

# Robustly decorrelating errors with mixed quantum gates

Anthony M. Polloreno\*  
Rigetti Computing, Berkeley, CA

Kevin C. Young  
Quantum Performance Laboratory, Sandia National Laboratories, Livermore, CA  
(Dated: August 1, 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 [1] and Campbell [2] 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 work begins by reformulating the results of Hastings and Campbell as a quantum optimal control problem. We then discuss a family of convex programs designed to improve the performance, implementability, and robustness of the resulting mixed quantum gates. Finally, we implement these mixed quantum gates on a superconducting qubit and discuss randomized benchmarking results consistent with a marked reduction in the coherent error.

[1] M. B. Hastings, *Quantum Information & Computation* 17, 488 (2017).

[2] E. Campbell, *Physical Review A* 95, 042306 (2017).

## I. INTRODUCTION

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 approximate gate compilations that induce repeatable, undesired unitary errors on the state of a quantum information processor (QIP). Errors of this type are correlated in time and may add up constructively or destructively, depending on the circuit. They are computationally expensive to model and, because coherent errors may amplify when gates are repeated, it can be difficult to place tight analytic bounds on circuit performance [3]. 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 accurately 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 [3].

Recent works by Campbell [2, 4] and Hastings [1] have shown that coherent noise can be strongly suppressed by probabilistically mixing several distinct implementations of the target quantum gates. They focus on errors arising from various gate compilations, such as the Solovay-Kitaev algorithm, for which any lingering approximation errors are generally coherent, even if the native gates are perfect. Different approximate gate compilations of the same target unitary will almost certainly result in a different unitary error. So by selecting from these compilations at random, the resulting quantum channel becomes

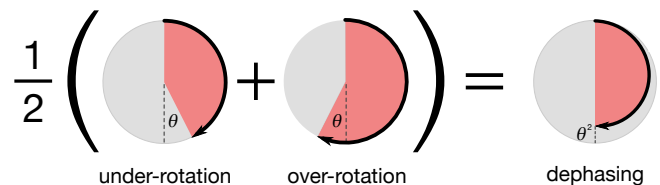


FIG. 1. An example of a mixed quantum gate. Using optimal control, two implementations of a  $Z_\pi$  gate are designed to have equal and opposite sensitivity to errors (if one implementation over-rotates by angle  $\theta$ , then the other *under*-rotates by  $\theta$ ). Each time the gate is used, 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  $\mathcal{O}(\theta^2)$ .

a mixture of unitaries [5], which has significantly less coherent error than any single compilation on its own. A simple example of this reduction in coherent error by mixed quantum gates is illustrated in Fig. 1.

In this article, we extend the work of Campbell and Hastings to numerically optimized quantum gates, and show that the advantages of this approach can be made robust to drift in the gate implementations. We demonstrate that, depending on what properties of the resulting channel are desired, different numerical optimization targets 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 [6]. 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

\* Email: [ampolloreno@gmail.com](mailto:ampolloreno@gmail.com)

unitary controls which are robust to drift and uncertainty in the control parameters.

## 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, *i.e.*, the qubits experience errors. If the device is fairly stable with time and context[7], then we can accurately model the gate action using a completely positive, trace-preserving (CPTP) map,  $\tilde{G}$ . This map can always be written as  $\tilde{G} = E \circ G$ , where  $E = \tilde{G} \circ G^{-1}$  is the *error map*, which is itself CPTP because  $G$  is unitary.

CPTP maps possess a number of useful representations, including Kraus operators[8], Choi matrices[9], and Jamiolkowski states[10]. But for the purposes of this article, the *process matrix* representation will be particularly convenient[11], 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 are composable and act through the usual matrix multiplication on the vectorized quantum state:

$$\begin{aligned} \text{vec}(\tilde{G}(\rho)) &= \tilde{G} \cdot \text{vec}(\rho) \\ &= \mathcal{E} \cdot \mathcal{G} \cdot \text{vec}(\rho) \end{aligned} \quad (1)$$

We shall work in the basis of 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,

$$\text{vec}(\rho)_i = \text{Tr}(\rho \cdot \Sigma_i) / 2^n, \quad (3)$$

and

$$(\tilde{G})_{ij} = \text{Tr}(\tilde{G}(\Sigma_j) \cdot \Sigma_i) / 2^n. \quad (4)$$

Process matrices written in the Pauli basis are often referred to as *Pauli transfer matrices* [12]. Error maps represented this way take the particularly nice form:

$$\mathcal{E} = \left( \begin{array}{c|c} 1 & \vec{0}^T \\ \hline \vec{m} & R \end{array} \right) \quad (5)$$

The top row of all trace-preserving (TP) error maps is fixed to  $\{1, 0, 0, 0, \dots\}$  and the remainder of the first column,  $\vec{m}$ , describes any deviations from unitality, as might

arise from amplitude damping [13]. If the error map is unitary, then the error is called *coherent*, and the unitary 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. 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.

