Advantages of mixed quantum gates for information processing

Anthony M. Polloreno*
Rigetti Computing, Berkeley, CA

Kevin C. Young
Sandia National Laboratories, Livermore, CA

(Dated: March 25, 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

[discuss variance narrowing and cite Biercuk] 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 [1–3], 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 [4] milestone has yet to be definitively reached. A critical limiting factor, of course, is errors in the quantum gate operations.

The ultimate impact of a gate error on a quantum circuit depends strongly on both the magnitude and the nature of the error. 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 [cite]. Contrast this against random, or *stochastic*, errors, which often result from high-frequency noise in the controls or the environment. Systems with stochastic errors can usually 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 [5].

Quantitative bounds on the performance of a quantum circuit composed of faulty gates can be constructed using the diamond distance, $||\cdot||_\diamond$. This particular metric, defined in Section II, has the appealing property of being subadditive, namely for a channels $\mathcal{F}_1$, $\mathcal{F}_2$ we have:

$$||\mathcal{F}_1 \circ \mathcal{F}_2||_\diamond \leq ||\mathcal{F}_1||_\diamond + ||\mathcal{F}_2||_\diamond \tag{1}$$

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, and while these errors can often be identified and reconstructed using various tomographic techniques, their impact on a given quantum difficult to predict. Additionally, 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[6–8], however, has shown that coherent noise can be strongly suppressed by probabilistically mixing several distinct implementations of the target quantum gates. In this case, the resulting effective quantum process has a diamond distance that grows only quadratically in the over/under rotation angle of the component gates.

[Discuss randomized compiling]

In this article we discuss various applications of these mixed unitary controls, and show that the advantages of

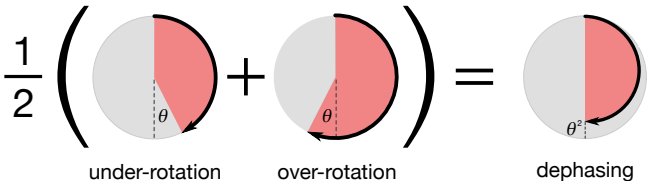

FIG. 1. An example of a mixed quantum gate. Using optimal control, two implementations of a Z_π gate are designed to have equal and opposite sensitivity to errors (if one implementation over-rotates by angle θ, then the other *under*-rotates by θ). Each time the gate is used, one of these implementations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

* Email: anthony@rigetti.com

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 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.

[Discuss the advantages of convex/quadratic programs. Discuss randomized compiling.]

II. REPRESENTING ERRORS

A. Representing Errors in Quantum Gates

Suppose that one wishes to apply a unitary gate, G, on some quantum device. Implementing this operation on a real system generally involves the application of a sequence of classical control fields to some set of qubits. But fluctuations in the environment or imperfections in these controls can cause the state of the qubits to change in a way that is different from what was intended. If the device is fairly stable with time and context[9], then we can accurately model the actual gate dynamics action using a completely positive, trace-preserving (CPTP) map, $\tilde{\mathsf{G}}$. This map can always be written as $\tilde{\mathsf{G}} = \mathsf{E} \circ \mathsf{G}$, where $\mathsf{E} = \tilde{\mathsf{G}} \circ \mathsf{G}^{-1}$ is the *error map*, which is itself CPTP because G is unitary.

CPTP maps possesses a number of useful representations, including Kraus operators[10], Choi matrices[11], and Jamiolkowsi states[12]. But for the purposes of this article, the *process matrix* representation will be particularly convenient[13], and we shall denote the process matrix associated with a given CPTP map G with the corresponding calligraphic character, $\mathcal{G}$. For a d-dimensional quantum state, the process matrix is a $d^2 \times d^2$ matrix. A key feature of process matrices is that they act through the usual matrix multiplication on the vectorized quantum state:

$$\mathsf{vec}\left(\tilde{\mathsf{G}}(\rho)\right) = \tilde{\mathcal{G}} \cdot \mathsf{vec}\left(\rho\right) \tag{2}$$

$$= \mathcal{E} \cdot \mathcal{G} \cdot \mathsf{vec}\left(\rho\right) \tag{3}$$

We shall work in the basis Pauli matrices, defining $\Sigma = \{I, \sigma_x, \sigma_y, \sigma_z\}^{\otimes n}$ as the collection of all 4^n n-qubit Pauli operators (including the identity). In this basis,

$$\mathsf{vec}\left(\rho\right)_i = \mathrm{Tr}\left(\rho \cdot \Sigma_i\right)/2^n, \tag{4}$$

and

$$(\tilde{\mathcal{G}})_{ij} = \mathrm{Tr}\left(\tilde{\mathsf{G}}\left(\Sigma_j\right) \cdot \Sigma_i\right)/2^n. \tag{5}$$

