[i(\omega+\Delta\omega_{nm})t][\mathbb{M}^{(g)}(t)]_{pq} \\ &= \delta_{pq}\left(\frac{\Delta t_g \exp[i(\omega+\Delta\omega_{nm})\Delta t_g]}{i(\omega+\Delta\omega_{nm})} + \frac{\exp[i(\omega+\Delta\omega_{nm})\Delta t_g]-1}{(\omega+\Delta\omega_{nm})^2}\right) \\ &\quad + \frac{1-\delta_{pq}}{i\Delta\omega_{pq}} \times \left(\frac{\exp[i(\omega+\Delta\omega_{nm}+\Delta\omega_{pq})\Delta t_g]-1}{i(\omega+\Delta\omega_{nm}+\Delta\omega_{pq})} - \frac{\exp[i(\omega+\Delta\omega_{nm})\Delta t_g]-1}{i(\omega+\Delta\omega_{nm})}\right)\end{aligned} \tag{B11a}$$

if $\omega_p{}^{(g)} \neq \omega_q{}^{(g)}$ and

$$\mathbb{K}_{pq}{}^{(mn)} = \frac{\Delta t_g \exp[i(\omega+\Delta\omega_{nm})\Delta t_g]}{i(\omega+\Delta\omega_{nm})} + \frac{\exp[i(\omega+\Delta\omega_{nm})\Delta t_g]-1}{(\omega+\Delta\omega_{nm})^2} \tag{B11b}$$

otherwise. Using this result, we can first evaluate the transformed commutator elementwise to

$$\mathbb{N}_{mn} = [\bar{C}_j, \mathbb{K}^{(mn)}\circ \bar{A}_h]_{mn}, \tag{B12}$$

and, consequently, derive a concise formula for the gradient of the control matrix at time step g in the frequency domain as

$$\frac{\partial \tilde{\mathcal{B}}_{\alpha j}{}^{(g)}(\omega)}{\partial u_h(t_{g'})} = i\delta_{gg'} s_\alpha{}^{(g)} \operatorname{tr}(\bar{B}_\alpha \mathbb{N}). \tag{B13}$$

Inserting Eqs. (B5), (B8), and (B13) into Eq. (B1) gives the analytic derivative of the total control matrix. The latter result can then be used within Eq. (10) to calculate analytic filter function derivatives for a general pulse sequence.

If the noise sensitivity $s(t)$ also depends on the control amplitude, the derivative of the control matrix in Eq. (B13) by the product rule has the additional summand

$$\frac{\partial s_\alpha{}^{(g)}}{\partial u_h(t_{g'})} \operatorname{tr}\{[\bar{B}_\alpha \circ \mathbb{O}^{(g)}(\omega)]\bar{C}_j\}. \tag{B14}$$

APPENDIX C: SUPPLEMENT INFORMATION ON SOFTWARE IMPLEMENTATION AND COMPUTATIONAL COMPLEXITY

1. Structure of implementation

In order to extend the `filter_functions` software package by filter function derivatives, the module `gradient` is implemented. Additionally, a method for directly obtaining filter function derivatives from a given pulse sequence is added to the `PulseSequence` class. The overall structure is illustrated in Fig. 3.

2. Verification of linear run-time dependency

Within the run-time analysis, the theoretically predicted linear dependency on the number of frequency samples, control operators, and noise operators could clearly be verified by means of asymptotic plots. Figure 4 displays the analysis' results.

