Learning quantum-state feedback control with backpropagation-free stochastic optimization

Ethan N. Evans,[1,*] Ziyi Wang,[2] Adam G. Frim,[3] Michael R. DeWeese,[3,4] and Evangelos A. Theodorou[1,2]

[1]*Department of Aerospace Engineering, Georgia Institute of Technology, Atlanta, Georgia 30332, USA*

[2]*Center for Machine Learning, Georgia Institute of Technology, Atlanta, Georgia 30332, USA*

[3]*Department of Physics, University of California, Berkeley, Berkeley, California 94720, USA*

[4]*Redwood Center for Theoretical Neuroscience and Helen Wills Neuroscience Institute, University of California, Berkeley, Berkeley, California 94720, USA*

(Received 23 December 2021; revised 15 July 2022; accepted 25 August 2022; published 3 November 2022)

High-fidelity state preparation represents a fundamental challenge in the application of quantum technology. While the majority of optimal control approaches use feedback to improve the controller, the controller itself often does not incorporate explicit state dependence. Here, we present a general framework for training deep feedback networks for open quantum systems with continuous weak measurement that allows a variety of system and control structures that are prohibitive by many other techniques and can in effect react to unmodeled effects through nonlinear filtering. Our approach benefits from characteristics of both stochastic sampling and gradient-based optimization methods yet does not require differentiability as in backpropagation approaches. We demonstrate that this method is efficient due to inherent parallelizability, robust to open system interactions, and outperforms landmark state-dependent feedback control results in simulation.

DOI: 10.1103/PhysRevA.106.052405

I. INTRODUCTION

The efficacy of quantum technologies is fundamentally linked to the ability to accurately prepare, stabilize, and steer between quantum states. Examples include gate synthesis and state preparation in quantum computing [1,2], quantum metrology [3], quantum chemistry [4], nuclear magnetic resonance [5,6], and molecular physics [7]. Complex scenarios require rich tools from optimization and control theory, which often provide successful protocols with guarantees. The lens of optimal control theory and stochastic optimization provides numerous methodologies which cast such efforts as optimization problems.

The intention is to communicate that most quantum control algorithms from the optimal control theory don't incorporate explicit state dependence in the controllers [6,8–28]. These approaches produce that explicitly depend on time but are independent of the current system state. Time-dependent control may perform well in certain circumstances where there are no unmodeled interactions or other effects. However, state-dependent feedback, i.e., where the control law explicitly depends on state information, is a primary tool for guaranteeing the stability of equilibria in the classical regime.

Quantum control approaches that incorporate feedback are broken into two subcategories: Measurement-based feedback control (MFC) and coherent feedback control (CFC) [29]. In MFC, state measurements are obtained through a classical measurement system coupling, which perturbs the system through a measurement backaction, and is used for control through an ancillary system coupling [30–37]. In constrast, CFC designs a coherent coupling that controls the system state without measurement backaction [38–41]. While CFC methods may have advantages in terms of extracting system entropy [42], MFC methods can amplify measurement signals and apply macroscopic fields for feedback [43].

Within the MFC setting, measurements typically cause a severe discontinuous jump into a system eigenstate (i.e., wave-function collapse) and, as a result, are often reserved for post-experiment feedback. However, continuous weak measurement protocols reduce the discontiuous backaction to a continuous Wiener diffusion process appended to the system state evolution, as originally suggested by Belavkin [44–46]. Quantum systems with continuous weak measurements are also referred to as *quantum trajectories*, and the partially observable state measurement can be used throughout a given experiment for state-dependent feedback control. Continuous measurement schemes have gained significant traction [47–51] and enable MFC architectures that can effectively perform control on a variety of tasks [32,52,53]. Continuous MFC may yet hold the key to reducing the necessary qubit overhead in modern quantum computing architectures [54] and has the promise of improving the robustness and performance of many other quantum technologies.

In this paper, we apply optimization principles to closed-loop state-dependent feedback control in a continuous weak MFC setting. We leverage stochastic optimization and optimal control theoretic techniques, as well as tools from machine learning, to develop a general framework for learning control policies that perform feedback control for quantum state preparation and stabilization tasks. The proposed framework updates the control policy parameters through a

*Corresponding author: eevans89@gmail.com. Present address: Naval Surface Warfare Center, Panama City Division, Panama City, FL 32407, USA.

performance-weighted average of quantum trajectories, allowing us to bypass the gradient backpropagation through dynamics and performance measures. To prove its utility, the approach is applied to a two-qubit stabilization task, showing significant improvements over previous works.

II. MODELING APPROACH

Continuous weak measurement yields dynamics driven by the Belavkin equation [44–46] or, more generally, a stochastic master equation (SME) of the form:

$$d\rho_t^c = \mathcal{L}_0\rho_t^c dt + \mathcal{D}[V]\rho_t^c dt + \big(V\rho_t^c + \rho_t^c V^\dagger - \mathrm{Tr}\big[(V+V^\dagger)\rho_t^c\big]\rho_t^c\big)dW_t, \quad (1)$$

with innovation process:

$$dW_t = dy_t - \mathrm{Tr}\big[(V+V^\dagger)\rho_t^c\big]dt \quad (2)$$

where dW_t is a standard zero-mean Wiener process in the classical sense [55] and ρ_t^c is the system density state conditioned on the measurement outcome. Here, the system closure includes the system S and the measurement process R with interaction operator V. One can similarly consider a closure that includes the system S, the environment B, and the measurement process R, however, this is omitted for brevity.

Equation (1) is a quantum stochastic partial differential equation (SPDE) with quantum unconditional evolution governed by the Lindblad terms $(\mathcal{L}_0 + D[V])\rho_t^c dt$ and weak measurement conditional evolution term $(V\rho_t^c + \rho_t^c V^\dagger - \mathrm{Tr}[(V+V^\dagger)\rho_t^c]\rho_t^c)dW_t$. It is interesting to note that one can draw parallels between Eq. (1) and the Kushner-Stratonovich SPDE [30]. Just as in the case of the Kushner-Stratonovich SPDE, the stochasticity is the result of conditioning on the measurement process dy_t. Following this logic, one can think of the Belavkin equation in terms of a partially observable stochastic optimal control problem.