Process matrices written in Pauli basis are is often referred to as *Pauli transfer matrices* [14], and error maps take an especially nice form in this basis:

$$\mathcal{E} = \left(\begin{array}{c|c} 1 & \vec{0}^T \\ \hline \vec{m} & R \end{array}\right) \tag{6}$$

The top row of all trace-preserving (TP) maps is fixed to $\{1, 0, 0, 0, \cdots\}$ and the remainder of the first column, $\vec{m}$, describes any deviations from unitality, as could arise from amplitude damping []. If the error map is unitary, then the error is coherent, and the unital submatrix R is perfectly antisymmetric, corresponding to a rotation of the generalized Bloch vector. Importantly, if R is diagonal, then the error is Pauli stochastic, with each diagonal entry corresponding to the probability that its associated Pauli error occurs in each application of the gate.[1]

In what follows it will be useful to define the *error generator*, $\mathcal{L}$, associated with a faulty gate:

$$\mathcal{E} = \exp\left(\mathcal{L}\right) \approx \mathcal{I}_d + \mathcal{L}. \tag{7}$$

If an implemented gate is relatively close to the target, then the error generator will be small under any of the usual matrix norms.

B. Mixed Quantum Gates

Now suppose that we have access to an ensemble of different implementations of the target gate $\{\tilde{\mathsf{G}}_i\}$. Each time the gate is to be applied to the system, we randomly select an implementation from this ensemble such that the probability of drawing E_i is w_i (and we ensure that $\sum_i w_i = 1$). This procedure is operationally indistinguishable from a situation in which we always apply the effective operation,

$$\tilde{\mathsf{G}}_{\text{eff}} = \sum_i w_i \tilde{\mathsf{G}}_i, \tag{8}$$

which we call a *mixed quantum gate* or MQG. Error metrics can then be computed in terms of the effective error map,

$$\mathsf{E}_{\text{eff}} = \sum_i w_i \mathsf{E}_i. \tag{9}$$

Two important error metrics are the average gate infidelity (AGI), $\epsilon_{\mathcal{F}}$, and the diamond distance to the target,

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

$\epsilon_\diamond$ [], defined as:

$$\epsilon_{\mathcal{F}}(\mathsf{E}) = \frac{d^2 - \mathrm{Tr}\,\mathcal{E}}{d^2 + d} \tag{10}$$

$$\epsilon_\diamond(\mathsf{E}) = \frac{1}{2}\sup_\rho ||(\mathsf{I}_d \otimes \mathsf{I}_d)(\rho) - (\mathsf{E} \otimes \mathsf{I}_d)(\rho)||_1, \tag{11}$$

where $d = 2^n$ and I_d is the d-dimensional identity operator. In Eq. (10), we have written the AGI for an error map, E in terms of its associated process matrix, $\mathcal{E}$. If the error channel is purely stochastic, then $\epsilon_\diamond(\mathsf{E}) = \epsilon_{\mathcal{F}}(\mathsf{E})$, but if the error channel has a unitary component, then the diamond distance will generically be larger than the average gate infidelity [Wallman - Error rates in quantum circuits]. The diamond distance is subaddative [Watroos TQI book] under gate composition, so is particularly useful for constructing bounds on quantum circuit performance: the diamond distance of a circuit is less than or equal to the sum of the diamond distances for the component gates.

Because $\epsilon_{\mathcal{F}}$ is linear in the error map, we have:

$$\epsilon_{\mathcal{F}}(\mathsf{E}_{\text{eff}}) = \sum_i w_i\, \epsilon_{\mathcal{F}}(\mathsf{E}_i). \tag{12}$$

So there can be little hope of reducing the AGI by combining various implementations. However, the diamond distance is a nonlinear function of error map. As we show in the appendix (VII A):

$$\epsilon_\diamond(\mathsf{E}_{\text{eff}}) \leq \sum_i w_i\, \epsilon_\diamond(\mathsf{E}_i). \tag{13}$$

So by mixing various implementations, each with a different error channel, the resulting channel can have a diamond distance error less than any of the component implementations.