APPENDIX D: AUTOMATIC DIFFERENTIATION WITH JAX

The JAX package [40] was developed by Google for automatic differentiation and high-performance compilation

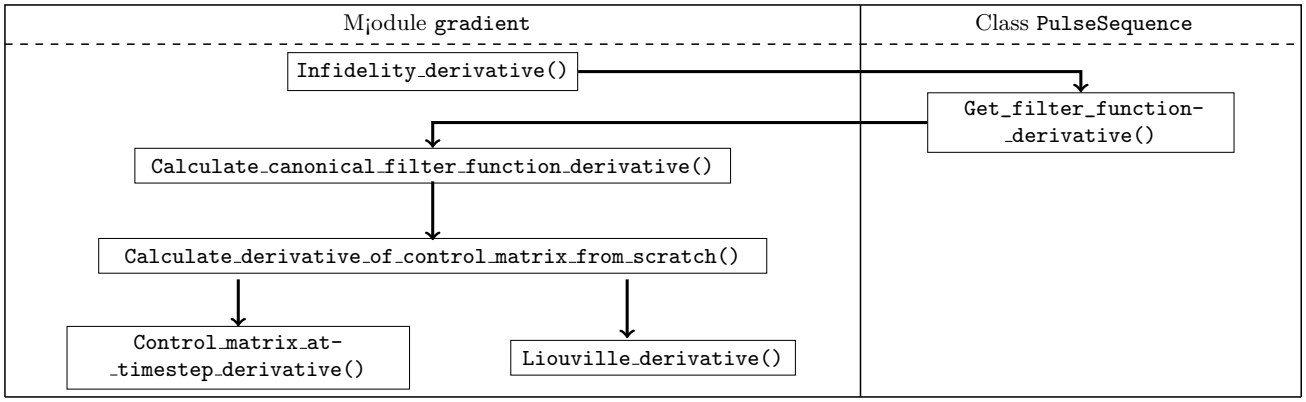

FIG. 3. Structure of the implemented module `gradient` and newly added method in the `PulseSequence` class. The chart shows which functions are called within each function. An arrow pointing from function A to function B indicates that A calls B.

of PYTHON code. It can for example be used in the computationally intensive training of neural networks, where gradients need to be calculated frequently.

For comparison, we calculate the gradient of the entanglement fidelity with automatic differentiation for random sequences of $n_{\Delta t}$ time steps. The averaged run time is plotted in Fig. 5 and fitted to the function $f\,(n_{\Delta t}) = a_0 \times n_{\Delta t}^{a_1}$ with the fit parameters a_0 and a_1. The resulting $a_1 = 2.21$ demonstrates that the automatic differentiation scales like the analytic calculation (cf. Table I).

Within the considered ranges of $n_{\Delta t}$, the calculation based on JAX outperforms the use of analytic gradients, but since both methods show the same scaling behavior, it is not clear whether the automatic differentiation is more efficient or if the run-time difference is achieved by the high-performance compilation integrated in JAX. For a fair comparison, the analytic gradients would need to be implemented using the same compilation methods.

The use of automatic differentiation relieves the user of the effort to derive and implement analytical derivatives, at the cost of implementing the calculation compatible to an automatic differentiation package like JAX. This includes the restriction to a less user-friendly and flexible procedural implementation. In the case of JAX, one is also limited to UNIX-like platforms at the time of writing.

The analytical derivatives also allow the user to easily access intermediary results and to reuse the derivatives of the filter functions easily to calculate derivatives of multiple quantities that are derived from the filter functions,