The open quantum system described by Eq. (1) describes an *uncontrolled* system. Control is introduced via a control Hamiltonian, which yields controlled open system dynamics given by:

$$d\rho_t^c = \mathcal{L}_0\rho_t^c dt + \mathcal{D}[V]\rho_t^c - i\sum_j \mathbf{u}_{t,j}\big[H_{u,j}, \rho_t^c\big]dt + \big(V\rho_t^c + \rho_t^c V^\dagger - \mathrm{Tr}\big[(V+V^\dagger)\rho_t^c\big]\rho_t^c\big)dW_t, \quad (3)$$

which can be equivalently expressed compactly by a simplified form:

$$d\rho_t^c = F\big(\rho_t^c\big)dt + G\big(\rho_t^c, \mathbf{u}_t\big)dt + B\big(\rho_t^c\big)dW_t, \quad (4)$$

where the term $F(\rho_t^c)dt := \mathcal{L}_0\rho_t^c dt + \mathcal{D}[V]\rho_t^c$ describes the uncontrolled drift of the dynamics, the term $G(\rho_t^c, u_t) := -i\sum_j u_{t,j}[H_{u,j}, \rho_t^c]dt$ describes the controlled drift of the dynamics, and the term $B(\rho_t^c)dW_t := (V\rho_t^c + \rho_t^c V^\dagger - \mathrm{Tr}[(V+V^\dagger)\rho_t^c]\rho_t^c)dW_t$ describes the diffusion. Note that in the closed-loop control setting, the control u has an explicit dependence on the state, $u = u(\rho_t^c)$, whereas in the time-dependent setting it may be $u = u(t)$ but does not have a direct state dependence.

A critical challenge in applying methods from stochastic optimal control is that the functional $B(\rho_t^c)$ can often be singular, leading to a degenerate diffusion process (see Appendix A for further details). Such degeneracies prove prohibitive for a variety of methods introduced in the stochastic optimal control literature, including path integral control [56–59], forward-backward stochastic differential equations using importance sampling [60,61], and, recently, spatiotemporal stochastic optimization [62,63]. In each case, such degeneracies must be carefully addressed. In this paper, we overcome them with a proposed stochastic optimization technique.

The form of the dynamics in Eq. (4) is quite general and familiar in the context of optimal control theory. From this perspective, state preparation tasks are described in terms a positive-definite performance metric or cost functional $J(\rho_t^c, u_t)$, which typically uses a distance metric to penalize deviation from a target state and may additionally seek to reduce the control effort exerted onto the system. In many cases, the cost functional may be discontinuous or nondifferentiable (e.g., in the case of barrier functions or indicator functions), which can impose difficulties on control approaches.

For concreteness, consider the task of reaching some target state ρ_{des}, as evaluated by the cost metric $J(\rho_t^c, u_t)$. The minimizing control is most generally expressed by the following path integral optimization problem:

$$\mathbf{u}^* = \underset{\mathbf{u}\in\mathcal{U}}{\mathrm{argmin}}\ \langle J(\rho^c, u)\rangle_Q, \quad (5)$$

subject to the dynamics given by Eq. (4). Here, the expectation defines a path integral over controlled state trajectories with path measure Q. The set $\mathcal{U}$ is the admissible set of controls and may impose constraints on the control; one may also include constraints on state ρ_t^c, however, these are omitted from this derivation for simplicity.

III. QUANTUM GRADIENT-BASED ADAPTIVE STOCHASTIC SEARCH FOR TRAINING FEEDBACK POLICIES

To solve this problem, we take an approach from stochastic optimization literature known as gradient-based adaptive stochastic search (GASS) [64]. The GASS approach offers generality, as well as having guarantees of convergence and rate of convergence. This approach manipulates the optimization problem by swapping the optimization variables from the control policy to the distribution parameters of the policy, thereby bypassing discontinuities and nondifferentiability in the dynamics and cost function. We provide details in Appendix B. The resulting stochastic optimization problem takes the form:

$$\boldsymbol{\theta}^* = \underset{\boldsymbol{\theta}}{\mathrm{argmax}}\ \ln\langle S(J(\rho^c, \mathbf{u}))\rangle_{Q, f(\varphi;\boldsymbol{\theta})}, \quad (6a)$$

$$\mathbf{u}_t = \boldsymbol{\Phi}\big(\rho_t^c; \varphi\big), \quad (6b)$$

$$\varphi \sim f(\varphi; \boldsymbol{\theta}), \quad (6c)$$

subject to the dynamics in Eq. (4). The subscript on the expectation denotes a double expectation with respect both to the path measure Q of the controlled system dynamics and to some distribution f belonging to the exponential

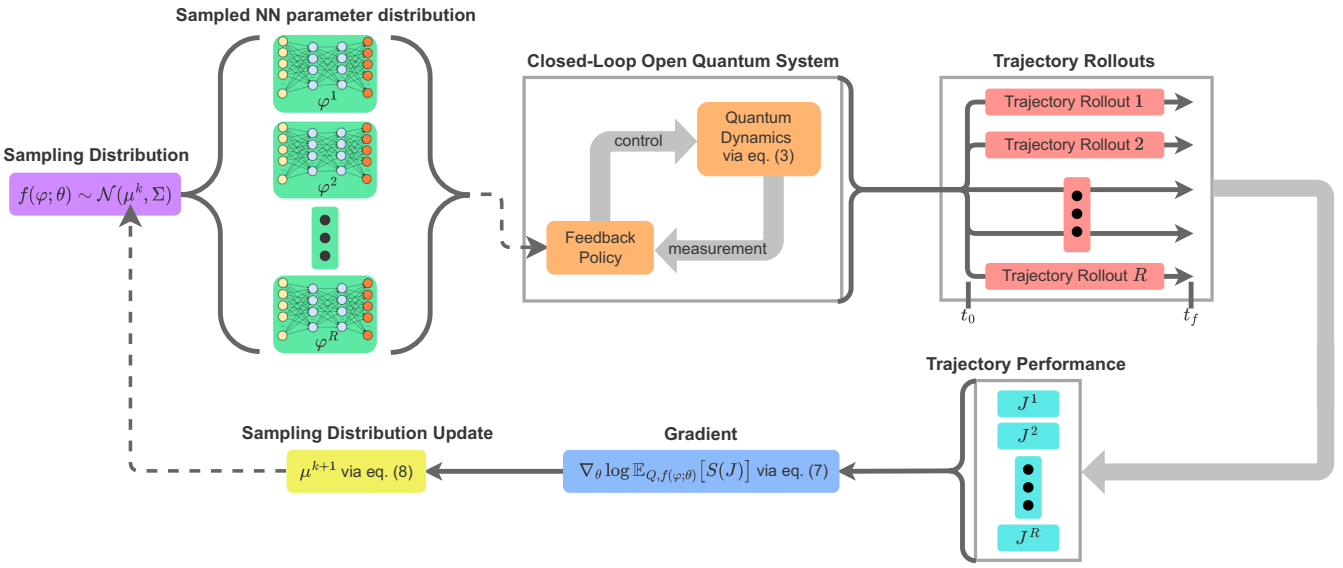

FIG. 1. Diagram of the QGASS policy-learning architecture. A distribution of policy network parameters are sampled from sampling distribution $f(\varphi;\boldsymbol{\theta})$, which are used to generate a set of trajectory rollouts from the closed-loop open-quantum system dynamics described in Eq. (3). The state output of the system is the result of a weak measurement process as described in Eq. (1). The performance of each rollout is evaluated on $J(\rho, u)$ over the trajectory, which is then used to approximate the path integral expectation in the gradient Eq. (7). The gradient is then used to generate an update to the distribution for the next iteration, and iterations continue until convergence. Network parameter samples and trajectory rollouts are performed in parallel for computational efficiency.

family of distributions. The function $S(\cdot)$ is a smooth, nonincreasing shaping function and $\boldsymbol{\Phi}$ is a neural network which takes state information and outputs a control action. Such a neural network is typically referred to as a policy network, and in this case is dependent on a set φ of weights and biases, which are sampled from distribution f with parameters $\boldsymbol{\theta}$. GASS has been applied for a variety of optimization problems [65–67] and has also been explored in the context of optimal control [68–70], however it is also appealing and pertinent in the context of policy learning as developed in this paper. We denote this quantum feedback policy-learning architecture the quantum gradient-based adaptive stochastic search (QGASS).