C. A simple example

As a simple example, consider a single-qubit and four different implementations of a PHASE gate: $\{3\pi, \pi, -\pi, 3\pi\}$ rotations about the σ_z-axis. If the control amplitude is too high, then the error maps associated with these four implementations are themselves unitary rotations about the σ_z-axis with rotation angles $\{3\eta, \eta, -\eta, -3\eta\}$, repectively (and $\eta \ll \pi$).

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 6. We implemented such a routine on a superconducting qubit.

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

$$\mathsf{E}(\eta) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\eta & \sin\eta & 0 \\ 0 & -\sin\eta & \cos\eta & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{14}$$

The associated diamond distance of this channel is $|\mathcal{I} - \mathsf{E}(\eta)|_\diamond \simeq \eta$, while the fidelity of the channel is $\mathcal{F}(\mathsf{E}(\eta)) = \eta^2$. If we were construct two channels, one with error $\mathsf{E}(\eta)$ and one with error $\mathsf{E}(-\eta)$, then we could construct a new channel with error $\mathsf{E}_{\text{eff}} = \frac{1}{2}(\mathsf{E}(\eta) + \mathsf{E}(\eta))$. The effective channel then has a matrix representation:

$$\mathsf{E}(\eta) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\eta & 0 & 0 \\ 0 & 0 & \cos\eta & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \simeq \mathcal{I} - \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & \eta^2/2 & 0 & 0 \\ 0 & 0 & \eta^2/2 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \tag{15}$$

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

$$G = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \tag{16}$$

The effective generator vanishes at first order.

$$\mathsf{E}(\eta) = \mathcal{I} + \eta G + \frac{1}{2}\eta^2 G^2 + \mathcal{O}(\eta^3) \tag{17}$$

$$\mathsf{E}_{\text{eff}} = \mathcal{I} + \sum_i \omega_i \eta_i G_i + \sum_i \omega_i \frac{1}{2}\eta_i^2 G_i^2 + +\mathcal{O}(\eta^3) \tag{18}$$

$$\mathsf{E}_{\text{eff}} = \mathcal{I} + \sum_i \omega_i \frac{1}{2}\eta_i^2 G_i^2 + \mathcal{O}(\eta^3) \tag{19}$$

Gate	H_{eff}	AGI	$\|\|\cdot\|\|_\diamond$
$U_{+2\eta}$	$2\eta\sigma_z$	$4\eta^2$	2η
$U_{+\eta}$	$\eta\sigma_z$	η^2	η
$U_{-\eta}$	$-\eta\sigma_z$	η^2	η
$U_{-2\eta}$	$-2\eta\sigma_z$	$4\eta^2$	2η

Alternatively, we might consider eliminating off-diagonal entries of $\mathsf{E}(\eta)$ [fix this]:

$$\begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & \boxed{-1} & \boxed{0} \\ 0 & 1 & \boxed{0} & \boxed{0} \\ 0 & 0 & 0 & 0 \end{pmatrix} \tag{20}$$

resulting in a matrix that is guaranteed to be Pauli stochastic, and thus somewhat easily simulable problem.

