Advantages of mixed unitary operators for quantum information processing

Anthony M. Polloreno*
Rigetti Computing, Berkeley, CA

Kevin C. Young
Sandia National Laboratories, Livermore, CA

(Dated: February 22, 2019)

Coherent errors in quantum operations are ubiquitous. Whether arising from spurious environmental couplings or errors in control fields, such errors can accumulate rapidly and degrade the performance of a quantum circuit significantly more than an average gate fidelity may indicate. As Hastings and Campbell have recently shown, randomly sampling an ensemble of implementations of a target gate yields an effective quantum channel that well-approximates the target, but with dramatically suppressed coherent error. Our results extend those of Hastings and Campbell to include robustness to drifting external control parameters. We implement these constructions using a superconducting qubit and will discuss randomized benchmarking results consistent with a marked reduction in coherent error.

I. INTRODUCTION

The past decade has seen a dramatic increase in the performance and scale of quantum information processors (QIPs). Gate fidelities are now routinely in the 99% to 99.99% range [1, 2], and dozens of individually-addressable qubits are becoming available on integrated devices. While these advances are promising steps forward on the path towards a computationally useful QIP, the quantum supremacy [3] milestone has yet to be definitively reached. The limiting factor, of course, is errors in the quantum gate operations.

The impact of an error in a quantum gate depends strongly on both the magnitude and the nature of the errors. Systematic, or *coherent*, errors can arise from poorly calibrated controls or imperfect gate compilations that induce repeatable, undesired unitary errors on the state of a QIP. Errors of this type are correlated in time may add up constructively or destructively, depending on the circuit. They are are computationally expensive to model and it can be difficult to place tight analytic bounds on circuit performance. Contrast this against random, or *stochastic*, errors, which result from high-frequency noise in the controls or the environment. Systems with stochastic errors can be modeled by defining a rate of various discrete errors in the system, such as a bit flips or phase flips. These errors are significantly easier to simulate on a classical computer, and their impact on quantum circuits is much easier to estimate [Ken Brown, Steve Flammia].

Quantitative bounds on the performance of quantum circuits can be constructed using the diamond distance []. [Discuss subadditivity of diamond norm here... The resulting effective quantum process has a diamond distance that grows only quadratically in the over/under rotation angle of the component gates. While these errors can often be identified and reconstructed using various tomographic techniques, their impact on a given quantum difficult to predict. The diamond distance can be used to bound the total variation distance (TVD) of a quantum circuit, but it is in general sensitive at first order to repeated application of a gate with coherent errors. For long circuits, this can add up extremely quickly.]. Despite the relative ease of modeling stochastic errors, coherent errors are often much more likely to appear in QIPs. Recent work by Campbell and Hastings[4–6], however, has shown that coherent noise can be strongly suppressed by probabilistically mixing several distinct implementations of the target quantum gates.

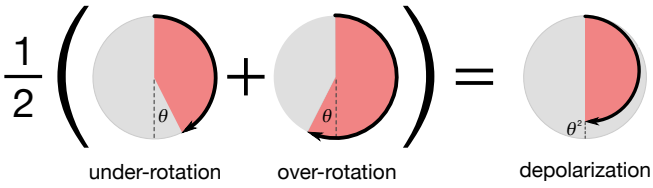

FIG. 1. An example of a mixed unitary process. Using optimal control, two implementations of a Z_π gate are designed to have equal and opposite sensitivity to errors (if one implementation over-rotates by angle θ, then the other *under*-rotates by θ). Each time the gate is used, one of these implementations is chosen at random. The resulting quantum channel is equivalent to a perfect implementation of the gate followed by dephasing of $\mathcal{O}\left(\theta^2\right)$.

In this article we discuss various applications of these mixed unitary controls, and show that the advantages of this approach can be made robust to drift in the gate implementations. We demonstrate that, depending on the objective, different numerical optimizations may be preferred. We present an experimental implementation of single-qubit mixed unitary controls on a superconducting qubit testbed at Rigetti Computing. Using randomized benchmarking, we are able to show a marked improvement in error rates, as well as a reduced variance

* Email: anthony@rigetti.com

in circuit outcome probabilities, indicating a reduction in the coherence of the error. We further provide an optimal control approach to the mixed unitary control design problem, and apply our methods in simulation where we construct single- and two-qubit mixed unitary controls which are robust to drift and uncertainty in the control parameters.

II. MATHEMATICAL PRELIMINARIES

Quantum gate operations are implemented by applying a sequence of classical control fields to some set of qubits. Fluctuations in the environment or imperfections in the controls can cause the state of the qubits to change in a way that is different from what was intended. But if the gates are fairly stable with time and context[7], then we can usually describe their action on the qubit state using *process matrices* – linear, Markovian maps on the state of some qubits. When working with process matrices, it is convenient to write the system density operator using a vectorized representation, and in this article, we'll make use of the generalized Bloch vector,

$$\vec{\rho} = \mathrm{Tr}\,(\rho\vec{\Sigma})/2^n, \tag{1}$$

where $\vec{\Sigma}$ is a vector of all 4^n n-qubit Pauli operators. For a single qubit, this is $\vec{\Sigma} = \{I, \sigma_x, \sigma_y, \sigma_z\}$. The action of a gate is then given by the usual matrix multiplication:

$$\vec{\rho} \to \mathcal{G}\vec{\rho} = \mathcal{E}\tilde{\mathcal{G}}\vec{\rho}. \tag{2}$$

Here $\mathcal{G}$ is the target operation, $\tilde{\mathcal{G}}$ is the actual gate as implemented, and $\mathcal{E}$ is the effective error channel:

$$\mathcal{E} = \left(\begin{array}{c|c} 1 & \vec{0}^T \\ \hline \vec{m} & R \end{array}\right) \tag{3}$$