As the name suggests, this approach performs a gradient-based update to the parameters of the sampling distribution $f(\varphi;\boldsymbol{\theta})$, namely, the parameter gradient is obtained as:

$$\nabla_{\boldsymbol{\theta}} \ln \langle S(J(\rho^c, \mathbf{u}))\rangle_{Q,f(\varphi;\boldsymbol{\theta})} = \frac{\langle S(J(\rho^c, \mathbf{u}))(T(x) - \nabla_{\boldsymbol{\theta}} A(\boldsymbol{\theta}))\rangle_{Q,f(\varphi;\boldsymbol{\theta})}}{\langle S(J(\rho^c, \mathbf{u}))\rangle_{Q,f(\varphi;\boldsymbol{\theta})}}, \tag{7}$$

where the sampling distribution $f(\varphi;\boldsymbol{\theta})$ belongs to the exponential family of distributions with sufficient statistics $T(x)$ and log partition function $A(\boldsymbol{\theta})$. Under a Gaussian sampling distribution $f(\varphi;\boldsymbol{\theta}) \sim \mathcal{N}(\mu, \Sigma)$, with mean update and fixed variance for simplicity, the parameter update becomes:

$$\mu^{k+1} = \mu^k + \alpha^k \frac{\langle S(J(\rho^c, \mathbf{u}))(\varphi - \mu^k)\rangle_{Q,f(\varphi;\boldsymbol{\theta})}}{\langle S(J(\rho^c, \mathbf{u}))\rangle_{Q,f(\varphi;\boldsymbol{\theta})}}. \tag{8}$$

The QGASS update scheme provides a parallelizable iterative training approach, which steers a distribution of network parameters toward learning optimal values, in turn, providing a feedback control policy for the system. We include a detailed derivation of the QGASS parameter update in Appendix C. Due to the path integral nature of this approach, the QGASS algorithm is independent of discretization schemes used for simulation. Furthermore, this approach can handle discontinuous jump-diffusion dynamics, such as in the discrete measurement case [32].

IV. QGASS ALGORITHM

The QGASS framework is shown in Fig. 1 . To apply the algorithm, we first initialize the policy parameter distribution, and different policy realizations are then sampled from this distribution. For each policy sample, process noise is sampled to generate trajectory rollouts; trajectory rollout propagation is performed in parallel to significantly reduce runtime. Finally, the parameter update, Eq. (8), is performed through empirical approximation of the expectation using the cost evaluated rollouts.

The pseudocode of the QGASS algorithm is presented in Algorithm 1. The inputs to the optimization include the final time (T), the number of iterations (K), the number of network parameter rollouts (P), the number of trajectory rollouts (R), the initial state (ρ_0), shape function parameter (κ), initial network weights ($\varphi^{(0)}$), initial sampling distribution mean ($\boldsymbol{\theta}^{(0)}$), and sampling distribution variance. This algorithmic pseudocode is written with multiple layers of for loops, however, in implementation, these loops were replaced by a vectorized, or batch computation, that leverages parallelization of the computation. Specifically, we performed vectorized time evolution of the controlled dynamics over the trajectory rollouts, and performed CPU parallelization over prarameter rollouts. For

our experiments, the dynamics were initialized by the QuTip rand-ket functionality.

V. LEARNING TO CONTROL TWO QUBITS

To illustrate the efficacy of the QGASS framework, we now demonstrate its use in practice. Specifically, we consider the control problem of stabilizing a two-qubit system to one of the Bell pair states of maximal entanglement. A stable solution to this problem has been proven [71], though the optimality of the solution was not considered. Consider the two-qubit quantum system given by the SME [71]:

$$\begin{aligned} d\rho_t^c = &-iu_1(t)\big[\sigma_y^{(1)}, \rho_t^c\big]dt - iu_2(t)\big[\sigma_y^{(2)}, \rho_t^c\big]dt \\ &- \frac{1}{2}\big[F_z, \big[F_z, \rho_t^c\big]\big]dt \\ &+ \sqrt{\eta}\big(\big\{F_z, \rho_t^c\big\} - \mathrm{Tr}[(F_z + F_z^\dagger)\rho_t^c\big]\rho_t^c\big)dW_t. \end{aligned} \tag{9}$$

The given task is to reach and stabilize the symmetric up-down, down-up maximally entangled Bell state:

$$|\Psi^+\rangle = \frac{1}{\sqrt{2}}(|\downarrow_1\uparrow_2\rangle + |\uparrow_1\downarrow_2\rangle), \tag{10}$$

starting from a random initial condition. This bell state can be written in density matrix form as

$$\rho_d = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0.5 & 0.5 & 0 \\ 0 & 0.5 & 0.5 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}. \tag{11}$$

Working in the in the basis of two-qubit Pauli operators (i.e., two-qubit span $= \{I \otimes \sigma_x, I \otimes \sigma_y, I \otimes \sigma_z, \dots\}$), the desired state can be written as:

$$\rho_d = \tfrac{1}{4}(I \otimes I + \sigma_x \otimes \sigma_x + \sigma_y \otimes \sigma_y - \sigma_z \otimes \sigma_z). \tag{12}$$

We focus on the expectations of these basis elements with respect to the conditioned density evolution to evaluate performance.

The performance is measured by a running cost metric given by:

$$J(\rho, \mathbf{u}) := \int_0^T \big(Q_s q(1 - \mathrm{Tr}[\rho_d \rho_\tau]) + \mathbf{u}_\tau^\top Q_u \mathbf{u}_\tau\big)d\tau, \tag{13}$$

where Q_s is a state cost weighting and Q_u is a control cost weighting. These were set to $Q_s = \mathrm{diag}(10)$ and $Q_u = \mathrm{diag}(0.1)$. Note that this state cost metric utilizes a computationally efficient trace metric [71] as compared to the standard trace distance metric [72], which is substantially slower in implementation as it requires an eigenvalue decomposition at each time step.