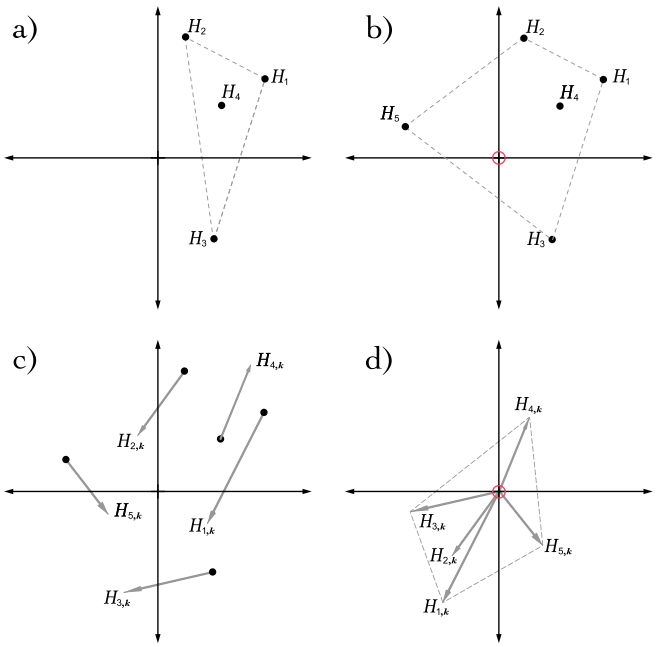

FIG. 2. A target unitary gate can be implemented a number of ways, each with a different effective Hamiltonian error. These error Hamiltonians lie in a vector space. a) Four effective Hamiltonians. The origin is not contained in their convex hull, so there are no 0MQCs. b) The origin is contained in the covex hull after adding an additional control solution. Because there are more than $n+1$ implementations, there exist an infinite number of 0MQCs. c) The error Hamiltonians shown with their derivative with respect to a control parameter. As this parameter drifts, a 0MQC 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 1MQC.

In the case of a single qubit, it is clear that our generator is expressible as $\hat{n}\cdot\vec{\Sigma}$, where $\vec{\Sigma}$ is again the vector of Pauli operators. The condition that the off-diagonal terms vanish is then again the same condition that the origin lies in the convex hull of the vectors $\vec{n}_i$ for each G_i. [more here about alternative method. Also discuss Fig 2.].

III. CONSTRUCTING USEFUL MIXED QUANTUM GATES

Motivated by the simple example provided in Section II C, we would like a general methodology for producing *useful* mixed quantum channels. In the examples that were discussed, all objective functions were linear, and reduced to finding a collection of controls whose convex hull contained the origin in some high-dimensional vector space. In those cases, the vectors might represent a collection of error generators or the off-diagonal elements of a collection of error maps. As we'll see in Section III B 1, we can also consider the derivatives of the error generators to minimize the sensitivity of the controls to drift. In this way, we see that by considering the appropriate collection of vectors, we can optimize our MQC for different purposes, and which routine we choose will depend on the quantities we are trying to minimize.

A. Optimization Targets

1. *First-order generators*

If our primary goal is for our MQC to have increased worst-case performance, then a useful target for constructing MQCs is to minimize the diamond norm. The diamond norm is a non-linear function that in general requires a convex optimizer to compute. However, if our errors are small enough, we can consider the linearized problem, and minimize the diamond norm to first-order in the error generator. As discussed in [6], a sufficient condition to minimize the diamond norm of an MQC with error generators $\{G_j\}$ to first order is:

$$\sum \omega_j G_j = 0 \tag{21}$$

In [6] Campbell constructs an algorithm that, given an oracle to approximate unitaries, generates controls and weightings to find an MQC with this property. Alternatively one might ask the question: *Given a fixed collection of controls, how can I produce a weighting with minimal diamond norm?* Such a situation might arise if there is a natural family of controls that implement a desired gate, or if, as we consider later (Section IV B), we randomly generate a collection of gate implementations. In these situations, one can use convex optimization to solve this problem. Given a collection of controls in our MQC, each will have an associated error channel E_i given by $\mathsf{E}_i\tilde{\mathsf{G}}_i = \mathsf{G}$. We can then consider the matrix whose columns are the vectorized generators of the error channels for each control, i.e. G_i such that $e^{G_i} = \mathsf{E}_i$. Then we will have m, $n\times n$ generators:

$$\mathbf{G} = \begin{pmatrix} G_{1_{1,1}} & G_{2_{1,1}} & \cdots \\ \vdots & \ddots & \\ G_{1_{n,n}} & & G_{m_{n,n}} \end{pmatrix} \tag{22}$$

If our weighting vector for our MQC 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 $\omega \geq 0$ and $|\omega|_1 = 1$. This forms a convex optimization problem, and can be written as:

$$\underset{\omega_j\geq 0, |\omega|_1=1}{\textbf{minimize}} : ||\mathbf{G}\omega||_2 \tag{23}$$

Linearly constrained minimization problems with quadratic cost functions like this have been proven to be efficiently solvable by methods like the Elipsoid

Method[15, 16], and there are many existing convex solver software packages that solve these problems efficiently in practice.