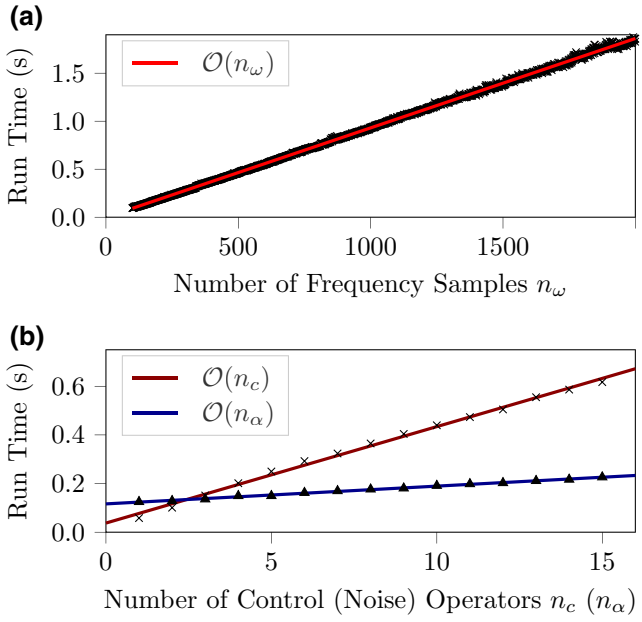

FIG. 4. Linear run-time behavior in n_ω, n_c, and n_α of computing the infidelity gradient for random pulse sequences of dimension $d = 4$ and $n_{\Delta t} = 3$ time steps. Panel (a) shows the run-time plot for various n_ω. To this end, a random pulse with $n_c = n_\alpha = 2$ control and noise contributions, and $n_d = 1$ drift contribution is generated. Panel (b) shows the run-time plot for various n_c (n_α). The plot shows the median data of 100 randomly generated pulse sequences with $n_\omega = 200$, and n_α (n_c) $= 2$ depending on which parameter is fixed.

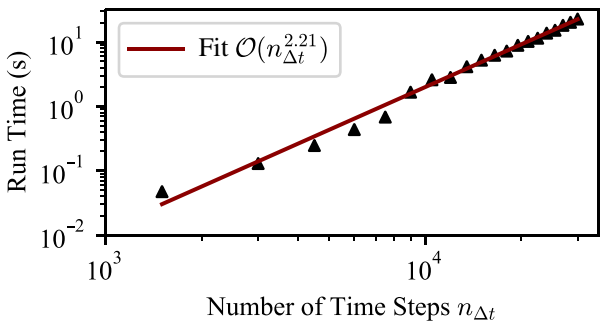

FIG. 5. Run-time behavior of the fidelity gradient calculation using automatic differentiation implemented with JAX [40]. The run time shows the same quadratic scaling in $n_{\Delta t}$ as the calculations of analytic gradients; cf. Fig. 1.

such as gate fidelities and leakage rates, or to expand the gradient calculation to concatenated filter functions [21].

APPENDIX E: SUPPLEMENT INFORMATION ON THE COMPARISON TO THE NELDER-MEAD METHOD

In the following we elaborate more on the details of the optimizations used to compare the use of analytical gradients to a gradient-free method. For better comparability, both optimization algorithms are started from the same initial pulses.

In Fig. 6 we plot the distribution of the optimized pulses' infidelities found in the optimization corresponding to Fig. 2. We can see that both algorithms find similar minimal infidelities over 100 runs. From the fact that the distribution of final cost values is wider for the L-BFGS-B algorithm, we can deduct that the gradient-based method is more prone to get stuck in local minima in the optimization space, while the Nelder-Mead algorithm appears to provide better global convergence.