The top row of all trace-preserving (TP) maps is fixed to $\{1, 0, 0, 0, \cdots\}$. The rest of the first column, $\vec{m}$, describes any deviations from unitality, as could arise from amplitude damping. If the error channel is unitary, then the error is coherent, and the submatrix R is perfectly antisymmetric, corresponding to a rotation of the generalized Bloch vector. If R is diagonal, then the error channel is Pauli stochastic, with each entry corresponding to the probability that the associated Pauli error occurs in each application of the gate. Any such Pauli error is expressible as a unitary matrix U that acts on the density matrix by conjugation, and has an action on the vectorized density matrix given by:

$$\vec{\rho} \to \rho_i \vec{\Sigma}_{ijk} U^\dagger_{kl} \vec{\Sigma}_{lmn} U_{mj} \tag{4}$$

If R is symmetric but not diagonal, then the channel is still stochastic, but the random errors consist of correlated Pauli operators (such as $X + Y$). For a single qubit, this describes everything, but the situation can be slightly more complicated for more qubits.

[this should go somewhere:] There are often many possible ways of implementing any given target quantum gate. Campbell and Hastings, for instance, consider gates compiled using the Solovey-Kitaev algorithm, for which many approximate gate compilations are possible.[4, 5]

In addition to the type of errors, we care about the size of an error. The *size* of an error in quantum gates may be quantified in a number of ways. Two of the most common metrics are the average gate fidelity, $\mathcal{F}$, and the diamond norm, $||\cdot||_\diamond$. These may be represented in terms of the error maps as

$$||I - \mathcal{E}||_\diamond = \sup_\rho ||(I \otimes I)(\rho) - (\mathcal{E} \otimes I)(\rho)||_1 \tag{5}$$

$$\mathcal{F}(\mathcal{E}) = \frac{\mathrm{Tr}\,\mathcal{E} + d}{d^2 + d} \tag{6}$$

Which metric is relevant depends on the application, and can yield very different numbers. For instance, the diamond norm is generally linear in the over-rotation angle of a quantum operation, while the average gate infidelity (AGI), given by 1-$\mathcal{F}$, is generally quadratic in the over-rotation angle of a quantum operation. These two metrics also vary significantly in performance on mixed processes.

[fix this] A *mixed unitary process* (MUP) consists of a set of unitary channels, $\tilde{\mathcal{G}}_j$, and associated weights, $\sum_j \omega_j = 1$. The process matrix for a mixed unitary channel is then the weighted sum of the component channels, $\tilde{\mathcal{G}}_\mathrm{M} = \sum_j \omega_j \tilde{\mathcal{G}}_j$, and the associated error channel is simply the weighted sum of the associated error channels, $\mathcal{E}_\mathrm{M} = \sum_j \omega_j \mathcal{E}_j$. From this definition, we can use linearity to compute the AGI of a MUP. We see that the AGI of any MUP will be the convex sum of the consituent fidelities, with the same weighting. The diamond norm, however, is non-linear function of the channel, and can in general be smaller for a MUP than any of the processes being mixed.

Campbell[4] considered the important problem of minimizing the diamond norm of the resulting error channel. Given a collection of component channels with error at most ϵ, he showed that if the Hamiltonians form a convex set containing the origin, then the diamond norm can be quadratically supressed. The diamond norm is a particularly appealing target because the it provides useful error bounds on quantum circuits. However, it is not the only optimization target that can be chosen.

A. A simple example

As a simple example, consider a scenario in which we have a single-qubit and four possible implementations of a π-pulse about the σ_x axis. The error channels for these four implementations are themselves unitary rotations about the σ_x axis with rotation angles of $\{-2\epsilon, -\epsilon, \epsilon, 2\epsilon\}$. Such a situation could appear, for instance, if there were

amplitude errors on the fields used to affect the gates, and if the control could be implemented by a rotation about the positive or negative σ_x axis.

In such a scenario, if the ultimate goal is to produce a channel whose effect can be Monte Carlo simulated, then a useful approach would be to construct a channel whose errors are Pauli stochastic. Such a channel could be constructed, for instance, by drawing from this collection uniformly at random. More generally, given a collection of control, such a channel could be produced by minimizing the off-diagonal elements of R in Equation 3. We implemented such a routine on a superconducting qubit.

The error operator corresponding to a rotation error by angle ϵ is then:

$$\mathcal{E}(\epsilon) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\epsilon & \sin\epsilon & 0 \\ 0 & -\sin\epsilon & \cos\epsilon & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{7}$$

[Don't be dumb with right hand rules] The associated diamond distance of this channel is $|\mathcal{I} - \mathcal{E}(\epsilon)|_\diamond \simeq \epsilon$, while the fidelity of the channel is $\mathcal{F}(\mathcal{E}(\epsilon)) = \epsilon^2$. If we were construct two channels, one with error $\mathcal{E}(\epsilon)$ and one with error $\mathcal{E}(-\epsilon)$, then we could construct a new channel with error $\mathcal{E}_{\text{eff}} = \frac{1}{2}(\mathcal{E}(\epsilon) + \mathcal{E}(\epsilon))$. The effective channel then has a matrix representation:

$$\mathcal{E}(\epsilon) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\epsilon & 0 & 0 \\ 0 & 0 & \cos\epsilon & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \simeq \mathcal{I} - \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & \epsilon^2/2 & 0 & 0 \\ 0 & 0 & \epsilon^2/2 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \tag{8}$$

And with same average process fidelity, ϵ^2, but now a suppressed diamond norm ϵ^2. Now write this in terms of the error generators $\mathcal{E}(\epsilon) = \exp(\epsilon G)$ where

$$G = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \tag{9}$$

The effective generator vanishes at first order.