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

$$\mathcal{E} = \exp(\mathcal{L}) \quad (6)$$

$$= \mathcal{I}_d + \mathcal{L} + \frac{1}{2}\mathcal{L}^2 + \mathcal{O}(\mathcal{L}^3). \quad (7)$$

If an implemented gate is relatively close to the target, then the error generator will be small under any of the usual matrix norms, and the Taylor expansion above may reliably be truncated at first or second order.

### B. Mixed Quantum Gates

Now suppose that we have access to an ensemble of different implementations of the target gate,  $\{\tilde{G}_i\}$ . 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  $\tilde{G}_i$  is  $w_i$  (and we ensure that  $\sum_i w_i = 1$ ). This procedure is operationally indistinguishable from always applying the effective operation,

$$\tilde{G}_{\text{eff}} = \sum_i w_i \tilde{G}_i. \quad (8)$$

We call such randomized quantum operations *mixed quantum gates* or MQGs. Error metrics for MQGs can be computed in terms of their effective error map,

$$E_{\text{eff}} = \sum_i w_i E_i. \quad (9)$$

Two important error metrics are the average gate infidelity (AGI),  $\epsilon_{\mathcal{F}}$ , and the diamond distance to the target,  $\epsilon_{\diamond}$  [14], defined as:

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

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

where  $d = 2^n$  and  $\mathbb{I}_d$  is the  $d$ -dimensional identity operator. In 10, we have written the AGI for an error map,  $E$ , in terms of its associated Pauli transfer matrix,  $\mathcal{E}$ . If the error channel is purely stochastic, then  $\epsilon_{\diamond}(E) = \epsilon_{\mathcal{F}}(E)$ , but if the error channel has a unitary component, then

the diamond distance will generically be larger than the average gate infidelity [15]. The diamond distance is sub-additive [14] under gate composition, so is particularly useful for constructing bounds on quantum circuit performance: the total variation distance between the outcome probabilities of a faulty and perfect quantum circuit is less than or equal to the sum of the diamond distances for the component gates [16].

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

$$\epsilon_{\mathcal{F}}(\mathbf{E}_{\text{eff}}) = \sum_i w_i \epsilon_{\mathcal{F}}(\mathbf{E}_i). \quad (12)$$

That is, the AGI of the MQG is simply the weighted average of the component AGIs, so there is no 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):

$$\epsilon_{\diamond}(\mathbf{E}_{\text{eff}}) \leq \sum_i w_i \epsilon_{\diamond}(\mathbf{E}_i). \quad (13)$$

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.

Campbell [2] and Hastings [17] independently considered this problem using gates constructed with the Solovay-Kitaev algorithm, for which many approximate gate compilations are possible. Campbell showed that, if the error generators of some ensemble of gate compilations form a convex set containing the origin, then one can construct a MQG with quadratically suppressed diamond distance to the target. Explicitly, the weights are chosen such that the error generator is canceled to first order. Using (9) and (6), the effective error map for a MQG in terms of the component error generators is:

$$\mathcal{E}_{\text{eff}} \simeq \left( \mathcal{I}_d + \sum_i w_i \mathcal{L}_i + \frac{1}{2} \sum_i w_i \mathcal{L}_i^2 \right) \quad (14)$$

If, as Campbell required, the origin is in the convex hull of the generators (see Fig. 2), then there exists a choice for the weights,  $w_i^*$ , such that  $\sum_i w_i^* = 1$  and:

$$\left\| \sum_i w_i^* \mathcal{L}_i \right\| = 0 \quad (15)$$

where  $\|\cdot\|$  is meant to mean the Frobenius norm.