2. Off-diagonals

Of course, there are other properties of a channel that one might consider optimizing. Instead of minimizing the diamond norm, we might want our resulting MQC to have an error channel that is well-approximated by a Pauli-Stochastic channel, so that the error on our channel is amenable to classical simulation. We can write this error channel as a process matrix, vectorize the R submatrices in Equation 6, and consider the matrix of off-diagonal terms:

$$\mathbf{R} = \begin{pmatrix} R_{1_{2,1}} & R_{2_{2,1}} & \dots \\ \vdots & \ddots & \\ R_{1_{n-1,n-1}} & & R_{m_{n-1,n-1}} \end{pmatrix} \tag{24}$$

then again we consider the same minimization problem, where we replace $\mathbf{G}$ with $\mathbf{R}$:

$$\underset{\omega_j \geq 0, |\omega|_1 = 1}{\textbf{minimize}} : ||\mathbf{R}\omega||_2 \tag{25}$$

Previous authors have considered minimizing the diamond distance to the nearest Pauli or Clifford Channel [17], however this requires multiple evaluations of the diamond norm, and in particular does not have the restriction that the resulting channel be decomposable into a given family of controls. Our approach, on the other hand, guarantees that the resulting MQC is able to be implemented with the available controls, and is an easy-to-solve quadratic program.

B. Additional Constraints

1. Adding robustness

While mixed quantum channels 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 MQC will degrade. Thus, we would like to design MQCs that are robust to this drift. To enforce robustness, we can consider higher derivatives of the generators in Equation 21. Instead of only requiring that the error generators average to zero, we can impose a similar condition on the higher-order derivatives with respect to parameters that may drift. If our control generator is parameterized by a vector $\vec{\delta}$, then the quantity we will be interested in is:

$$D_j^n = \frac{1}{n!} \frac{\partial^n}{\partial \delta_{i_1} \dots \partial \delta_{i_n}} G_j(\vec{\delta})|_{\delta = \vec{0}} \tag{26}$$

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 quantum gate is robust to order ℓ (an ℓMQC) if for all $1 \leq k \leq \ell$:

$$\sum_j \omega_j (\sum_{n=0}^{k} D_j^n)^n = \vec{0} \tag{27}$$

In particular, we see that a 0MQC satisfies Equation 21. More generally, these conditions imply that an ℓMQC 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 MQC as:

$$\begin{aligned} \tilde{\mathsf{G}}_j(\vec{\delta}) = \exp(-i(G_j(\vec{0}) + \frac{\partial}{\partial \delta_i} G_j(d\delta_i) \\ + \frac{1}{2} \frac{\partial^2}{\partial \delta_i \partial \delta_k} G_j(d\delta_i d\delta_k) + \dots)) \mathsf{G} \end{aligned} \tag{28}$$

By Taylor expanding Equation 28 in $\vec{\delta}$, one can see that the the first ℓ derivatives of an ℓMQC will be zero. Furthermore, if we are only interested in being first order insensitve to drift and can find controls such that $|D_j^n| \leq \epsilon$, we can approximate Equation 27 as:

$$\sum \omega_j D_j^n + \mathcal{O}(\epsilon^2) = 0 \tag{29}$$

This condition guarantees that errors will be supressed quadratically for all derivatives up to order ℓ. Figure 2 gives geometric intuition for the conditions required to produce an ℓMQC. To generate robustly mixed quantum channels, we first define the vectorized derivative matrix $\mathbf{D}^\ell$ in a similar way to Equation 22:

$$\mathbf{D}^\ell = \begin{pmatrix} D^\ell_{1_{1,1}} & D^\ell_{2_{1,1}} & \dots \\ \vdots & \ddots & \\ D^\ell_{1_{n,n}} & & D^\ell_{m_{n,n}} \end{pmatrix} \tag{30}$$

Using this, we can then solve the following convex optimization problem, generalizing Equation 23:

$$\begin{aligned} &\underset{\omega_j \geq 0, |\omega|_1 = 1}{\textbf{minimize}} : ||\mathbf{D}^\ell \omega|| \\ &\textbf{subject to: } \forall n < \ell, \sum \omega_j D_j^n = \vec{0} \end{aligned} \tag{31}$$

with D_n^j defined in Equation 28. The size of the matrix that must be computed before solving grows as Nd^ℓ, and the time required to solve the optimization problem will depend on the particular optimizer and algorithm used.