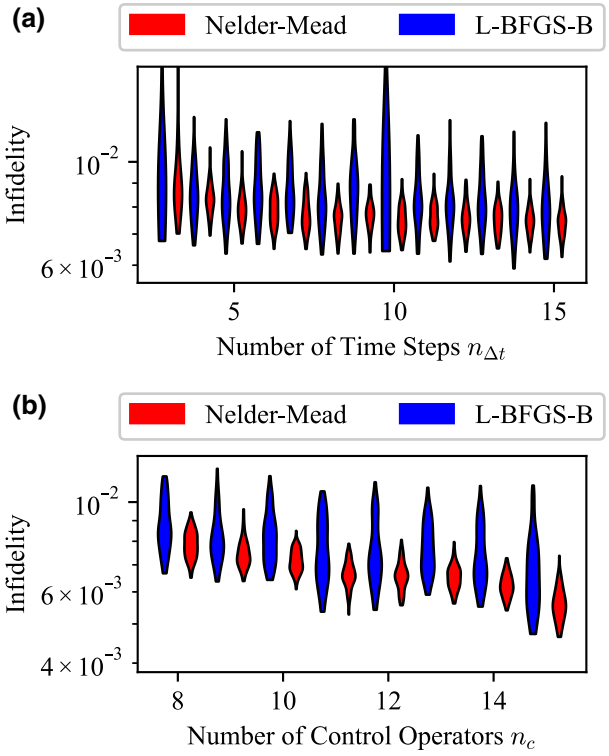

FIG. 6. Violin plot of the distribution of final total infidelities $\mathcal{I}$ of the optimizations (see Fig. 2). Each pair of a blue (L-BFGS) and a red (Nelder-Mead) violin shows the distributions of final infidelities for one set of 100 optimization runs, belonging to the number around which they are slightly shifted. (a) The distribution of final infidelities is plotted as a function of the number of control operators $n_{\Delta t}$, while the number of time steps is kept constant at $n_c = 8$. (b) Here, as a function of the number of control operators n_c, the number of time steps is constant at $n_{\Delta t} = 6$.

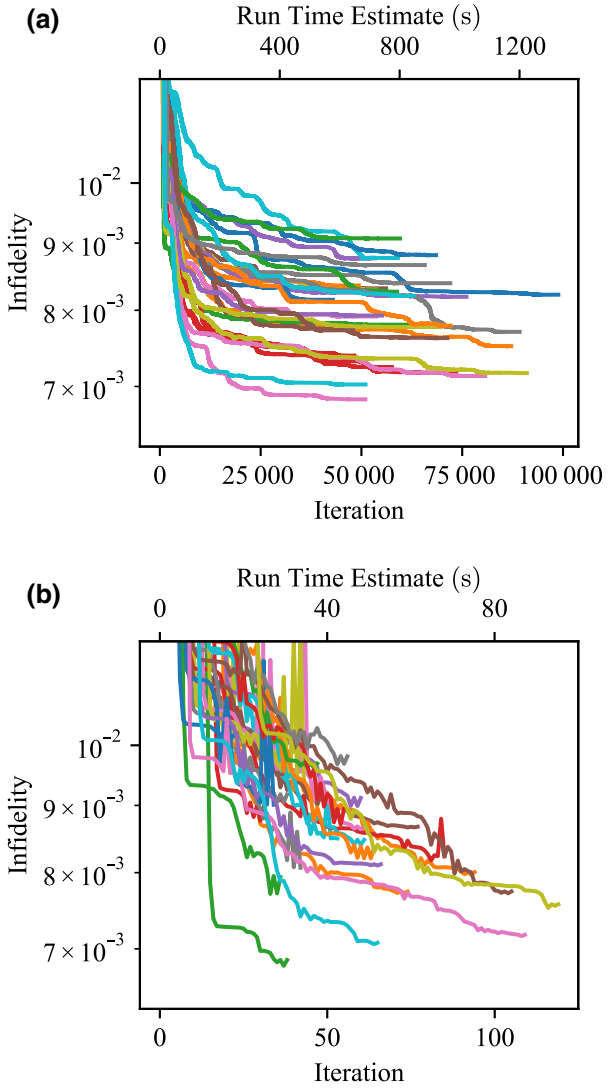

FIG. 7. Convergence behavior of the (a) Nelder-Mead and (b) L-BFGS-B algorithms. Each plot shows the infidelity during 30 optimization runs as a function of the optimization algorithm's iteration. The runs are taken from the optimization with $n_{\Delta t} = 6$ and $n_c = 8$. We calculate the run-time estimates by multiplying the iteration scale with the average duration of an iteration.

In the convergence plots in Fig. 7, it can be seen that the gradient-based method requires far less iterations than the Nealder-Mead algorithm. The plateaus in the plots, where the algorithm reduces the Infidelity only marginally over several iterations, can be interpreted as features in the optimization landscape with the approximate form of local minima. We also observe that the termination condition favors neither algorithm as both do not spend an excessive amount of iterations on minor improvements.