If the error rates, as measured by the diamond distance, of the component gates are bounded by  $\alpha$ , then Campbell shows that this first-order cancellation of the error generators can ensure that the error rate of the MQG is bounded by  $\alpha^2$ . Campbell considered error channels that were strictly unitary, so the error generators in 15 were Hamiltonians. Hamiltonian generators can have positive or negative coefficients, so it is possible that the origin may lie in their convex hull. If, however, the error map contains stochastic components, then the condition

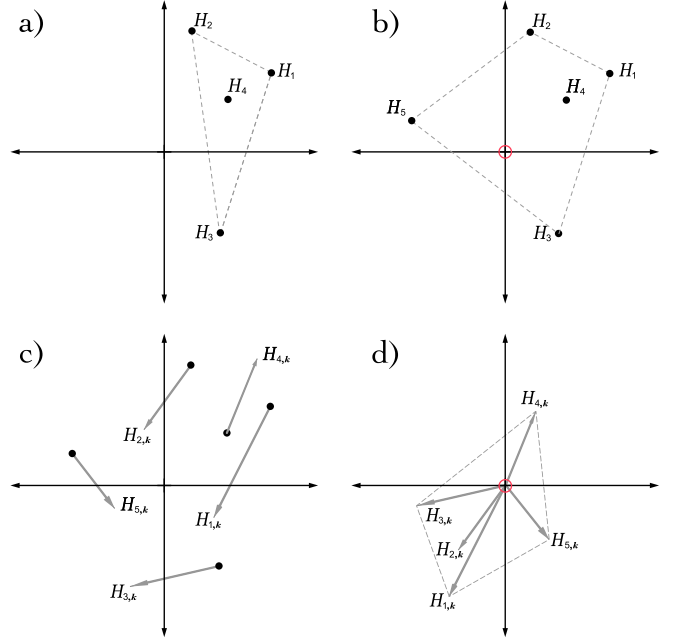


FIG. 2. A target unitary gate can be implemented a number of ways, each with a different effective Hamiltonian error. These error Hamiltonians lie in a vector space. a) Four effective Hamiltonians. The origin is not contained in their convex hull, so there are no 0MQGs. b) The origin is contained in the convex hull after adding an additional control solution. Because there are more than  $n+1$  implementations ( $n=3$ ), there exist an infinite number of 0MQGs. c) The error Hamiltonians shown with their derivative with respect to a control parameter. As this parameter drifts, a 0MQG 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 1MQG.

of 15 might be impossible to satisfy. Such errors always have positive probabilities, so the origin can *never* lie in their convex hull. This can be rectified by restricting the sum in 15 to only the Hamiltonian component of the generators. In the rest of this paper, we will be considering gates that have only unitary errors, so we will neglect this complication going forward.

While Campbell's approach can provide a guaranteed suppression of the error rate, it does not necessarily lead to a purely stochastic channel. Second- and higher-order terms in (14) can easily contribute to lingering coherent errors that impact the efficient simulability of the channel. In order to definitively eliminate these coherent errors, we seek weights,  $w_i$ , that annihilate the off-diagonal entries of the unitary submatrix of the effective error map, resulting in an effective error:

$$\mathcal{E}_{\text{eff}} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & p_x & 0 & 0 \\ 0 & 0 & p_y & 0 \\ 0 & 0 & 0 & p_z \end{pmatrix}. \quad (16)$$

The resulting mixed quantum gate is then guaranteed to have a Pauli stochastic error channel, and significantly increasing its simulability. We can formulate this problem analogously to the generator approach. The off-diagonal elements of the error map form a vector space, so we need only check that the origin is in the convex hull of the vectors of off-diagonal elements for each of the component operations,  $\mathcal{E}_i$ .

In the remainder of this paper, we will formalize this process, presenting explicit convex optimizations to both generate a family of unitary controls and construct the optimal weighting vector. We will discuss how one might include additional desiderata, including a reduced sensitivity to environmental fluctuation as well as limiting the number of native unitaries that participate in the mixed gate.

### III. CONSTRUCTING MIXED QUANTUM GATES

In this section, we present a methodology for producing useful mixed quantum channels, formalizing the intuitive approach discussed above. As mentioned, our method requires two steps. The first is a control synthesis step, in which we construct an ensemble of gate implementations. Campbell and Hastings draw their ensembles from various different gate compilations, but for this work we use quantum optimal control theory to produce these ensembles. By running simple gradient descent algorithms, such as GRAPE [18], with different random initial seeds, one can quickly construct large ensembles of approximate quantum gates. We discuss this approach in some detail in Sec. IV B. We note that advanced quantum control protocols may provide a more principled approach to control synthesis. DMORPH [19], for example, allows one to explore families of controls on fidelity level sets.

The remainder of this section will focus on the second step of constructing the optimal weight vector to reduce the coherence of the effective error channel. We will begin by discussing two classes of optimization targets, i) cancellation of the first-order generator, and ii) elimination of the off-diagonal elements of the error channel. We then discuss a set of additions to the numerical optimization that can improve the performance by i) adding robustness to drift, ii) targeting low-error rate solutions, and iii) reducing the number of native unitaries that comprise the effective channel.