2. 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 supressing the diamond norm to first order 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 C satisfy Equation 23. 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:

$$\begin{aligned}
&\textbf{minimize}\{\\
&\quad \underset{n\in[N]}{\textbf{minimize}} : ||\mathbf{D}^{\ell}\omega|| + \eta \sum \omega_j ||D_j^0||_2 \\
&\quad \underset{\omega_j \geq 0, |\omega|_1 = 1}{\textbf{minimize}} : ||\mathbf{D}^{\ell}\omega|| + \eta \sum \omega_j ||D_j^0||_2 \\
&\quad \textbf{subject to: } \forall n < \ell, \sum \omega_j D_j^n = 0 \\
&\}
\end{aligned} \tag{32}$$

with $\eta \geq 0$. By varying η, we encourage the optimizer to include better controls in addition to minimizing the diamon norm. The absolute value of η depends on the particular numerical values in $\mathbf{D}^{\ell}$.

3. *Sparsity Constraints*

As a practical consideration, we would also like to regularize our objective function to penalize for non-sparse weightings. Control electronics often have a limited amount of waveform memory, and thus it is important that MQCs only require a small number of controls. As an example, consider 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 a 0MQC, ignore H_4. However, if we additionally want our controls to form a 1MQC, 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 other controls' derivatives. Thus we would like to be frugal in which controls we select. In many machine learning contexts, lasso regularization [18] can be used to enforce sparsity in solutions, however this is insufficient for our purposes, 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 [19] and can be expressed via another convex program that extends Equation 23:

$$\begin{aligned}
&\textbf{minimize}\{\\
&\quad \underset{n\in[N]}{\textbf{minimize}} : ||\mathbf{D}^{\ell}\omega|| + t \\
&\quad \underset{\omega_j \geq 0, |\omega|_1 = 1, \\ t \geq 0}{\textbf{minimize}} : ||\mathbf{D}^{\ell}\omega|| + t \\
&\quad \textbf{subject to: } \omega_n > \frac{\lambda}{t} \\
&\qquad\qquad \forall n < \ell, \sum \omega_j D_j^n = 0 \\
&\}
\end{aligned} \tag{33}$$

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 0MQC and the 1MQC case in our numerical implementation in Section IV A we impose the same ℓ_2 penalty, so that the algorithm preferentially selects controls with smaller errors. Adding this constraint made the 0MQC perform better at the origin by nearly an order of magnitude, and moved the 1MQC from being out-performed by the 0MQC 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 at the origin. This shows that through adding constraints to our optimization routine, we can make the MQC practically useful.

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 [20]. To implement an MQC 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 MQC, we then performed six randomized benchmarking experiments[21]: 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

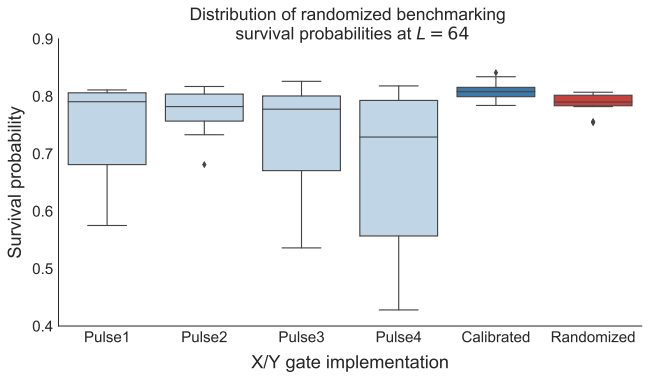

FIG. 3. Randomized benchmarking experiments ran using different pulse definitions. The first four boxes result from using each of four different implementations of the $\pi/2$ rotations. The coherent noise present in these implementations leads to large variance of the survival probability over sequences. 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 MQC 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 MQC is a tell-tale sign of reduced coherent error in the effective channel.

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 MQC, 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 [22]. 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 MQC better than the constituent gates, it also has a signicantly less-coherent error channel.

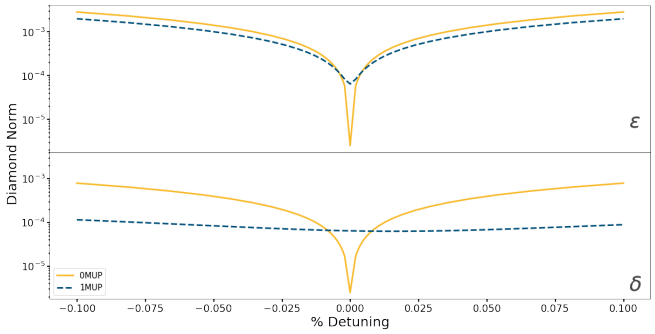

FIG. 4. Numerical results comparing a 0MQC to a 1MQC for a single tunable qubit, for $RY(\frac{\pi}{2})$. The results are qualitatively similar to those for $RX(\frac{\pi}{2})$. In this case the 0MQC outperforms both the 1MQC by two orders of magnitude, and the constituent controls by three orders of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