To compare the convergence of the algorithms with respect to the initial parameters, we plot in Fig. 8 the expectation value of the infidelity as a function of the number of optimization runs n_R. This quantifies the infidelity of

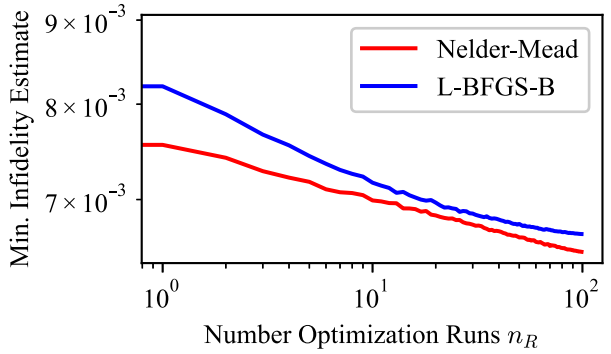

FIG. 8. Estimation of the minimal infidelity achieved in n_R optimization runs with different initial conditions. The estimate is calculated by averaging over the minimum of n_R randomly drawn samples from the distribution plotted in Fig. 6 at $n_c = 8$.

the best pulse found by starting the optimization n_R times with different initial pulse values. To find a pulse of similar infidelity on average, the L-BFGS-B algorithm needs to be restarted roughly twice as many times as the Nelder-Mead algorithm.

For the numerical pulse optimization, we used the open-source PYTHON package `qopt` [35,36].

[1] M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information* (Cambridge University Press, New York, 2010).

[2] A. W. Harrow and A. Montanaro, Quantum computational supremacy, Nature **549**, 203 (2017).

[3] W. G. Unruh, Maintaining coherence in quantum computers, Phys. Rev. A **51**, 992 (1995).

[4] J. Preskill, Quantum computing in the NISQ era and beyond, Quantum **2**, 79 (2018).

[5] S. J. Glaser, U. Boscain, T. Calarco, C. P. Koch, W. Köckenberger, R. Kosloff, I. Kuprov, B. Luy, S. Schirmer, T. Schulte-Herbrüggen, D. Sugny, and F. K. Wilhelm, Training Schrödinger's cat: Quantum optimal control: Strategic report on current status, visions and goals for research in Europe, Eur. Phys. J. D **69**, 279 (2015).

[6] A. P. Peirce, M. A. Dahleh, and H. Rabitz, Optimal control of quantum-mechanical systems: Existence, numerical approximation, and applications, Phys. Rev. A **37**, 4950 (1988).

[7] H. Rabitz, Focus on quantum control, New J. Phys. **11**, 105030 (2009).

[8] X. Wang, L. S. Bishop, J. P. Kestner, E. Barnes, K. Sun, and D. Sarma, Composite pulses for robust universal control of singlet-triplet qubits, Nat. Commun. **3**, 997 (2012).

[9] X. Wang, L. S. Bishop, E. Barnes, J. P. Kestner, and S. D. Sarma, Robust quantum gates for singlet-triplet spin qubits using composite pulses, Phys. Rev. A **89**, 022310 (2014).

[10] X. Wang, F. A. Calderon-Vargas, M. S. Rana, J. P. Kestner, E. Barnes, and S. Das Sarma, Noise-compensating pulses for electrostatically controlled silicon spin qubits, Phys. Rev. B **90**, 155306 (2014).

[11] X. C. Yang and X. Wang, Noise filtering of composite pulses for singlet-triplet qubits, Sci. Rep. **6**, 28996 (2016).

[12] F. C. Wellstood, C. Urbina, and J. Clarke, Low-frequency noise in dc superconducting quantum interference devices below 1 k, Appl. Phys. Lett. **50**, 772 (1987).