$$\mathcal{E}(\epsilon) = \mathcal{I} + \epsilon G + \frac{1}{2}\epsilon^2 G^2 + \mathcal{O}(\epsilon^3) \tag{10}$$

$$\mathcal{E}_{\text{eff}} = \mathcal{I} + \sum_i \omega_i \epsilon_i G_i + \sum_i \omega_i \frac{1}{2}\epsilon_i^2 G_i^2 + + \mathcal{O}(\epsilon^3) \tag{11}$$

$$\mathcal{E}_{\text{eff}} = \mathcal{I} + \sum_i \omega_i \frac{1}{2}\epsilon_i^2 G_i^2 + \mathcal{O}(\epsilon^3) \tag{12}$$

Gate	H_{eff}	AGI	$\|\|\cdot\|\|_\diamond$
$U_{+2\epsilon}$	$2\epsilon\sigma_x$	$4\epsilon^2$	2ϵ
$U_{+\epsilon}$	$\epsilon\sigma_x$	ϵ^2	ϵ
$U_{-\epsilon}$	$-\epsilon\sigma_x$	ϵ^2	ϵ
$U_{-2\epsilon}$	$-2\epsilon\sigma_x$	$4\epsilon^2$	2ϵ

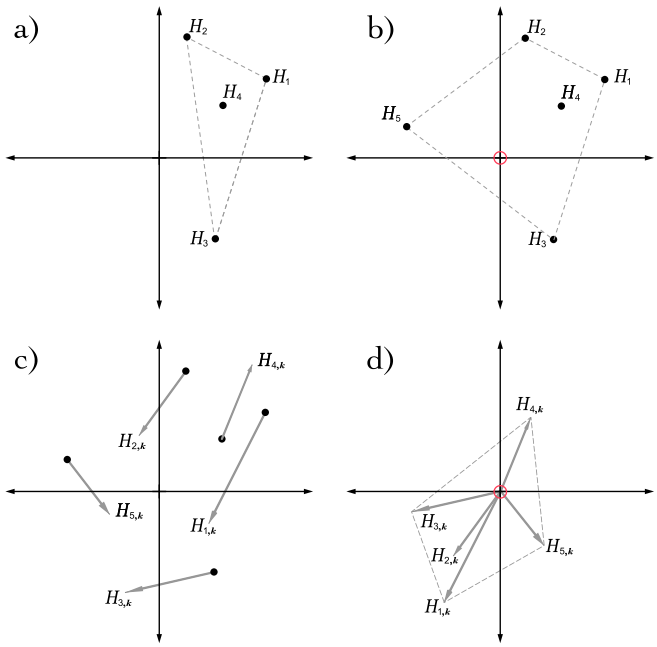

FIG. 2. A target unitary gate can be implemented a number of ways, each with a different effective Hamiltonian error. These error Hamiltonians lie in a vector space. a) Four effective Hamiltonians. The origin is not contained in their convex hull, so there are no 0MUPs. b) The origin is contained in the covex hull after adding an additional control solution. Because there are more than $n+1$ implementations, there exist an infinite number of 0MUPs. c) The error Hamiltonians shown with their derivative with respect to a control parameter. As this parameter drifts, a 0MUP may drift, leading to a first-order error. d) The derivatives also lie in a vector space. If the origin lies in their convex hull, then it may be possible to construct a 1MUP.

Alternatively, [Draw box around the (2,4),(2,3),(3,4) elements of the G matrix]. We can then eliminate off diagonal entries, resulting in a matrix that is guaranteed to be Pauli stochastic, and thus somewhat easily simulable. [more here about alternative method. Also discuss Fig 2.].

[END]

III. CONSTRUCTING USEFUL UNITARY PROCESSES

[write an introduction to this section here] In general, we want to construct a problem so that 0 is in the convex hull of some set of vectors. These vectors might represent the off-diagonal elements of the error maps, or they could be the error generator. As we'll see in section ... we could also include the derivatives with respect to various parameters. With appropriate weighting, we could minimize any number of effects. Which optimization we choose will depend on the approach. [Then go into examples below.]

A. First-order generators

[say that we are linearizing the diamond distance calculation - rewrite this whole thing] If our primary goal is for our MUP to have increased performance, then a useful target for constructing MUPs is instead minimizing the diamond norm. As in Section II A, the MUP that minimizes the diamond norm has matrix elements given by:

$$\mathcal{M}_{jk} = \frac{1}{2}\mathrm{Tr}\left(\sigma_j^\dagger U_{-\epsilon}\sigma_k U_{-\epsilon}^\dagger\right) + \frac{1}{2}\mathrm{Tr}\left(\sigma_j^\dagger U_{\epsilon}\sigma_k U_{\epsilon}^\dagger\right) \quad (13)$$

This channel will have a diamond norm proportional to ϵ^2, whereas any other combination will have a linear dependence on ϵ. Contrast this with the fact that any linear combination of $\{U_\epsilon, U_{-\epsilon}\}$ minimizes the AGI of the resulting channel with a value of ϵ^2. While selecting the convex mixture of unitaries that minimizes the AGI is trivial, the problem of efficiently minimizing the diamond norm can be difficult, and may be the relevant quantity to consider depending on the application. However, as discussed in [4], a sufficient condition to minimize the diamond norm of a MUP with error generators $\{H_j\}$ to first order is:

$$\sum \omega_j H_j = 0 \quad (14)$$

In [4] Campbell constructs an algorithm that, given an oracle to approximate unitaries, finds a MUP with this property. Alternatively one might ask the question: *Given a collection of controls, how can I produce a MUP with minimal diamond norm?* Such a situation may arise if there is a natural family of controls that implement a desired gate, or if, as we consider later (Section **??**), we simply randomly generate a collection of gate implementations. In these situations, one can equivantly use convex optimization to solve this problem. Consider the matrix whose rows are the vectorized Hamiltonians at our disposal, i.e. for m, $n \times n$ Hamiltonians:

$$\mathbf{H} = \begin{pmatrix} H_{1_{11}} & H_{2_{21}} & \dots \\ \vdots & \ddots & \\ H_{m_{11}} & & H_{m_{nn}} \end{pmatrix} \quad (15)$$

If our weighting vector for our MUP is ω, we can rewrite this sum as a matrix product whose two-norm will be zero if and only if the sum is zero. Additionally, this optimization needs to be constrained so that ω only contains positive values that sum to one. This forms a convex optimization problem, and can be written as:

$$\underset{\omega_j \geq 0, |\omega|_1 = 1}{\textbf{minimize}} : ||\mathbf{H}^T\omega||_2 \quad (16)$$

Linearly constrained minimization problems with quadratic cost functions like this have been proven to be efficiently solvable by methods like the Elipsoid Method[citation needed], however these have very poor average case performance. However, there are many existing convex solver software packages that solve these problems efficiently in practice, with some formal proofs showing runtimes as good as $\mathcal{O}(\log\frac{1}{\epsilon}L^2n^4)$, where ϵ is the accuracy in the solution, L is the number of bits in the input, and n^4 is the number of variables.