B. Numerical Implementation

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

$$H(\delta, \epsilon, t) = \epsilon\sigma_z + (1+\delta)(c_x(t)\sigma_x + c_y(t)\sigma_y) \tag{34}$$

We use the GRAPE algorithm[23] 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 signal. Solving the optimization problem defined in Section **??** yields similar MQCs 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 MQCs 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 [24]:

$$\begin{aligned} H(\vec{\delta}, \vec{\epsilon}, t) = &\sum_{j=1}^{2} (\epsilon_j \sigma_z^j + (1+\delta_j)(c_x^j x(t)\sigma_x^j + c_y^j(t)\sigma_y^j)) \\ &+ \frac{1}{10}(XX + YY) \end{aligned} \tag{35}$$

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 [25], consisting of all combinations of simultaneous π pulses activated at multiples of 8 steps from the

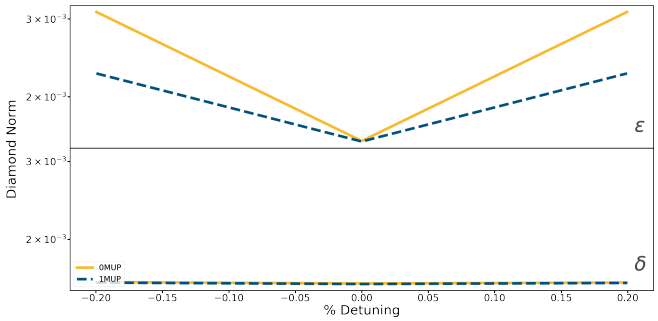

FIG. 5. Numerical results comparing a 0MQC to a 1MQC for a pair of tunable qubits, with a resonant exchange interaction. Shown with lower alpha values are example constituent controls. The 0MQC and 1MQC can be seen to outperform these controls by half of an order of magnitude at the origin. For all detuning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

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 an MQC 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 1MQC performs as well or better than 0MQC.

V. CONCLUSION AND FUTURE WORK

We have shown numerically that using MQCs 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 [26] and GOAT[27], and using more sophisticated benchmarking routines such as GST[28] 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 [29–31] 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] S. S. Hong, A. T. Papageorge, P. Sivarajah, G. Crossman, N. Dider, A. M. Polloreno, E. A. Sete, S. W. Turkowski, M. P. da Silva, and B. R. Johnson, "Demonstration of a parametrically-activated entangling gate protected from flux noise," (2019), arXiv:1901.08035.

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

[5] S. J. Beale, J. J. Wallman, M. Gutiérrez, K. R. Brown, and R. Laflamme, Physical Review Letters **121** (2018),