The function $q : [0, 1] \to [0, \alpha]$ is an angle resolution function which is added to help resolve numerically close angular values. Recall that for a single qubit, the trace inner product can be thought of as measuring perpendicularity of Bloch phases. Since the cosine function is relatively flat (derivative near zero) near 0, one may encounter bad numerical resolution near the desired minimum $1 - \mathrm{Tr}[\rho_d \rho_\tau]) = 0$ in an n-qubit setting. The resolving function applies a logarithm transformation to improve numerical resolution, and is given by:

$$q(x) = \alpha \frac{\ln(1 + \beta x)}{\ln(1 + \beta)}, \tag{14}$$

where α is the maximum of the range, and β controls the slope by effectively changing the base of the natural logarithm. In our experiments, $\alpha = 10$, and $\beta = 100$.

The cost function is passed through a differentiable and nonincreasing shape function $S(\cdot) : \mathbb{R} \to \mathbb{R}$. Many different shape functions can be used, as explained in greater detail in Ref. [69]. In our simulated experiments, we used the function

$$S(J) := \exp(-\kappa J), \tag{15}$$

where $\kappa = 1.0$ affects the slope and may be tuned for greater performance.

This experiment utilized a simulation environment built in Python, with state evolution adapted from the QuTip Python library [72], and policy networks coded in PyTorch [73]. All network weights were initialized by PyTorch's default layer initialization, which is a uniform random initialization [74]. The algorithm computation speed is numerically improved by using vectorized (or batch) computations of the simulated trajectories, and CPU parellelization for policy parameter rollouts, resulting in ∼20 seconds per iteration for 1000 time steps of an Euler-Maruyama discretization of Eq. (3) with $R = 50$ rollouts and $P = 200$ policy parameter rollouts. The algorithm was run on a desktop computer with an Intel Xeon 12-core CPU with a NVIDIA GeForce GTX 1060 GPU and used less than 10 GB of RAM.

A single-layer fully connected (FC) policy network was used for the experiment and was trained for 850 iterations of the QGASS algorithm over a 1000 time-step window. The trained policy network was then applied to an unseen test set of dynamics, and achieved quick stabilization convergence. Despite being trained on just 1000 time steps of dynamics, the linear policy was tested and performed effective stabilization on up to 100 000 timesteps of dynamics. We plot its

Algorithm 1 Quantum gradient-based adaptive stochastic search optimization

Function: $\boldsymbol{\theta}^* = \textit{OptimizePolicyVars}(T, K, R, P, \rho_0, \kappa, \varphi^{(0)}, \boldsymbol{\theta}^{(0)}, \sigma)$
for $k = 0$ to K **do**
  $\mu \leftarrow \boldsymbol{\theta}^{(k)}$
  for $p = 0$ to P **do**
    $\varphi_p \leftarrow \mathrm{SampleWeights}(\mu, \sigma)$
    for $r = 0$ to R **do**
      for $t = 0$ to T **do**
        $dW_{t,r} \leftarrow \mathrm{SampleNoise}()$
        $u_{t,r,p} \leftarrow \mathrm{Policy}(\rho_{t,r,p}; \varphi_p)$
        $\rho_{t+1,r,p} \leftarrow \mathrm{Dynamics}(\rho_{t,r,p}, u_{t,r,p}, dW_{t,r})$
        $J_{t,r,p} \leftarrow \mathrm{RunningCost}(\rho_{t,r,p}, u_{t,r,p})$
      end for
      $J_{r,p} \leftarrow \sum_t J_{t,r,p} + \mathrm{TerminalCost}(\rho_{T,r,p})$
    end for
    $S_p \leftarrow \mathrm{ShapeFunction}(J_{r,p}; \kappa)$
  end for
  $\boldsymbol{\theta}^{(k+1)} \leftarrow \gamma\,\mathrm{GradientStep}(\boldsymbol{\theta}^{(k)}, S_p)$
end for

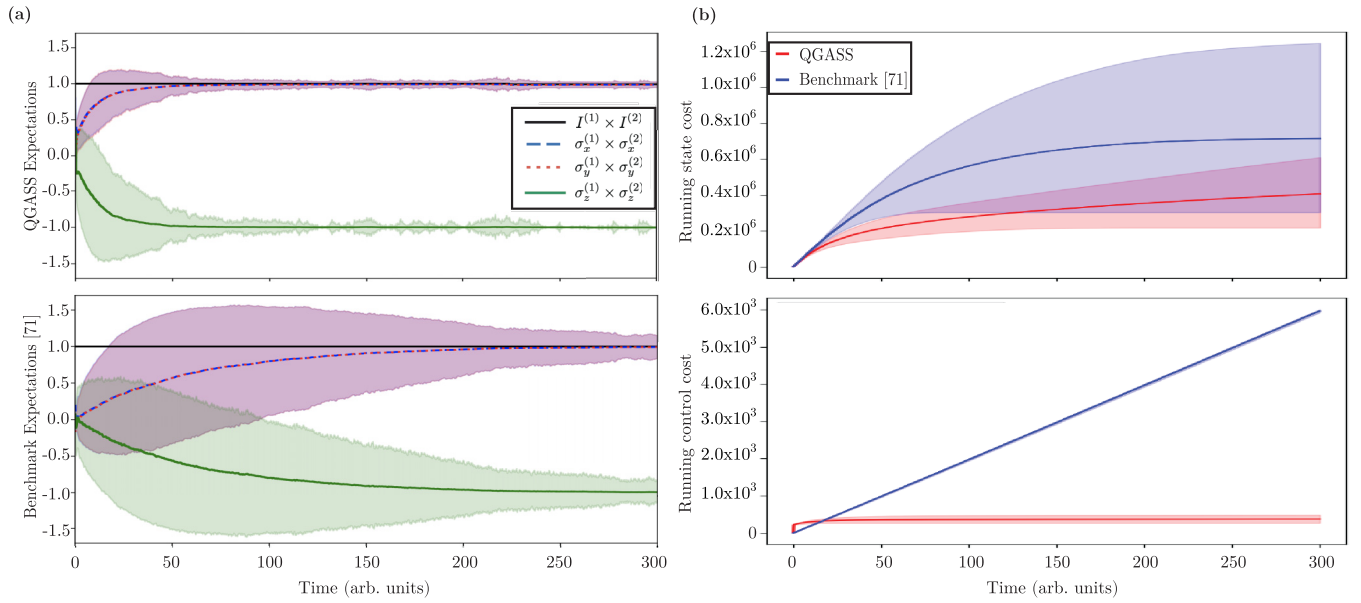

FIG. 2. QGASS performance on a stabilization task. (a) Top panel: Convergence of two-qubit basis elements $I^{(1)} \otimes I^{(2)}$ (red), $\sigma_x^{(1)} \otimes \sigma_x^{(2)}$ (blue), $\sigma_y^{(1)} \otimes \sigma_y^{(2)}$ (black), and $\sigma_z^{(1)} \otimes \sigma_z^{(2)}$ (green) for a linear policy trained by QGASS as a function of time. Lines denote the means and shaded regions denote 2-σ variances, both taken over 1000 sampled trajectories. Bottom panel: Same as top panel for the benchmark feedback policy given by Ref. [71]. (b) Running state costs, top panel, and running control costs, bottom panel, for the linear policy network trained with QGASS (red) and the benchmark policy of Ref. [71] (blue). Lines denote means and shaded regions denote 1-σ variances, both taken over 1000 sampled trajectories. The imposed data symmetry from this Gaussian depiction is corrected only when a large variance would lead to an infeasible negative instantaneous cost

performance in Fig. 2(a). The left subfigure depicts the QGASS method, and the right subfigure depicts the policy suggested in Ref. [71] under identical experimental parameters. Solid lines represent mean expectations of basis elements averaged over 1000 test system trajectories, and shaded regions represent their 2-σ variance. The QGASS method can be observed to converge in approximately one order of magnitude faster than the benchmark and has dramatically lower variance.