[13] J. Bylander, S. Gustavsson, F. Yan, F. Yoshihara, K. Harrabi, G. Fitch, D. G. Cory, Y. Nakamura, J. S. Tsai, and W. D. Oliver, Noise spectroscopy through dynamical decoupling with a superconducting flux qubit, Nat. Phys. **7**, 565 (2011).

[14] D. Drung, J. Beyer, J. Storm, M. Peters, and T. Schurig, Investigation of low-frequency excess flux noise in dc squids at mk temperatures, IEEE Trans. Appl. Supercond. **21**, 340 (2011).

[15] S. M. Anton, C. Müller, J. S. Birenbaum, S. R. O'Kelley, A. D. Fefferman, D. S. Golubev, G. C. Hilton, H.-M. Cho, K. D. Irwin, F. C. Wellstood, G. Schön, A. Shnirman, and J. Clarke, Pure dephasing in flux qubits due to flux noise with spectral density scaling as $1/f^{\alpha}$, Phys. Rev. B **85**, 224505 (2012).

[16] A. V. Kuhlmann, J. Houel, A. Ludwig, L. Greuter, D. Reuter, A. D. Wieck, M. Poggio, and R. J. Warburton, Charge noise and spin noise in a semiconductor quantum device, Nat. Phys. **9**, 570 (2013).

[17] J. Yoneda, K. Takeda, T. Otsuka, T. Nakajima, M. R. Delbecq, G. Allison, T. Honda, T. Kodera, S. Oda, Y. Hoshi, N. Usami, K. M. Itoh, and S. Tarucha, A quantum-dot spin qubit with coherence limited by charge noise and fidelity higher than 99.9%, Nat. Nanotechnol. **13**, 102 (2018).

[18] T. Struck, A. Hollmann, F. Schauer, O. Fedorets, A. Schmidbauer, K. Sawano, H. Riemann, N. V. Abrosimov, Ł. Cywiński, D. Bougeard, and L. R. Schreiber, Low-frequency spin qubit energy splitting noise in highly purified 28Si/SiGe, npj Quantum Inf. **6**, 40 (2020).

[19] T. J. Green, J. Sastrawan, H. Uys, and M. J. Biercuk, Arbitrary quantum control of qubits in the presence of universal noise, New J. Phys. **15**, 095004 (2013).

[20] P. Cerfontaine, T. Hangleiter, and H. Bluhm, Filter Functions for Quantum Processes under Correlated Noise, Phys. Rev. Lett. **127**, 170403 (2021).

[21] T. Hangleiter, P. Cerfontaine, and H. Bluhm, Filter-function formalism and software package to compute quantum processes of gate sequences for classical non-markovian noise, Phys. Rev. Res. **3**, 043047 (2021).

[22] A. Soare, H. Ball, D. Hayes, J. Sastrawan, M. C. Jarratt, J. J. Mcloughlin, X. Zhen, T. J. Green, and M. J. Biercuk, Experimental noise filtering by quantum control, Nat. Phys. **10**, 825 (2014).

[23] M. A. Nielsen, A simple formula for the average gate fidelity of a quantum dynamical operation, Phys. Lett. A **303**, 249 (2002).

[24] T. Green, H. Uys, and M. J. Biercuk, High-Order Noise Filtering in Nontrivial Quantum Logic Gates, Phys. Rev. Lett. **109**, 020501 (2012).

[25] C.-H. Huang and H.-S. Goan, Robust quantum gates for stochastic time-varying noise, Phys. Rev. A **95**, 062325 (2017).

[26] C.-H. Huang, C.-H. Yang, C.-C. Chen, A. S. Dzurak, and H.-S. Goan, High-fidelity and robust two-qubit gates for quantum-dot spin qubits in silicon, Phys. Rev. A **99**, 042310 (2019).