B. Off-diagonals

C. Adding robustness

[this section is a little bit heavy on the notation - is there anything we can do to ease that? Perhaps in the introduction we can specify the problem a little more clearly and then this can be just a restriction of that general model] While mixed unitary processes offer significant improvements to gate performance, they fail to take into account the reality that most control electronics experience drift over time scales relevant to QIP performance. Because of this drift, the quality of the MUP will degrade. Thus, we would like to design MUPs that are *robust* to this drift. To enforce robustness, we can consider higher derivatives of the Hamiltonians in Equation 14. Instead of only requiring that the 0^{th} derivative averages to zero, we will impose a similar condition on the derivatives of the Hamiltonians with respect to parameters that may drift:

$$D_j^n = \frac{1}{n!}\frac{\partial^n}{\partial\delta_{i_1}\dots\partial\delta_{i_n}}H_j(\vec{\delta})|_{\delta=\vec{0}} \quad (17)$$

If the dimension of $\vec{\delta}$ is d, the indices $i_0, \dots, i_n$ take on values in $1, \dots, d$ and this matrix has d^n entries. We say that a mixed unitary process is robust to order ℓ (an ℓMUP) if for all $1 \leq k \leq \ell$:

$$\sum_j \omega_j (\sum_{n=0}^{k} D_j^n)^n = \vec{0} \quad (18)$$

In particular, we see that a 0MUP satisfies Equation 14. More generally, these conditions imply that an ℓMUP is insensitive to the ℓ^{th} order in drift in $\vec{\delta}$. To see this, we can rewrite the error on each control in the MUP as:

$$\begin{aligned}\tilde{\mathcal{G}}_j(\vec{\delta}) = \exp(-i(H_j(\vec{0}) + \frac{\partial}{\partial\delta_i}H_j(d\delta_i) \\ + \frac{1}{2}\frac{\partial^2}{\partial\delta_i\partial\delta_k}H_j(d\delta_i d\delta_k) + \dots))\mathcal{G}\end{aligned} \quad (19)$$

By Taylor expanding Equation 19 in $\vec{\delta}$, one finds that the the first ℓ derivatives of an ℓMUP will be zero. Furthermore, if we are only interested in being first order insensitve to drift and can find controls such that $|D_j^n| \approx \epsilon$, we can approximate Equation 18 as:

$$\sum \omega_j D_j^n = 0 \quad (20)$$

This condition guarantees that errors will be supressed quadratically for all derivatives up to order ℓ. A proof is included in the appendix that generalizes the Hastings-Campbell Mixing Lemma in [4]. Namely, Campbell showed that if $0 \in \text{Conv}[\{H_i(\vec{\delta})\}]$, where Conv is the convex hull of its arguments, then an $\vec{\omega}$ exists that quadratically decreases the diamond norm. We prove that $0 \in \text{Conv}[\{D_j^n(\vec{\delta})\}]$ implies there is an $\vec{\omega}$ exists that quadratically decreases the ℓ^{th}-order sensitivity of the diamond norm of an ℓMUP. Figure 2 gives geometric intuition for the conditions required to produce an ℓMUP.

To generate robustly mixed unitary processes, we first define the vectorized derivative matrix D^ℓ in a similar way to Equation 15:

$$\mathbf{D}^\ell = \begin{pmatrix} D^\ell_{1_{11}} & D^\ell_{2_{21}} & \cdots \\ \vdots & \ddots & \\ D^\ell_{m_{11}} & & D^\ell_{m_{nn}} \end{pmatrix} \tag{21}$$

Using this, we can then solve the following convex optimization problem, generalizing Equation 16:

$$\begin{aligned} &\underset{\omega_j \geq 0, |\omega|_1 = 1}{\textbf{minimize}} : ||\mathbf{D}^{\ell^T}\omega|| \\ &\textbf{subject to: } \forall n < \ell, \sum \omega_j D_j^n = \vec{0} \end{aligned} \tag{22}$$

with D_n^j defined in Equation 19. The time to solve these convex optimization problems is independent of the number of rows, which in our case is the number of drifting parameters. However, the size of the matrix that must be computed before solving grows as Nd^ℓ, and the runtime of solving the optimization problem grows as N^4, with d being the number of drifting parameters, and N being the number of controls.