The efficacy of the policy trained by QGASS can also be visualized in terms of the cost metric $J(\rho_t^c, u_t)$. In Fig. 2(b), the running cost components of the policy trained by QGASS are depicted. The top subfigure depicts the running state cost component of $J(\rho_t^c, u_t)$, given by:

$$J_{\text{state},t}(\rho_t) := \int_0^t Q_s q(1 - \text{Tr}[\rho_d \rho_\tau]) d\tau, \tag{16}$$

while the bottom subfigure depicts the running control cost component of $J(\rho_t^c, u_t)$, given by:

$$J_{\text{control},t}(\mathbf{u}) := \int_0^t \mathbf{u}_\tau^\top Q_u \mathbf{u}_\tau d\tau, \tag{17}$$

where Q_s and Q_u are state cost and control cost weightings, respectively, which are diagonal matrices weighing the cost along each state or control dimension. The solid line depicts the means of the running cost trajectories of each policy and the shading depicts the 1-σ variance, each computed over 1000 trajectory rollouts. The policy trained by QGASS has a lower state cost on average, with a significantly lower 1-σ variance of state cost, which suggests that the state performance may have better guarantees of performance as compared to Ref. [71]. The control effort of each policy is depicted in the right subfigure: it is observed that the policy trained by QGASS applies a strong initial control impulse to the system followed by a relatively small control signal: this policy can be interpreted as a form of bang-bang control. In contrast, the policy of Ref. [71] injects a fairly constant control signal over the time window, which yields a cumulative control effort approximately 12× higher than that of QGASS.

This same state performance may also be visualized through the commonly used fidelity measure, which is typically used as a distance measure on qubit states and can analogously measure task performance. The fidelity measure $\mathcal{F}(\cdot,\cdot) : H^2 \times H^2 \to [0, 1]$ may be thought of as an inverse cost measure on [0,1] as values closer to 1.0 correspond to qubits that are more closely aligned, that is, $\mathcal{F}(\rho_1, \rho_2) \to 1.0$ as $\rho_1 \to \rho_2$. The results are depicted in Eq. (3), and again demonstrate the advantages of the policy trained by QGASS both in terms of convergence time as well as lower variance on the performance. Note that this measure was not used to measure performance during training since it has a much higher computational complexity compared to Eq. (16).

Note that the impulsive control signal produced by the trained policy is likely to be experimentally realizable due to the viability of pulsed electromagnetic fields. However, if one were to desire a less impulsive control signal, one could add a running penalization term on the derivative of the control, effectively penalizing large rates of change in the control signal applied to the system [75]. One could also add a control rate indicator, effectively suppressing this additional cost until some control rate threshold is reached. This flexibility is possible since we do *not* require any differentiability or even continuity of the cost functional in the QGASS framework. While this will likely lead to a larger time-integrated

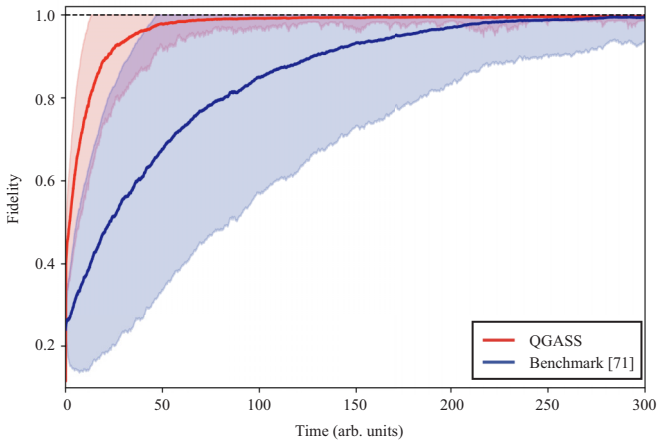

FIG. 3. Comparison of the performance of the policy trained by QGASS, depicted in red, against the benchmark policy [71], depicted in blue. Solid lines denote means and shaded regions denote 1-σ variances, both taken over 1000 sampled trajectories. The imposed data symmetry from this mean and variance Gaussian fit to the data is corrected only when a large variance would lead to an infeasible fidelity greater than 1.0

control effort, one may either introduce terms to the cost functional or introduce hard constraints to enforce various experimental hardware constraints, including bounds on the control rate.

VI. DISCUSSION AND OUTLOOK

In this paper, we suggest a method, the QGASS framework, of training networks to utilize state information in an explicit state-dependent feedback control scheme, allowing for successful extrapolation and generalization of the trained feedback scheme outside of the training window. State-dependent feedback control can be contrasted with existing methods that utilize a type of time-dependent control that is iteratively improved through measurement-based feedback. While omitting explicit feedback is often much simpler from a controller synthesis and optimization perspective, state-dependent feedback provides numerous benefits including stabilization performance over larger timescales.

Our results also demonstrate the efficacy of applying principles from stochastic optimization for MFC. We applied QGASS to train explicit feedback policy networks to control and stabilize a two-qubit experiment. The generality of the problem formulation suggests that this approach can be applied to the large class of problems involving open quantum systems with either continuous measurement schemes or discrete measurement schemes. Furthermore, this approach is quite flexible; while we have only considered one form of a cost functional and shape functional, there are virtually no limitations on the form of the cost functional. Combined with the fast computational speed of iterations, the results suggest that this approach can scale to larger numbers of qubits, which we are actively exploring.

One can also generalize the QGASS framework to train parametric coherent controllers coupled to the quantum system through implicit feedback (i.e., without an explicitly recorded measurement outcome). In such a case, the dynamics follow (deterministic) unconditional evolution governed by the Lindblad equation, forgoing a weak measurement process entirely. Furthermore, we conjecture that through careful experiment design, it may be possible to apply QGASS in a MFC setting to learn a parametric coherent controller, which is then applied to the system in a CFC setting, which holds advantages in certain contexts [42,76].

Our approach has similarities to the policy gradient method [35], wherein gradients of the cost functional with respect to the control policy parameters are computed. However, the QGASS approach is distinctive in that it assumes a probability distribution on the policy parameters and performs gradient updates on the distribution parameters instead. This allows one to bypass the often problematic differentiation steps through the cost functional and dynamics as in policy gradient methods. In addition, this means that our framework can be easily applied to problems with nondifferentiable or discontinuous dynamics and cost functions, e.g., photon counting and control thresholding.