[27] I. Kuprov and C. T. Rodgers, Derivatives of spin dynamics simulations, J. Chem. Phys. **131**, 234108 (2009).
[28] H. Ball, M. J. Biercuk, A. R. R. Carvalho, J. Chen, M. Hush, L. A. D. 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 Sci. Technol. **6**, 044011 (2021).
[29] P. Cerfontaine, T. Botzem, D. P. DiVincenzo, and H. Bluhm, High-Fidelity Single-Qubit Gates for Two-Electron Spin Qubits in GaAs, Phys. Rev. Lett. **113**, 150501 (2014).
[30] U. Haeberlen and J. S. Waugh, Coherent averaging effects in magnetic resonance, Phys. Rev. **175**, 453 (1968).
[31] S. Blanes, F. Casas, J. A. Oteo, and J. Ros, The magnus expansion and some of its applications, Phys. Rep. **470**, 151 (2009).
[32] W. Magnus, On the exponential solution of differential equations for a linear operator, Commun. Pure Appl. Math. **7**, 649 (1954).
[33] P. Szańkowski, G. Ramon, J. Krzywda, D. Kwiatkowski, and Ł. Cywiński, Environmental noise spectroscopy with qubits subjected to dynamical decoupling, J. Phys. Condens. Matter **29**, 333001 (2017).
[34] T. Hangleiter, I. N. M. Le, and J. D. Teske, filter_functions: A package for efficient numerical calculation of generalized filter functions to describe the effect of noise on quantum gate operations (version v1.0.0), (2021),.
[35] J. Teske, qopt, A simulation and quantum optimal control package, (2021), https://github.com/qutech/qopt.
[36] J. D. Teske, P. Cerfontaine, H. Bluhm, qopt, An experiment-oriented qubit simulation and quantum optimal control package (2021), ArXiv:2110.05873.
[37] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms* (The MIT Press, Cambridge, Massachusetts, 2009), 3rd ed.
[38] D. Coppersmith and S. Winograd, Matrix multiplication via arithmetic progressions, J. Symb. Comput. **9**, 251 (1990).
[39] We used a Intel®Core^{TM}i5-2400 processor with four logical cores.
[40] J. Bradbury, R. Frostig, P. Hawkins, M. J. Johnson, C. Leary, D. Maclaurin, G. Necula, A. Paszke, J. VanderPlas, S. Wanderman-Milne, and Q. Zhang, JAX: composable transformations of Python+NumPy programs (2018).
[41] C. Zhu, R. H. Byrd, P. Lu, and J. Nocedal, Algorithm 778: L-bfgs-b: Fortran subroutines for large-scale bound-constrained optimization, ACM Trans. Math. Softw. **23**, 550 (1997).
[42] J. L. Morales and J. Nocedal, Remark on "algorithm 778: L-bfgs-b: Fortran subroutines for large-scale bound constrained optimization", ACM Trans. Math. Softw. **38**, 7 (2011).
[43] J. A. Nelder and R. Mead, A simplex method for function minimization, Comput. J. **7**, 308 (1965).
[44] A. Blaessle, constrnmpy: Constrained nealder mead implemented in python.
[45] N. D. Pozza, S. Gherardini, M. M. Müller, and F. Caruso, Role of the filter functions in noise spectroscopy, Int. J. Quantum Inf. **17**, 1941008 (2019).
[46] J. Kelly, *et al.*, Optimal Quantum Control Using Randomized Benchmarking, Phys. Rev. Lett. **112**, 240504 (2014).
[47] P. Cerfontaine, T. Botzem, J. Ritzmann, S. S. Humpohl, A. Ludwig, D. Schuh, D. Bougeard, A. D. Wieck, and H. Bluhm, Closed-loop control of a GaAs-based singlet-triplet spin qubit with 99.5% gate fidelity and low leakage, Nat. Commun. **11**, 4144 (2020).
[48] M. J. Biercuk, H. Uys, A. P. VanDevender, N. Shiga, W. M. Itano, and J. J. Bollinger, High-fidelity quantum control using ion crystals in a penning trap (2009), ArXiv:0906.0398.