10.1103/physrevlett.121.190501.
[6] E. Campbell, Physical Review A **95** (2017), 10.1103/physreva.95.042306.
[7] M. B. Hastings, "Turning gate synthesis errors into incoherent errors," (2016), arXiv:1612.01011.
[8] E. Campbell, "A random compiler for fast hamiltonian simulation," (2018), arXiv:1811.08017.
[9] 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.
[10] K. Kraus, A. Böhm, J. D. Dollard, and W. H. Wootters, eds., *States, Effects, and Operations Fundamental Notions of Quantum Theory* (Springer Berlin Heidelberg, 1983).
[11] M.-D. Choi, Linear Algebra and its Applications **10**, 285 (1975).
[12] K. Życzkowski and I. Bengtsson, Open Systems & Information Dynamics (OSID) **11**, 3 (2004).
[13] J. L. O'Brien, G. J. Pryde, A. Gilchrist, D. F. V. James, N. K. Langford, T. C. Ralph, and A. G. White, Physical Review Letters **93** (2004), 10.1103/physrevlett.93.080502.
[14] J. M. Chow, J. M. Gambetta, A. D. Córcoles, S. T. Merkel, J. A. Smolin, C. Rigetti, S. Poletto, G. A. Keefe, M. B. Rothwell, J. R. Rozen, M. B. Ketchen, and M. Steffen, Physical Review Letters **109** (2012), 10.1103/physrevlett.109.060501.
[15] S. Wright and J. Nocedal, Springer Science **35**, 7 (1999).
[16] L. Khachiyan, Soviet Mathematics Doklady **20** (1979).
[17] E. Magesan, D. Puzzuoli, C. E. Granade, and D. G. Cory, Physical Review A **87** (2013), 10.1103/physreva.87.012324.
[18] R. Tibshirani, Journal of the Royal Statistical Society. Series B (Methodological) , 267 (1996).
[19] 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.
[20] 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.
[21] E. Magesan, J. M. Gambetta, and J. Emerson, Physical Review Letters **106** (2011), 10.1103/physrevlett.106.180504.
[22] H. Ball, T. M. Stace, S. T. Flammia, and M. J. Biercuk, Physical Review A **93** (2016), 10.1103/physreva.93.022303.
[23] N. Khaneja, T. Reiss, C. Kehlet, T. Schulte-Herbrüggen, and S. J. Glaser, Journal of Magnetic Resonance **172**, 296 (2005).
[24] 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.
[25] L. Viola and S. Lloyd, Physical Review A **58**, 2733 (1998).
[26] T. Caneva, T. Calarco, and S. Montangero, Physical Review A **84** (2011), 10.1103/physreva.84.022326.
[27] S. Machnes, E. Assémat, D. Tannor, and F. K. Wilhelm, Physical Review Letters **120** (2018), 10.1103/physrevlett.120.150401.
[28] R. Blume-Kohout, J. K. Gamble, E. Nielsen, K. Rudinger, J. Mizrahi, K. Fortier, and P. Maunz, Nature Communications **8** (2017), 10.1038/ncomms14485.
[29] R.-B. Wu, B. Chu, D. H. Owens, and H. Rabitz, Physical Review A **97** (2018), 10.1103/physreva.97.042122.
[30] 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.
[31] C. Ferrie and O. Moussa, Physical Review A **91** (2015), 10.1103/physreva.91.052306.

VII. APPENDIX

A. Diamond distance inequality

Here we prove the claim of (13) that:

$$\epsilon_\diamond(\mathsf{E}_{\text{eff}}) \leq \sum_i w_i\, \epsilon_\diamond(\mathsf{E}_i). \tag{36}$$

The effective error channel for a mixed quantum gate is $\mathsf{E}_{\text{eff}} = \sum_i w_i\, \mathsf{E}_i$, where E_i are the error channels for the component gates. The diamond distance to the identity of the effective error channel is:

$$\epsilon_\diamond\left(\mathsf{E}_{\text{eff}}\right) = \frac{1}{2} \sup_\rho ||(\mathsf{I}_d \otimes \mathsf{I}_d)(\rho) - (\mathsf{E}_{\text{eff}} \otimes \mathsf{I}_d)(\rho)||_1 \tag{37}$$

$$= \frac{1}{2} \sup_\rho ||\sum_i w_i \left((\mathsf{I}_d - \mathsf{E}_i) \otimes \mathsf{I}_d\right)(\rho)||_1 \tag{38}$$

For qubits, the space of density matrices is compact, so the supremum is achievable. Call a state that achieves the supremum ρ^*. Then

$$\epsilon_\diamond\left(\mathsf{E}_{\text{eff}}\right) = \frac{1}{2}||\sum_i w_i \left((\mathsf{I}_d - \mathsf{E}_i) \otimes \mathsf{I}_d\right)(\rho^*)||_1 \tag{39}$$

$$= \frac{1}{2}||\sum_i w_i\, \rho_i^*||_1, \tag{40}$$

where we have defined $\rho_i^* = ((\mathsf{I}_d - \mathsf{E}_i) \otimes \mathsf{I}_d)(\rho^*)$. The nuclear norm above is equal to the sum of the singular values of $\sum_i w_i \rho_i$. Using the Ky Fan singular value inequality [K. Fan, Maximum properties and inequalities for the eigenvalues of completely continuous operators, Proc. Nat. Acad. Sci. U.S.A. 37 (1951), 760–766] , we have

$$\epsilon_\diamond\left(\mathsf{E}_{\text{eff}}\right) \leq \frac{1}{2} \sum_i w_i ||\rho_i^*||_1 \tag{41}$$

$$\leq \sum_i w_i\, \epsilon_\diamond(\mathsf{E}_i) \tag{42}$$

The second inequality above follows because ρ^* defines an explicit lower bound for the diamond distance for each of the component error maps.