The QGASS derivation for the policy parameter update holds true for arbitrary policy networks. In the simulated two-qubit control experiment, we used a rather shallow and simple linear feedback policy parametrization. An interesting next step is to investigate the effect of the size and depth of the policy network. The widespread success of utilizing deep networks for a variety of learning applications suggests that deeper network architectures may outperform the results presented here, especially for experiments with larger numbers of qubits. In addition, for more complex and higher dimensional experiments, convolutional or long-short term memory networks are also worth exploring to improve the scalability and temporal correlation of the policy.

ACKNOWLEDGMENTS

We thank K. Jacobs for the helpful comments and discussions concerning our paper. E.N.E. was supported by the Department of Defense though the SMART Scholarship program and the SMART SEED grant [77]. A.G.F. is supported by the NSF GRFP under Grant No. DGE 1752814. This paper was in part supported by the Army Research Office Contract No. W911NF2010151.

APPENDIX A: DEGENERATE DIFFUSIONS IN THE STOCHASTIC MASTER EQUATION

The measurement-based feedback scheme for control of quantum systems has many advantages, as outlined in the main text. However, this scheme also has certain pitfalls, which include problems of degeneracy of the diffusion dynamics. Degenerate diffusions can cause many control frameworks to fail, often due to the inability to find inverses or pseudoinverses of the covariance operator. In this section, we demonstrate two common examples in the context of quantum feedback control where this degeneracy may emerge due to a singular covariance operator.

1. Degenerate diffusions in the two-qubit system with continuous measurement

Consider the two-qubit quantum system given by the SME [71]:

$$\begin{aligned} d\rho_t^c &= -iu_1(t)\big[\sigma_y^{(1)}, \rho_t^c\big]dt - iu_2(t)\big[\sigma_y^{(2)}, \rho_t^c\big]dt \\ &\quad - \frac{1}{2}\big[F_z, \big[F_z, \rho_t^c\big]\big]dt \\ &\quad + \sqrt{\eta}\Big(\big\{F_z, \rho_t^c\big\} - \mathrm{Tr}\big((F_z + F_z^\dagger)\rho_t^c\big)\rho_t^c\Big)dW_t, \end{aligned} \tag{A1}$$

where $u_j(t)$ are two time-varying magnetic fields coupled to the two qubits, $F_z := \sigma_z^{(1)} \otimes I^{(2)} + I^{(1)} \otimes \sigma_z^{(2)}$ defines the coupling between the cavity and the electromagnetic field produced by the probe laser, as depicted in Ref. [71], Fig. 1.1, and as usual $[\cdot, \cdot]$ and $\{\cdot, \cdot\}$ are the commutator and anticommutator, respectively. We can simplify this equation by defining the usual superoperators:

$$\mathcal{H}[A]\rho := \{A, \rho\} - 2\mathrm{Tr}(A\rho)\rho, \tag{A2}$$

$$\mathcal{D}[A]\rho := \tfrac{1}{2}[A, [A, \rho]]. \tag{A3}$$

Also, note that in this system $H = 0$ and $H_{u,j} = \sigma_y^{(j)}$. This yields:

$$\begin{aligned} d\rho_t^c &= -\sum_j u_j(t)\big[H_{u_j}, \rho_t^c\big]dt - \mathcal{D}[F_z]\rho_t^c dt \\ &\quad + \sqrt{\eta}\mathcal{H}[F_z]\rho_t^c dW_t, \end{aligned} \tag{A4}$$

so we have the form in Eq. (4) repeated here for clarity,

$$d\rho_t^c = F\big(\rho_t^c\big)dt + G\big(\rho_t^c\big)\mathbf{u}(t)dt + B\big(\rho_t^c\big)dW_t, \tag{A5}$$

where we have defined the superoperators:

$$F(\rho) := -\mathcal{D}[F_z]\rho, \tag{A6}$$

$$G(\rho)u(t) := -i\sum_j u_j(t)[H_{u_j}, \rho], \tag{A7}$$

$$B(\rho) := \sqrt{\eta}\mathcal{H}[F_z]\rho. \tag{A8}$$

A key requirement of many stochastic optimal control methods is the invertability of the superoperator $\mathcal{H}[\cdot]$, which can become singular. We can see this by simply looking at the F_z operator. In this case, it becomes

$$\begin{aligned} F_z &:= I^{(1)} \otimes \sigma_z^{(2)} + \sigma_z^{(1)} \otimes I^{(2)} \\ &= \begin{bmatrix} \sigma_z & 0 \\ 0 & \sigma_z \end{bmatrix} + \begin{bmatrix} I & 0 \\ 0 & -I \end{bmatrix} \\ &= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix} + \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix} \\ &= \begin{bmatrix} 2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -2 \end{bmatrix}. \end{aligned} \tag{A9}$$

Thus, if the system is in the Bell state,

$$\rho_{\text{desired}} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0.5 & 0.5 & 0 \\ 0 & 0.5 & 0.5 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \tag{A10}$$

the superoperator $\mathcal{H}[F_z]\rho$ is a singular operator. The singularity above also arises if we rotate the magnetic fields such that they are coupled to the x or y axis of the spin representation, so we have the coupling operator as $F_x := \sigma_x^{(1)} \otimes I^{(2)} + I^{(1)} \otimes \sigma_x^{(2)}$ or $F_y := \sigma_y^{(1)} \otimes I^{(2)} + I^{(1)} \otimes \sigma_y^{(2)}$. In either of the three cases, the eigenvalues are $\lambda = \{0, 0, 2, 2\}$. This is the case for any n-qubit system.

2. Degenerate diffusions in the homodyne continuous measurement experiment

The homodyne detection experiment was among the first nondemolition measurement experiments, and can be viewed from the photon counting (jump noise) or continuous diffusion (Brownian noise) cases. In this experiment, a cavity system emits photons when the atoms in the cavity are excited. The photon leakage is mixed with a local oscillator of the same frequency, and the mixed beam is then detected. The experimental setup is depicted in Ref. [78], Fig. A1.