#### A. Optimization Targets

##### 1. First-order generators

If our primary goal is for our MQG to have better worst-case performance, then a useful optimization target is minimizing the diamond norm. The diamond norm

is a non-linear function that in general requires a convex optimizer to compute [14]. 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 [2], a sufficient condition to minimize the diamond norm of an MQG with error generators  $\{\mathcal{L}_i\}$  to first order is:

$$\|\sum \omega_i \mathcal{L}_i\| = 0 \quad (17)$$

In [2] Campbell presents an algorithm that, given an oracle to approximate arbitrary unitary operations, generates controls and weightings to find an MQG with this property. Alternatively, one might consider the following problem: *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 are given a collection of precomputed gate implementations. Given such an MQC we can consider the  $n^2 \times m$  matrix whose columns are the vectorized error generators:

$$\mathbf{L} = \begin{pmatrix} \mathcal{L}_{1,1} & \mathcal{L}_{2,1} & \cdots \\ \vdots & \ddots & \\ \mathcal{L}_{1,n} & & \mathcal{L}_{m,n} \end{pmatrix} \quad (18)$$

To satisfy (17), we can write the sum as a matrix product and solve the following optimization problem, which will be zero if and only if the sum is zero:

$$\underset{\omega_i \geq 0, \|\omega\|_1 = 1}{\text{minimize}} : \|\mathbf{L}\omega\|_1 \quad (19)$$

where  $\|\cdot\|_1$  is the  $\ell_1$  norm.

As denoted, this optimization needs to be constrained so that  $\omega$  is a proper probability distribution. Nevertheless, this forms a convex optimization problem, and such linearly constrained minimization problems with quadratic cost functions have been proven to be efficiently solvable by methods like the Ellipsoid Method [20, 21]. More importantly, there are many existing convex solver software packages that solve these problems efficiently in practice.

##### 2. Off-diagonals

There are other properties of a channel that one might consider optimizing. For instance, if an MQG has an error channel that is well-approximated by a Pauli-Stochastic channel, then it may be amenable to classical simulation. We can modify our optimization problem accordingly. By vectorizing the  $R$  submatrices in (5), and considering the matrix of off-diagonal terms, we can again consider the minimization problem in (19), where we replace  $\mathbf{L}$  with  $\mathbf{R}$ :

$$\mathbf{R} = \begin{pmatrix} R_{12,1} & R_{22,1} & \cdots \\ \vdots & \ddots & \\ R_{1_{n-1},n-1} & R_{m_{n-1},n-1} \end{pmatrix} \quad (20)$$

In this case, if (19) is satisfied, the resulting error channel will be *exactly* a Pauli Channel. Previous authors have considered finding the closest Pauli or Clifford Channel to a given process [22], but this does not have the restriction that the resulting channel be decomposable into a given family of controls.

## 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 MQG will degrade. Thus, we would like to design MQGs that are robust to this drift. To enforce robustness, we can consider the generators in (17) to be functions of a control vector  $\vec{\delta}$ , and instead of 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. Specifically, we are interested in the  $n^{th}$  order derivatives:

$$D_i^n = \frac{1}{n!} \frac{\partial^n}{\partial \delta_{j_1} \dots \partial \delta_{j_n}} \mathcal{L}_i(\vec{\delta})|_{\vec{\delta}=\vec{0}} \quad (21)$$

If the dimension of  $\vec{\delta}$  is  $d$ , the indices  $j_0, \dots, j_n$  take on values in  $1, \dots, d$  and this matrix has  $d^n$  entries. We say that an MQG is robust to order  $\ell$  (an  $\ell$ MQG) if for all  $1 \leq k \leq \ell$ :

$$\left\| \sum_i \omega_i \left( \sum_{n=0}^k D_i^n \right)^n \right\| = 0 \quad (22)$$

In particular, we see that a 0MQG satisfies (17). More generally, these conditions imply that an  $\ell$ MQG is insensitive to  $\ell^{th}$  order in drift in  $\vec{\delta}$ . To see this, we can rewrite the implementations of the target gate as:

$$\begin{aligned} \tilde{\mathcal{G}}_i(d\vec{\delta}) &= \exp(-i(\mathcal{L}_i(\vec{0}) + \frac{\partial}{\partial \delta_j} \mathcal{L}_i(\vec{0}) d\delta_j \\ &\quad + \frac{1}{2} \frac{\partial^2}{\partial \delta_j \partial \delta_k} \mathcal{L}_i(\vec{0}) d\delta_j d\delta_k + \dots)) \mathcal{G} \end{aligned} \quad (23)$$