D. Hamiltonian Norm Regularization

While up until this point, these particular convex optimization problem could be solved using a system of linear equations, casting them as convex optimization problems allows us to penalize the cost function to encourage different behavior in the solution. In particular, while this minimization problem is sufficient for quadratically decreasing the diamond norm relative to the *worst* controls in the collection, it does not preferentially select the controls with the least error. That is to say, both $\{U_{+2\epsilon}, U_{-2\epsilon}\}$ and $\{U_\epsilon, U_{-\epsilon}\}$ from Section II A satisfy Equation 16. To encourage the inclusion of controls with smaller error, we may impose a penalty proportional to the norm of the included Hamiltonians. In our case, we choose to penalize for the ℓ_2 ($||\cdot||_2$) norm, and thus we modify our cost function to be:

$$\begin{aligned} &\underset{n\in[N]}{\textbf{minimize}}\{ \\ &\quad \underset{\omega_j \geq 0, |\omega|_1 = 1}{\textbf{minimize}} : ||\mathbf{D}^{\ell^T}\omega|| + \eta \sum \omega_j ||D_j^0||_2 \\ &\quad \textbf{subject to: } \forall n < \ell, \sum \omega_j D_j^n = 0 \\ &\} \end{aligned} \tag{23}$$

with $\eta \geq 0$. By making η larger, we can supress the diamond norm to second order, while ensuring that we are selecting those that will give us a smaller prefactor. The absolute value of η depends on the particular numerical values in $\mathbf{D}^\ell$. Naively generating the 1Q 0MUP in the previous section results in nontrivial support on all the members of the control family. However, by rewriting the minimization to impose this sparsity constraint discussed in Section **??**, the resulting 0MUP uses just five of the controls. This shows that through adding constraints to our optimization routine, we can make the MUP practically useful.

E. Sparsity Constraints

As a practical consideration, we would also like to regularize our objective function to enforce sparsity. Control electronics often have a limited amount of waveform memory, and thus it is important that MUPs have nontrivial probability support on a small number of controls. As an example of where this would be necessary is given in Figure 2. In b), it is clear that H_4 is unecessary to contain the origin in the convex hull of the error generators. Thus we would prefer that our solution, if we are forming a 0MUP, ignores H_4. However, if we additionally want our controls to form a 1MUP, we see from Subfigure D that we need H_4 in our control set, in which case we would like our algorithm to exclude H_1, since its derivative is contained in the convex hull of the others'. Thus we would like to be frugal in which controls we select. In many machine learning contexts, lasso regularization [8] can be used to enforce sparsity in solutions, however this is insufficient in this context as we already constrain the one norm of ω to be one. Conveniently, the problem of enforcing sparsity in such situations has been considered in [9] and can be expressed via another convex program that extends Equation 16:

$$\begin{aligned} &\underset{n\in[N]}{\textbf{minimize}}\{ \\ &\quad \underset{\substack{\omega_j \geq 0, |\omega|_1 = 1, \\ t \geq 0}}{\textbf{minimize}} : ||\mathbf{D}^{\ell^T}\omega|| + t \\ &\quad \textbf{subject to: } \omega_n > \frac{\lambda}{t} \\ &\qquad\qquad \forall n < \ell, \sum \omega_j D_j^n = 0 \\ &\} \end{aligned} \tag{24}$$

with $\lambda \geq 0$. As with η in the last subsection, the optimal value of this parameter is problem specific, depending on $\mathbf{D}^{\ell}$, and how sparse the solution needs to be. For λ small, the problem reduces to the original problem, and for λ large we see that if ω is not sparse then t must be very large, which increases the cost of that particular solution. In both the 0MUP and the 1MUP case in our numerical implementation we imposed the same ℓ_2 penalty, so that the algorithm preferentially selects controls with smaller errors. Adding this constrain made the 0MUP perform better at the origin by nearly an order of magnitude, and moved the 1MUP from being out-performed by the 0MUP for all detunings, to out performing by nearly an order of magnitude when there is .1% drift in the controls. Imposing this constraint allows us to trade off flatness at the origin for performance.

IV. RESULTS

A. Experimental

Here we present experimental results from implementing this routine on a fixed-frequency superconducting transmon qubit. In particular, we used qubit 8 on the Rigetti 19Q-Acorn chip, whose characterization can be found in [10]. To implement a MUP on this qubit, four incorrectly calibrated Gaussian pulses were produced by scaling the pulseshape amplitude for a calibrated 10 sample 50ns $RX(\frac{\pi}{2})$ pulse by 106.4%, 103.9%, 93.7% and 91.2%.

As discussed in the previous section, we chose here to minimize the off diagonal elements of the process matrix. To benchmark the quality of the MUP, we then performed six randomized benchmarking experiments[11]: one for each over– and under–calibrated pulse, one for the calibrated pulse, and one for the mixed process. We used 1000 shots per experiment, 10 sequences per sequence length, for sequence lengths of $2, 4, 8, 16, 32$ and 64. In each case, our Clifford operations were decomposed into RX($\frac{\pi}{2}$)) and RY($\frac{\pi}{2}$)) pulses. In our implementation, these gates are implemented using the same pulse envelope definitions and control electronics, phase shifted by $\frac{\pi}{2}$ radians, and are therefore subject to identical miscalibration errors. The results are shown in Figure 3 for sequence lengths $L = 64$. Fitting to the randomized benchmarking decay curves, we find one-qubit gate fidelities of 99.3% for the calibrated pulse, 98.9% for Pulse1, 99.1% for Pulse2, 98.9% for Pulse3, 98.5% for Pulse4, and 99.2% for the MUP, demonstrating that it performs almost as well as the calibrated pulse, and better than the constituent pulses.

Additionally, by minimizing the off-diagonal elements of the process matrix, we expect to produce a process with minimal coherent error. To see that this is the case, we cite the results in [12]. For non-Markovian error models, noise will manifest as gamma distributed points for each sequence length. On the other hand, Markovian noise, such as depolarizing noise, will result in Gaussian distributed fidelity estimates for each randomized benchmarking sequence length. We see that the coherently miscalibrated controls in our RB experiment have long tails, consistent with gamma distributed random variables, while the calibrated and randomized implementations both have much shorter tails, consistent with Gaussian distributed random variables. Thus, our experiment demonstrates that not only is the performance of the MUP better than the constituent gates, it also has a signicantly less-coherent error channel.