The dynamics of the dissipative homodyne detection experiment are given in Fock space by the SME:

$$\begin{aligned} d\rho_t &= -i[H_0, \rho_t]dt - i\sum_j u_j[H_{u_j}, \rho_t]dt \\ &\quad - \frac{1}{2}\sqrt{1-\eta}\sqrt{\gamma}[a, [a, \rho_t]]dt \\ &\quad + \sqrt{\eta}\sqrt{\gamma}(\{a, \rho_t\} - 2\mathrm{Tr}(a\rho_t)\rho_t)dW_t, \end{aligned} \tag{A11}$$

where H_0 is the typical unforced Hamiltonian of the quantum harmonic oscillator, a is the usual annihilation operator, and $\mathbf{H}_u$ is the Hamiltonian of the external forcing, in this case provided by a coupled electromagnetic field. Using the previously defined $\mathcal{D}$ and $\mathcal{H}$ superoperators in Eqs. (A3) and (A2) yields the simplified form:

$$\begin{aligned} d\rho_t &= -i[H_0, \rho_t]dt - i\sum_j u_j[H_{u_j}, \rho_t]dt \\ &\quad - \sqrt{1-\eta}\sqrt{\gamma}\mathcal{D}[a]\rho_t dt + \sqrt{\eta}\sqrt{\gamma}\mathcal{H}[a]\rho_t dW_t. \end{aligned} \tag{A12}$$

Again we have the form in Eq. (4), with:

$$F(\rho_t) := -i[H_0, \rho_t] - \sqrt{1-\eta}\sqrt{\gamma}\mathcal{D}[a]\rho_t, \tag{A13}$$

$$G(\rho_t)u(t) := -i\sum_j u_j(t)[H_{u_j}, \rho_t], \tag{A14}$$

$$B(\rho_t) := \sqrt{\eta}\sqrt{\gamma}\mathcal{H}[a]\rho_t. \tag{A15}$$

Investigating the $B(\rho_t)$ operator, we again find that it is singular, as can be seen by the form of the matrix representation of

the annihilation operator a for an N-level cavity,

$$a = \begin{bmatrix} 0 & \sqrt{1} & 0 & \cdots & & 0 \\ 0 & 0 & \sqrt{2} & 0 & \cdots & 0 \\ \vdots & & \ddots & & & \vdots \\ 0 & & \cdots & & 0 & \sqrt{N} \\ 0 & & \cdots & & & 0 \end{bmatrix}, \tag{A16}$$

which again leads to a singular superoperator for certain states. This singularity also arises if we use the creation operator $a^\dagger$ as the coupling operator.

APPENDIX B: QGASS FORMULATIONS FOR LEARNING FEEDBACK POLICIES

The GASS method was introduced in Ref. [64] and has recently been applied as a control optimization strategy (cf. Ref. [79]). This approach has provable convergence characteristics, and offers generality and flexibility. In this Appendix, we will demonstrate this flexibility by exploring several problem formulations that leverage the approach for training policy networks for feedback control.

Consider the two qubit Stochastic Master Equation (SME) in the general simplified form:

$$d\rho_t = F(\rho_t)dt + G(\rho_t, \mathbf{u}_t)dt + B(\rho_t)dW_t, \tag{B1}$$

where $G(\rho_t, \mathbf{u}_t)$ is a state-dependent controlled drift term. In the two-qubit problem, $G(\rho_t, \mathbf{u}_t)$ takes the form $G(\rho_t, \mathbf{u}_t) = \sum_i^2 u_i[\sigma_y^{(i)}, \rho_t]$, however, in a more general N-qubit experiment, one may require all single-particle Pauli matrices. Thus, $G(\rho_t, \mathbf{u}_t)$ may have the more general form:

$$G(\rho_t, \mathbf{u}_t) = \sum_{i,j=1}^{N,3} u_{ij}\left[\sigma_j^{(i)}, \rho_t\right], \tag{B2}$$

where $\sigma_j^{(i)}$, $j \in 1, 2, 3$ denote single-particle Pauli matrices of each axis x, y, z. Despite appearing in the context of qubit systems, the form of Eq. (B1) is quite general and can represent virtually *any* open quantum system with continuous weak measurement.

Quantum control problems often consider the task of reaching some target state ρ_{des}, as measured by some general cost metric $J(\rho_t, \mathbf{u}_t)$. The minimizing control is most generally expressed by the following path integral optimization problem,

$$\mathbf{u}^* = \underset{\mathbf{u}\in\mathcal{U}}{\text{argmin}} \langle J(\rho, \mathbf{u})\rangle_Q, \tag{B3a}$$

$$\text{such that} \quad d\rho_t = F(\rho_t)dt + G(\rho_t, \mathbf{u}_t)dt + B(\rho_t)dW_t, \tag{B3b}$$

where the expectation defines a path integral over controlled state trajectories with measure Q. The set $\mathcal{U}$ is the admissible set of controls and may impose constraints on the control. One may also include constraints on state ρ, however, these are omitted from this derivation for simplicity.

The cost functional $J : H^2 \times \mathbb{R}^m \to \mathbb{R}$ is some real-valued, potentially nonconvex, discontinuous, and nondifferentiable functional, which must be minimized. Such a function imposes many difficulties from the context of optimization theory and optimal control theory. In the GASS approach, we bypass these difficulties through stochastic approximation. Let $f(u;\boldsymbol{\theta})$ be a distribution belonging to the exponential family of distributions. Then the optimization problem is approximated as:

$$\theta^* = \underset{\boldsymbol{\theta}}{\text{argmin}} \langle J(\rho, \mathbf{u})\rangle_{Q,f(\mathbf{u};\boldsymbol{\theta})}, \tag{B4a}$$

$$\text{s.t.} \quad d\rho_t = F(\rho_t)dt + G(\rho_t, \mathbf{u}_t)dt + B(\rho_t)dW_t, \tag{B4b}$$

$$\mathbf{u}_t \sim f(\mathbf{u}_t;\boldsymbol{\theta}). \tag{B4c}$$

Furthermore, we introduce the smooth (continuously differentiable almost everywhere), nonincreasing shape function $S : \mathbb{R} \to \mathbb{R}$ and the logarithm function to obtain the following modified optimization problem:

$$\theta^* = \underset{\boldsymbol{\theta}}{\text{argmax}} \log \langle S(J(\rho, \mathbf{u}))\rangle_{Q,f(u;\boldsymbol{\theta})}, \tag{B5a}$$

$$\text{such that} \quad d\rho_t = F(\rho_t)dt + G(\rho_t, \mathbf{u}_t)dt + B(\rho_t)dW_t, \tag{B5b}$$

$$\mathbf{u}_t \sim f(\mathbf{u}_t;\boldsymbol{\theta}). \tag{B5c}$$

Solving this optimization problem with gradient-based parameter adaptation has been shown to have numerous appealing convergence characteristics detailed in Ref. [64], however, a key observation is that this formulation does not incorporate the measurement from the measurement process dW and is a purely feed-forward control. In this representation, one may compare this framework to popular feedforward frameworks such as GRAPE or Krotov for optimal control of quantum systems without state feedback (cf. the approaches in Ref. [26]), however, the goal in defining the Stochastic Master Equation (SME) in Eq. (B1) is to realize an explicit state feedback control optimization algorithm. In the following, we consider a number of modifications to the above optimization problem to achieve this goal, each able to leverage the QGASS training approach as outlined in the main paper.