By Taylor expanding (23) in  $d\vec{\delta}$ , one can see that the the first  $\ell$  derivatives of an  $\ell$ MQG will be zero. Furthermore if we can find controls such that  $\|D_i^n\| \leq \epsilon$ , we can approximate (22) as:

$$\left\| \sum_i \omega_i D_i^n \right\| + \mathcal{O}(\epsilon^2) = 0 \quad (24)$$

This condition guarantees that errors will be suppressed to second order in  $\epsilon$  for all derivatives up to order  $\ell$ . As discussed in [2], the geometric picture for 0MQGs is to consider the error generators as vectors, and to ensure zero is contained in their convex hull. Figure 2 generalizes this picture to 1MQGs and gives intuition for the conditions required to produce them. To generate  $\ell$ MQGs, we first define the vectorized derivative matrix  $\mathbf{D}^\ell$  in a similar way to (18):

$$\mathbf{D}^\ell = \begin{pmatrix} D_{1,1}^\ell & D_{2,1}^\ell & \cdots \\ \vdots & \ddots & \\ D_{1,n}^\ell & & D_{m,n}^\ell \end{pmatrix} \quad (25)$$

Using this, we can then solve the following convex optimization problem, generalizing (19):

$$\begin{aligned} \text{minimize : } & \|\mathbf{D}^\ell \omega\|_1 \\ & \omega_i \geq 0, |\omega|_1 = 1 \\ \text{subject to: } & \forall n < \ell, \left\| \sum \omega_i D_i^n \right\| = 0 \end{aligned} \quad (26)$$

with  $D_i^n$  defined in (21).

### 2. Hamiltonian Norm Regularization

The convex optimization problems discussed up until this point can be solved analytically, however casting them as convex optimization problems will allow us to add penalty terms to the cost function to promote different properties of the MQG. In particular, while (26) is sufficient for suppressing 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. As an example, any MQG that mixes equally over equal and opposite errors will satisfy (26), but the magnitude of the error of the resulting channel will depend on the magnitude of the original coherent errors. 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  $\ell_2$  ( $\|\cdot\|_2$ ) norm:

$$\begin{aligned} \text{minimize : } & \|\mathbf{D}^\ell \omega\|_1 + \eta \sum \omega_i \|D_i^0\|_2 \\ & \omega_i \geq 0, |\omega|_1 = 1 \\ \text{subject to: } & \forall n < \ell, \left\| \sum \omega_i D_i^n \right\| = 0 \end{aligned} \quad (27)$$

with  $\eta \geq 0$ . By increasing  $\eta$ , we can encourage the optimizer to include better controls while minimizing the diamond norm. The value of  $\eta$  that is best for any particular problem instance will depend on the values in  $\mathbf{D}^\ell$ .

### 3. Sparsity Constraints

As a practical consideration, we would also like to regularize our objective function to promote sparse weightings. Control electronics often have a limited amount of



waveform memory, and thus it is important that MQGs only require a small number of controls. As an example, consider Figure 2. In b), it is clear that  $H_4$  is unnecessary to contain the origin in the convex hull of the error generators. Thus we would prefer that a 0MQG exclude  $H_4$ . However, if we additionally want our controls to form a 1MQG, we see from 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.

In many machine learning contexts, lasso regularization [23] can be used to enforce sparsity in solutions, however this is insufficient for our purposes, as we already constrain  $\omega$  to be a valid probability distribution. The problem of enforcing sparsity in such situations has been considered in [24] and can be expressed via another convex program that extends (26):

$$\begin{aligned}
& \text{minimize} \\
& \quad n \in [N] \\
& \text{minimize : } \|\mathbf{D}^\ell \omega\|_1 + t \\
& \quad \omega_i \geq 0, \|\omega\|_1 = 1, \\
& \quad t \geq 0 \\
& \text{subject to: } \omega_n > \frac{\lambda}{t} \\
& \quad \forall n < \ell, \|\sum \omega_i D_i^n\| = 0
\end{aligned} \tag{28}$$

with  $\lambda \geq 0$ . As with  $\eta$  in the last subsection, the optimal value of  $\lambda$  is problem specific, depending on  $\mathbf{D}^\ell$  and how sparse the solution needs to be.