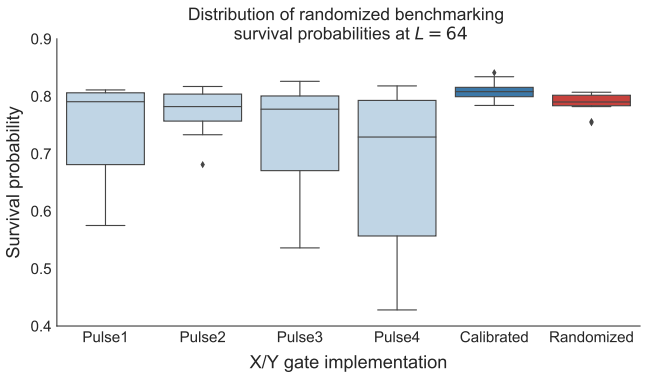

FIG. 3. Randomized benchmarking experiments ran using different pulse definitions. The first four boxes result from using each of four different implementations of the $\pi/2$ rotations. The coherent noise present in these implementations leads to large variance of the survival probability over sequences. The fifth (dark blue) box illustrates the survival probability using a highly-tuned gate implementation. It displays improved average survival probability as well as reduced variance. The final box (dark red) illustrates the distribution over survival probabilities for a randomized MUP composed of Pulse1 through Pulse4. It performs comparably to the highly-calibrated implementation in both average survival probability and variance over random sequences. The reduced variance of the MUP is a tell-tale sign of reduced coherent error in the effective channel.

B. Numerical Implementation

In the following numerical results, we explore using the methods in Section III to build MUPs. We consider the following model for a single tunable qubit:

$$H(\delta, \epsilon, t) = \epsilon\sigma_z + (1+\delta)(c_x(t)\sigma_x + c_y(t)\sigma_y) \qquad (25)$$

We use the GRAPE algorithm[13] with N=25 steps and total evolution time of π to generate 100 candidate controls. In our implementation, we modify the gradient so that we find controls that perform well in a Gaussian weighted neighborhood, with a standard deviation of $\sigma = .001$. We assume that the errors on σ_x and σ_y are perfecly correlated, as is the case in systems that implement RZ rotations with phase shifts of the control

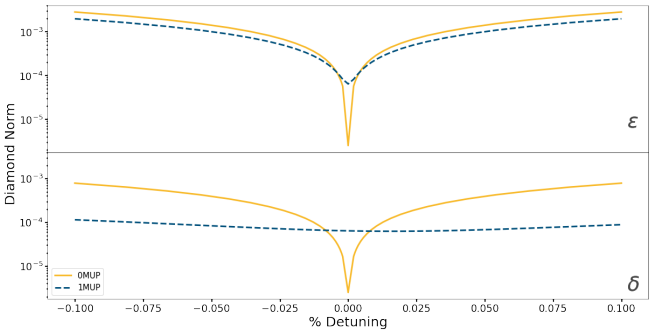

FIG. 4. Numerical results comparing a 0MUP to a 1MUP for a single tunable qubit, for $RY(\frac{\pi}{2})$. The results are qualitatively similar to those for $RX(\frac{\pi}{2})$. In this case the 0MUP outperforms both the 1MUP by two orders of magnitude, and the constituent controls by three orders of magnitude at the origin. However, varying over δ we see that the 1MUP outperforms the 0MUP by up to an order of magnitude when there is .1% drift in the qubit control amplitudes.

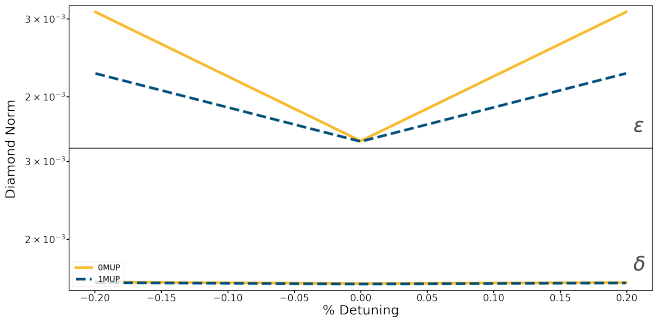

FIG. 5. Numerical results comparing a 0MUP to a 1MUP for a pair of tunable qubits, with a resonant exchange interaction. Shown with lower alpha values are example constituent controls. The 0MUP and 1MUP can be seen to outperform these controls by half of an order of magnitude at the origin. For all detuning values the 1MUP performs as well or better than the 0MUP. When there is .2% drift in the qubit frequency, the 1MUP outperforms members of the control famlies by almost an order of magnitude in diamond norm. Similarly, for .2% drift in the qubit control amplitude, we see that the 1MUP outperforms the the constituent controls by over half an order of magnitude.

signal. Solving the optimization problem defined in Section **??** yields similar MUPs for $RX(\frac{\pi}{2})$ and $RY(\frac{\pi}{2})$, with the results for $RY(\frac{\pi}{2})$ shown in Figure 4. These results demonstrate several properties that make MUPs both useful and tractable.

In our two-qubit example we consider the following model for two tunable qubits coupled by a resonant exchange interaction, similar to that in [14]:

$$H(\vec{\delta},\vec{\epsilon},t) = \sum_{j=1}^{2}(\epsilon_j\sigma_z^j + (1+\delta_j)(c_x^j x(t)\sigma_x^j + c_y^j(t)\sigma_y^j)) + \frac{1}{10}(XX+YY) \tag{26}$$

In this example it was infeasible to use GRAPE to return non-trivial solutions. Instead we manually selected piecewise constant echoing sequences with 500 steps and total evolution time of $\frac{5\pi}{2}$. In particular, we considered $RX(\pi)$, $RX(-\pi)$, $RY(\pi)$ and $RY(-\pi)$ bang-bang sequences [15], consisting of all combinations of simultaneous π pulses activated at multiples of 8 steps from the beginning of the controls, and the same multiple of 8 steps prior to the end of the controls. To give the control family a variety of RF errors, we added on uniformly distributed errors to each π pulse, between $-.25\%$ and $.25\%$.

In this example, we find more modest improvements to performance, as shown in Figure 5. There are now four free parameters to optimize over, and the uncontrolled entangling interaction means that there is little room for variation in the controls. Nonetheless, using a MUP improves performance by half of an order of magnitude at the origin relative to the constituent controls, and up to an order of magnitude away from the origin. For all values of the drifting parameters we see that the 1MUP performs as well or better than 0MUP.

V. CONCLUSION AND FUTURE WORK

We have shown numerically that using MUPs can reduce coherent error on a quantum channel by more than an order of magnitude in diamond norm, over a wide range of quasi-static values of noise. In addition, we have demonstrated that these approximate controls can be generated through optimal control (GRAPE), and that the minimization problem is tractable.

Future directions for this work include demonstrating the routine experimentally on a two-qubit gate, moving the random gate selection from a precompilation step to runtime logic onboard the control electronics, investigating other optimization routines such as CRAB [16] and GOAT[17], and using more sophisticated benchmarking routines such as GST[18] to quantitatively investigate the performance of our method.

Another interesting area of research would be using model-free approaches. The numerical work in the paper assumes access to a model of the system, however an experimentalist may not have a model readily available to describe the system, e.g. in the presence of unknown on-chip crosstalk, or an uncalibrated transfer function of the system. Even if a model is available, it might be computationally inconvenient to simulate, i.e. for more than a few qubits.

In these situations, one approach would be to use *in-situ* optimal control techniques [19–21] to generate candidate controls, and then use an optimizer like Nealder-Mead to perform the minimization. While performing a complete optimization in this way would require full process tomography, one could intead optimize via partial tomography. By selecting pre– and post –rotations that correspond to measuring Pauli-moments of interest

in the Hamiltonian, such as unwanted $Z \otimes Z$ crosstalk, one could perform optimization over fewer parameters.

VI. ACKNOWLEDGEMENTS

Sandia National Laboratories is a multimission laboratory managed and operated by National Technology and Engineering Solutions of Sandia, LLC, a wholly owned subsidiary of Honeywell International, Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525.

[1] R. Barends, J. Kelly, A. Megrant, A. Veitia, D. Sank, E. Jeffrey, T. C. White, J. Mutus, A. G. Fowler, B. Campbell, Y. Chen, Z. Chen, B. Chiaro, A. Dunsworth, C. Neill, P. O'Malley, P. Roushan, A. Vainsencher, J. Wenner, A. N. Korotkov, A. N. Cleland, and J. M. Martinis, Nature **508**, 500 (2014).

[2] C. Ballance, T. Harty, N. Linke, M. Sepiol, and D. Lucas, Physical Review Letters **117** (2016), 10.1103/physrevlett.117.060504.

[3] J. Preskill, "Quantum computing and the entanglement frontier," (2012), arXiv:1203.5813.

[4] E. Campbell, Physical Review A **95** (2017), 10.1103/physreva.95.042306.

[5] M. B. Hastings, "Turning gate synthesis errors into incoherent errors," (2016), arXiv:1612.01011.

[6] E. Campbell, "A random compiler for fast hamiltonian simulation," (2018), arXiv:1811.08017.

[7] K. Rudinger, T. Proctor, D. Langharst, M. Sarovar, K. Young, and R. Blume-Kohout, "Probing context-dependent errors in quantum processors," (2018), arXiv:1810.05651.

[8] R. Tibshirani, Journal of the Royal Statistical Society. Series B (Methodological) , 267 (1996).

[9] M. Pilanci, L. E. Ghaoui, and V. Chandrasekaran, in *Advances in Neural Information Processing Systems 25*, edited by F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger (Curran Associates, Inc., 2012) pp. 2420–2428.

[10] J. S. Otterbach, R. Manenti, N. Alidoust, A. Bestwick, M. Block, B. Bloom, S. Caldwell, N. Didier, E. S. Fried, S. Hong, P. Karalekas, C. B. Osborn, A. Papageorge, E. C. Peterson, G. Prawiroatmodjo, N. Rubin, C. A. Ryan, D. Scarabelli, M. Scheer, E. A. Sete, P. Sivarajah, R. S. Smith, A. Staley, N. Tezak, W. J. Zeng, A. Hudson, B. R. Johnson, M. Reagor, M. P. da Silva, and C. Rigetti, "Unsupervised machine learning on a hybrid quantum computer," (2017), arXiv:1712.05771.

[11] E. Magesan, J. M. Gambetta, and J. Emerson, Physical Review Letters **106** (2011), 10.1103/physrevlett.106.180504.

[12] H. Ball, T. M. Stace, S. T. Flammia, and M. J. Biercuk, Physical Review A **93** (2016), 10.1103/physreva.93.022303.

[13] N. Khaneja, T. Reiss, C. Kehlet, T. Schulte-Herbrüggen, and S. J. Glaser, Journal of Magnetic Resonance **172**, 296 (2005).

[14] D. C. McKay, S. Filipp, A. Mezzacapo, E. Magesan, J. M. Chow, and J. M. Gambetta, Physical Review Applied **6** (2016), 10.1103/physrevapplied.6.064007.

[15] L. Viola and S. Lloyd, Physical Review A **58**, 2733 (1998).

[16] T. Caneva, T. Calarco, and S. Montangero, Physical Review A **84** (2011), 10.1103/physreva.84.022326.

[17] S. Machnes, E. Assémat, D. Tannor, and F. K. Wilhelm, Physical Review Letters **120** (2018), 10.1103/physrevlett.120.150401.

[18] R. Blume-Kohout, J. K. Gamble, E. Nielsen, K. Rudinger, J. Mizrahi, K. Fortier, and P. Maunz, Nature Communications **8** (2017), 10.1038/ncomms14485.

[19] R.-B. Wu, B. Chu, D. H. Owens, and H. Rabitz, Physical Review A **97** (2018), 10.1103/physreva.97.042122.