For small  $\lambda$ , the sparsity penalty is negligible, and for large  $\lambda$  we see that if  $\omega$  is not sparse then  $t$  must be very large, which increases the cost of that particular solution. In the following sections we discuss both experimental and numerical implementations of MQGs, leveraging these optimization techniques.

## IV. RESULTS

### A. Experimental

Here we present experimental results from implementing this routine on a fixed-frequency, superconducting transmon qubit. In particular, we use qubit 8 on the Rigetti 19Q-Acorn chip, whose characterization can be found in [25]. To implement an MQG on this qubit, four miscalibrated Gaussian pulses were produced by scaling the 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 Section III A 2, we chose to minimize the off diagonal elements of the process matrix. To benchmark the quality of the MQG, we then performed six randomized benchmarking experiments[26]: 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 and 10 sequences per sequence

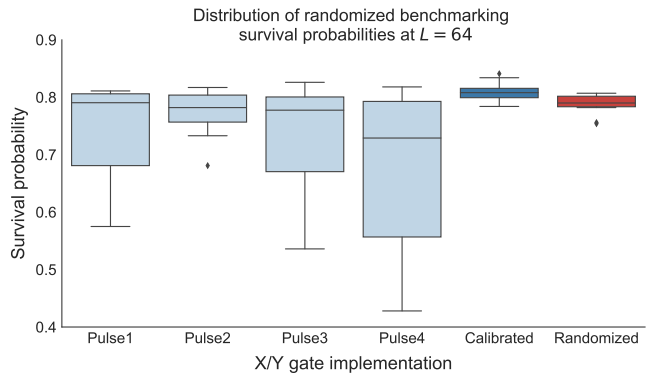


FIG. 3. Randomized benchmarking experiments ran using different pulse definitions. The first four boxes result from using each of four different implementations of the  $\pi/2$  rotations. The coherent noise present in these implementations leads to large variance of the survival probability over sequences.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

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  $RY(\frac{\pi}{2})$ . 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 MQG, demonstrating that it performs almost as well as the calibrated pulse, and better than the constituent pulses.

More importantly, minimizing the off-diagonal elements of the process matrix, we expect to produce a process with suppressed coherent error. To see that this is the case, we can observe the drastic reduction in the variance at each point for the MQG relative to the miscalibrated pulses. As discussed in [6], 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 the MQG has a significantly less-coherent error channel.

### B. Numerical Implementation

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

$$H(\delta, \epsilon, t) = \epsilon \sigma_z + (1 + \delta)(c_x(t)\sigma_x + c_y(t)\sigma_y) \quad (29)$$

where  $\epsilon$  is a random variable used to model fluctuations in the qubit frequency, and  $\delta$  is a random variable used to model fluctuations in the control field. We use the GRAPE algorithm[18] with  $N=25$  steps and total evolution time of  $\pi$  to generate 100 candidate controls.

In our implementation, we considered the performance function presented in [18], and averaged over different values of  $\delta$  and  $\epsilon$  using Gaussian quadrature when computing the gradient, so that we find controls that perform well in a neighborhood. The standard deviations considered for all random variables in our numerical experiments were fixed to  $\sigma = .001$ . Finally, we assume that the errors on  $\sigma_x$  and  $\sigma_y$  are perfectly correlated, as in our experimental implementation.

Using controls generated in this way, the MQGs produced for  $RX(\frac{\pi}{2})$  and  $RY(\frac{\pi}{2})$  are qualitatively similar, with the results for  $RY(\frac{\pi}{2})$  shown in Figure 4. By imposing the  $\ell_2$  penalty from Section III B 2, so that the algorithm preferentially selects controls with smaller errors. Adding this constraint made the 0MQG perform better at the origin by nearly an order of magnitude, and moved the 1MQG from being out-performed by the 0MQG 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 MQG practically useful.

In our two-qubit example we consider the following model for two tunable qubits coupled by a resonant exchange interaction, similar to that in [27]:

$$H(\vec{\delta}, \vec{\epsilon}, t) = \sum_{j=1}^2 (\epsilon_j \sigma_z^j + (1 + \delta_j)(c_x^j(t)\sigma_x^j + c_y^j(t)\sigma_y^j)) + \frac{1}{10}(XX + YY) \quad (30)$$

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 [28], consisting of all combinations of simultaneous  $\pi$  pulses activated at multiples of 8 steps from the