[20] J. Kelly, R. Barends, B. Campbell, Y. Chen, Z. Chen, B. Chiaro, A. Dunsworth, A. Fowler, I.-C. Hoi, E. Jeffrey, A. Megrant, J. Mutus, C. Neill, P. O'Malley, C. Quintana, P. Roushan, D. Sank, A. Vainsencher, J. Wenner, T. White, A. Cleland, and J. M. Martinis, Physical Review Letters **112** (2014), 10.1103/physrevlett.112.240504.

[21] C. Ferrie and O. Moussa, Physical Review A **91** (2015), 10.1103/physreva.91.052306.

VII. APPENDIX

A. Robust Mixing Lemma

We begin by generalizing Lemma 2 from [4]. If all of our error Hamiltonians $\{H_j\}$ have bounded error, that is:

$$||H_j|| \leq c \tag{27}$$

Then we may consider the derivative of any mixture of unitaries as:

$$\frac{d}{d\vec{\delta}} \sum_j \omega_j e^{iH_j} = \sum_j \omega_j i \frac{d}{d\vec{\delta}}(H_j) e^{iH_j} \tag{28}$$

By assumption, the derivatives of the first order term sum to zero, and so we see by A3 in [4] that

$$||\frac{d}{d\vec{\delta}} \sum_j \omega_j e^{iH_j}|| \leq \sum_j \omega_j ||i \frac{d}{d\vec{\delta}}(H_j)(iH_j + \sum_{n=2}^{\infty} \frac{(iH_j)^n}{n!})|| \leq \sum_j \omega_j (c^2 + \frac{c^2}{2}) \leq c^2 + \frac{c^3}{2} \tag{29}$$

Where we have used the fact that $||AB|| \leq ||A||||B||$ with $||\cdot||$ being the ∞-norm. Additionally,

$$\begin{aligned}
||\frac{d}{d\vec{\delta}} e^{iH_j}|| &= ||\frac{d}{d\vec{\delta}}(iH_j) e^{iH_j}|| && (30)\\
&\leq ||\frac{d}{d\vec{\delta}}(iH_j)|| + ||\frac{d}{d\vec{\delta}}(iH_j) H_j|| + ||\frac{d}{d\vec{\delta}}(iH_j) \sum_{n=2}^{\infty} \frac{1}{(n)!}(iH_j)^n|| && (31)\\
&\leq ||\frac{d}{d\vec{\delta}}(iH_j)|| + ||\frac{d}{d\vec{\delta}}(iH_j)|| \cdot ||H_j|| + ||\frac{d}{d\vec{\delta}}(iH_j)|| \cdot ||\sum_{n=2}^{\infty} \frac{1}{(n)!}(iH_j)^n|| && (32)\\
&\leq c + c^2 + \frac{c^3}{2} && (33)
\end{aligned}$$

And so we see that Lemma 2 generalizes. We will use the above to upper bound the 1-norm (and hence diamond norm) in the following.

Next we generalize the Mixing Lemma from [4]. We must assume that the 1-norm is uniformly differentiable. In that case, we see that:

$$\begin{aligned}
&= \frac{d}{d\vec{\delta}} \sup |\mathcal{I} \otimes \mathcal{I} - \mathcal{E}(\vec{\delta}) \otimes \mathcal{I}|_1 && (34)\\
&= \sup \frac{d}{d\vec{\delta}} |\mathcal{I} \otimes \mathcal{I} - \mathcal{E}(\vec{\delta}) \otimes \mathcal{I}|_1 && (35)\\
&\leq \sup |\frac{d}{d\vec{\delta}} (\mathcal{I} \otimes \mathcal{I} - \mathcal{E}(\vec{\delta}) \otimes \mathcal{I})|_1 && (36)\\
& && (37)
\end{aligned}$$

And so we see it is sufficient to upperbound:

$$|\frac{d}{d\vec{\delta}} (\mathcal{I} \otimes \mathcal{I} - \mathcal{E}(\vec{\delta}) \otimes \mathcal{I})|_1 \tag{38}$$

From Equation 18 in [4]:

$$(\mathcal{V} \circ \mathcal{E} - \mathbb{1})(X) = \sum_j \omega_j (\tilde{\delta}_j X + X \tilde{\delta}_j^\dagger + \tilde{\delta}_j X \delta_j^\dagger) \tag{39}$$

$$\frac{d}{d\vec{\delta}} (\mathcal{V} \circ \mathcal{E} - \mathbb{1})(X) = \sum_j \omega_j (\frac{d}{d\vec{\delta}} \tilde{\delta}_j X + X \frac{d}{d\vec{\delta}} \tilde{\delta}_j^\dagger + \frac{d}{d\vec{\delta}} \tilde{\delta}_j X \delta_j^\dagger + \tilde{\delta}_j X \frac{d}{d\vec{\delta}} \delta_j^\dagger) \tag{40}$$

$$\tag{41}$$

Thus, by Hölder's inequality we find:

$$||\frac{d}{d\vec{\delta}}(\mathcal{V}\circ\mathcal{E}-\mathbb{1})(X)||_1 \leq ||\sum_j \omega_j \frac{d}{d\vec{\delta}}\tilde{\delta}_j|| + ||\sum_j \omega_j \frac{d}{d\vec{\delta}}\tilde{\delta}_j^\dagger|| + \sum_j \omega_j ||\frac{d}{d\vec{\delta}}\tilde{\delta}_j|| \cdot ||\delta_j^\dagger|| + \sum_j \omega_j ||\tilde{\delta}_j|| \cdot ||\frac{d}{d\vec{\delta}}\delta_j^\dagger|| \tag{42}$$

By assumption, all of these terms are neglible, and therefore we find:

$$||\frac{d}{d\vec{\delta}}(\mathcal{V}\circ\mathcal{E}-\mathbb{1})(X)||_1 \leq 2a^2 + 2b^2 \tag{43}$$

And so we see that the mixing lemma extends to first order. In a similar way, it can be extended to arbitary order, and thus if we can find constructions that satisfy the generalized Lemma 2, we can use the Robust Mixing Lemma to conclude that there exists a weighting with linearly supressed diamond norm.