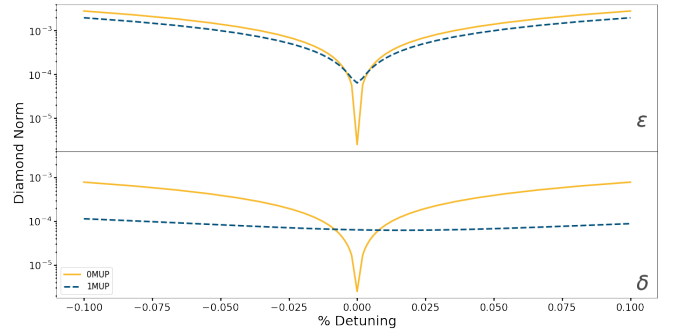


FIG. 4. Numerical results comparing a 0MQG to a 1MQG for a single tunable qubit, for  $RY(\frac{\pi}{2})$ . The results are qualitatively similar to those for  $RX(\frac{\pi}{2})$ . In this case the 0MQG outperforms both the 1MQG by two orders of magnitude, and the constituent controls by three orders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

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 also added uniformly distributed amplitude errors to each  $\pi$  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 MQG improves performance by half 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 1MQG performs as well or better than 0MQG.

### V. CONCLUSION AND FUTURE WORK

We have shown numerically that using MQGs can reduce coherent error 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.

Randomized protocols have a long history of outperforming their deterministic counterparts. In [29, 30] the use of randomness is discussed in the context of dynamical decoupling. In particular, it is shown that both stochastic and hybrid approaches can result in better or more tractable coherent controls. Additionally Pauli Frame Randomization[31] has been shown [32] to reduce coherent errors by introducing randomness in a pre-compilation step. The techniques discussed in this paper are different from both of these approaches in that they explicitly produce incoherent errors through mixing, and



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

that the randomness is hardware efficient for any desired gate.

Future directions for this work include demonstrating the routine experimentally on a two-qubit gate, moving the random gate selection from a pre-compilation step to runtime logic onboard the control electronics, investigating other optimization routines such as CRAB [33] and GOAT[34], and using more sophisticated benchmarking

routines such as GST[35] 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 [36–38] to generate candidate controls, and then use an optimizer like Nelder-Mead to perform the minimization. While performing a complete optimization in this way would require full process tomography, one could instead 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

This material was funded in part by the U.S. Department of Energy, Office of Science, Office of Advanced Scientific Computing Research Quantum Testbed Program. 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] M. B. Hastings, *Quantum Information & Computation* **17**, 488 (2017).
  - [2] E. T. Campbell, B. M. Terhal, and C. Vuillot, *Nature* **549**, 172 (2017).
  - [3] S. J. Beale, J. J. Wallman, M. Gutiérrez, K. R. Brown, and R. Laflamme, *Physical Review Letters* **121** (2018), 10.1103/physrevlett.121.190501.
  - [4] E. Campbell, “A random compiler for fast hamiltonian simulation,” (2018), [arXiv:1811.08017](https://arxiv.org/abs/1811.08017).
  - [5] D. W. Leung, *CoRR* **cs.CC/0012017** (2000).
  - [6] H. Ball, T. M. Stace, S. T. Flammia, and M. J. Biercuk, *Physical Review A* **93** (2016), 10.1103/physreva.93.022303.
  - [7] K. Rudinger, T. Proctor, D. Langharst, M. Sarovar, K. Young, and R. Blume-Kohout, *Physical Review X* **9** (2019), 10.1103/physrevx.9.021045.
  - [8] 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).
  - [9] M.-D. Choi, *Linear Algebra and its Applications* **10**, 285 (1975).
  - [10] K. Życzkowski and I. Bengtsson, *Open Systems & Information Dynamics (OSID)* **11**, 3 (2004).
  - [11] 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.
  - [12] 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.
  - [13] J. Preskill, Accessible via <http://www.theory.caltech.edu/people/preskill/ph229> **2015** (1997).
  - [14] J. Watrous, *The theory of quantum information* (Cambridge University Press, 2018).
  - [15] J. J. Wallman, “Bounding experimental quantum error rates relative to fault-tolerant thresholds,” (2015), [arXiv:1511.00727](https://arxiv.org/abs/1511.00727).
  - [16] D. Aharonov, A. Kitaev, and N. Nisan, *arXiv preprint quant-ph/9806029* (1998).
  - [17] M. B. Hastings, “Turning gate synthesis errors into incoherent errors,” (2016), [arXiv:1612.01011](https://arxiv.org/abs/1612.01011).
  - [18] N. Khaneja, T. Reiss, C. Kehlet, T. Schulte-Herbrüggen, and S. J. Glaser, *Journal of Magnetic Resonance* **172**, 296 (2005).



- [19] J. Dominy and H. Rabitz, *Journal of Physics A: Mathematical and Theoretical* **41**, 205305 (2008).
- [20] S. Wright and J. Nocedal, *Springer Science* **35**, 7 (1999).
- [21] L. Khachiyan, *Soviet Mathematics Doklady* **20** (1979).
- [22] E. Magesan, D. Puzzuoli, C. E. Granade, and D. G. Cory, *Physical Review A* **87** (2013), 10.1103/physreva.87.012324.
- [23] R. Tibshirani, *Journal of the Royal Statistical Society. Series B (Methodological)*, 267 (1996).
- [24] 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.
- [25] 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](https://arxiv.org/abs/1712.05771).
- [26] E. Magesan, J. M. Gambetta, and J. Emerson, *Physical Review Letters* **106** (2011), 10.1103/physrevlett.106.180504.
- [27] 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.
- [28] L. Viola and S. Lloyd, *Physical Review A* **58**, 2733 (1998).
- [29] L. Viola and E. Knill, *Physical Review Letters* **94** (2005), 10.1103/physrevlett.94.060502.
- [30] L. F. Santos and L. Viola, *Physical Review Letters* **97** (2006), 10.1103/physrevlett.97.150501.
- [31] J. J. Wallman and J. Emerson, *Physical Review A* **94** (2016), 10.1103/physreva.94.052325.
- [32] M. Ware, G. Ribeill, D. Ristè, C. A. Ryan, B. Johnson, and M. P. da Silva, “Experimental demonstration of pauli-frame randomization on a superconducting qubit,” (2018), [arXiv:1803.01818](https://arxiv.org/abs/1803.01818).
- [33] T. Caneva, T. Calarco, and S. Montangero, *Physical Review A* **84** (2011), 10.1103/physreva.84.022326.
- [34] S. Machnes, E. Assémat, D. Tannor, and F. K. Wilhelm, *Physical Review Letters* **120** (2018), 10.1103/physrevlett.120.150401.
- [35] R. Blume-Kohout, J. K. Gamble, E. Nielsen, K. Rudinger, J. Mizrahi, K. Fortier, and P. Maunz, *Nature Communications* **8** (2017), 10.1038/ncomms14485.
- [36] R.-B. Wu, B. Chu, D. H. Owens, and H. Rabitz, *Physical Review A* **97** (2018), 10.1103/physreva.97.042122.
- [37] 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.
- [38] C. Ferrie and O. Moussa, *Physical Review A* **91** (2015), 10.1103/physreva.91.052306.
- [39] K. Fan, *Proceedings of the National Academy of Sciences of the United States of America* **37**, 760 (1951).

## VII. APPENDIX

### A. Diamond distance inequality

Here we prove the claim of (13) that:

$$\epsilon_{\diamond}(\mathbf{E}_{\text{eff}}) \leq \sum w_i \epsilon_{\diamond}(\mathbf{E}_i). \quad (31)$$

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

$$\epsilon_{\diamond}(\mathbf{E}_{\text{eff}}) = \frac{1}{2} \sup_{\rho} \|(\mathbf{I}_d \otimes \mathbf{I}_d)(\rho) - (\mathbf{E}_{\text{eff}} \otimes \mathbf{I}_d)(\rho)\|_1 \quad (32)$$

$$= \frac{1}{2} \sup_{\rho} \left\| \sum w_i ((\mathbf{I}_d - \mathbf{E}_i) \otimes \mathbf{I}_d)(\rho) \right\|_1 \quad (33)$$

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

$$\epsilon_{\diamond}(\mathbf{E}_{\text{eff}}) = \frac{1}{2} \left\| \sum w_i ((\mathbf{I}_d - \mathbf{E}_i) \otimes \mathbf{I}_d)(\rho^*) \right\|_1 \quad (34)$$

$$= \frac{1}{2} \left\| \sum w_i \rho_i^* \right\|_1, \quad (35)$$

where we have defined  $\rho_i^* = ((\mathbf{I}_d - \mathbf{E}_i) \otimes \mathbf{I}_d)(\rho^*)$ . The nuclear norm above is equal to the sum of the singular values of  $\sum w_i \rho_i$ . Using the Ky Fan singular value inequality [39], we have

$$\epsilon_{\diamond}(\mathbf{E}_{\text{eff}}) \leq \frac{1}{2} \sum_i w_i \|\rho_i^*\|_1 \quad (36)$$

$$\leq \sum_i w_i \epsilon_{\diamond}(\mathbf{E}_i) \quad (37)$$

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