1. SME with linear parametric state feedback compensation

Consider the optimization problem:

$$\boldsymbol{\theta}^* = \underset{\boldsymbol{\theta}}{\text{argmax}} \log \langle S(J(\rho, \mathbf{u}))\rangle_{Q,f(\varphi;\boldsymbol{\theta})}, \tag{B6a}$$

$$\text{s.t.} \quad d\rho_t = F(\rho_t)dt + G(\rho_t, \mathbf{u}_t)dt + B(\rho_t)dW_t, \tag{B6b}$$

$$\mathbf{u}_t = K_1(\varphi_1)\rho_t + K_2(\varphi_2), \tag{B6c}$$

$$\varphi := [\varphi_1, \varphi_2] \sim f(\varphi;\boldsymbol{\theta}). \tag{B6d}$$

Under the realization that the controller in Ref. [71] is quite similar to a P controller on the trace distance to the goal state, this has a static compensator with an explicit parametric linear feedback policy. The expectation in Eq. (B6a) is a double expectation composed of an expectation over the Stochastic Master Equation (SME) and an expectation over the exponential family.

Note that this control policy can be realized through a fully connected network with rectified linear unit activations as:

$$\mathbf{u}_t = K(\rho_t;\varphi), \tag{B7}$$

where $K : H^2 \to \mathbb{R}^m$ is the linear policy network. This motivates the use of nonlinear policy networks.

2. SME with nonlinear parametric state feedback compensation

Consider the optimization problem:

$$\boldsymbol{\theta}^* = \underset{\boldsymbol{\theta}}{\text{argmax}} \log \langle S(J(\rho, \mathbf{u}))\rangle_{Q, f(\varphi;\boldsymbol{\theta})}, \tag{B8a}$$

$$\text{s.t.} \quad d\rho_t = F(\rho_t)dt + G(\rho_t, \mathbf{u}_t)dt + B(\rho_t)dW_t, \tag{B8b}$$

$$\mathbf{u}_t = \boldsymbol{\Phi}(\rho_t; \varphi), \tag{B8c}$$

$$\varphi \sim f(\varphi; \boldsymbol{\theta}), \tag{B8d}$$

where $\boldsymbol{\Phi}$ is a nonlinear feedback policy parametrized by φ. This could be a FC network or a convolutional neural network but, in general, simply represents a nonlinear function of ρ without explicit time dependence.

3. SME with nonlinear recurrent state feedback compensation

Consider the optimization problem:

$$\boldsymbol{\theta}^* = \underset{\boldsymbol{\theta}}{\text{argmax}} \log \langle S(J(\rho, \mathbf{u}))\rangle_{Q, f(\varphi,\boldsymbol{\theta})}, \tag{B9a}$$

$$\text{s.t.} \quad d\rho_t = F(\rho_t)dt + G(\rho_t, \mathbf{u}_t)dt + B(\rho_t)dW_t, \tag{B9b}$$

$$\mathbf{u}_{t_{k+1}} = \boldsymbol{\Phi}_{\text{RNN}}(\rho_{t_k}, \mathbf{u}_{t_k}; \varphi), \tag{B9c}$$

$$\varphi \sim f(\varphi; \boldsymbol{\theta}), \tag{B9d}$$

where $\boldsymbol{\Phi}_{\text{RNN}}$ is a recurrent neural network (RNN) (e.g., LSTM network). Incorporating time dependence in the policy endows the compensator with dynamics and enables treatment of a larger class of problems compared to a static compensator.

One may also apply a neural ordinary differential equation (NODE) network [80] in place of Eq. (B9c). Instead of specifying a discrete sequence of hidden layers, NODE networks parametrize the derivative of the hidden state using a neural network and, as a result, demonstrate *constant* memory cost as a function of network depth, significantly lower training losses, and can handle time irregularity in the discretization scheme. In many cases, NODE networks outperform RNN networks, and are a closer representation to a dynamic compensation approach.

4. SME with stochastic actuators and dynamic feedback compensation

Consider the optimization problem:

$$\boldsymbol{\theta}^* = \underset{\boldsymbol{\theta}}{\text{argmax}} \log \langle S(J(\rho, \mathbf{u}))\rangle_{Q, U, f(\varphi;\boldsymbol{\theta})}, \tag{B10a}$$

$$\text{such that} \quad d\rho_t = F(\rho_t)dt + G(\rho_t, \mathbf{u}_t)dt + B(\rho_t)dW_t, \tag{B10b}$$

$$d\mathbf{u}_t = G_u(\rho_t, \mathbf{u}_t; \varphi_1) + \Sigma dV_t, \tag{B10c}$$

$$\mathbf{u}_{t_0} = G_0(\varphi_2), \tag{B10d}$$

$$\varphi := [\varphi_1, \varphi_2] \sim f(\varphi; \boldsymbol{\theta}), \tag{B10e}$$

where Q is the measure of the controlled dynamics, U is the measure of the dynamic compensator, and $f(\varphi; \boldsymbol{\theta})$ is a distribution, parameterized by $\boldsymbol{\theta}$, which belongs to the exponential family of distributions. We include noise in the compensator to represent a realistic noisy digital compensation signal, however, this can be neglected to reduce the sampling complexity. The function $J : H^2 \times \mathbb{R}^m \to \mathbb{R}$ is some real-valued, potentially nonconvex and nondifferentiable metric, and the function $S : \mathbb{R} \to \mathbb{R}$ is a smooth shape function. The function $G_u : H^2 \times \mathbb{R}^m \times \mathbb{R}^p$ is the drift of the dynamic compensator.

Here, we must approximate the expectation with finite samples from three processes, namely, the original Stochastic Master Equation (SME), the stochastic dynamic compensator, and the compensator initial condition distribution. This approach may enable substantially more exploration of the state space, however, this comes at the cost of sampling *three* distributions, which can quickly become computationally expensive. One may notice that these compensator dynamics are functionally similar to a stochastic RNN.

APPENDIX C: QGASS PARAMETER UPDATE DERIVATION

The GASS method was first derived in Ref. [64]. Here we derive the parameter update under a minimization problem instead of a maximization problem, and use the above notation. Start with the general optimization problem:

$$\mathbf{u}^* = \underset{\mathbf{u}\in\mathcal{U}}{\text{argmin}}\, J(\rho, \mathbf{u}), \tag{C1}$$

where $\mathcal{U} \subseteq \mathbb{R}^n$ is a nonempty compact set and $J : H \times \mathcal{U} \to \mathbb{R}$ is a real-valued, potentially nonconvex, discontinuous, and/or nondifferentiable function. We avoid the inherent difficulties in $F(\mathbf{u})$ by transforming the problem into an approximation where $\mathbf{u}$ is sampled from the distribution $f(\mathbf{u}; \boldsymbol{\theta})$:

$$\boldsymbol{\theta}^* = \underset{\boldsymbol{\theta}}{\text{argmin}} \int_{\mathcal{U}} J(\rho, \mathbf{u}) f(\mathbf{u}; \boldsymbol{\theta}) d\mathbf{u} = \langle J(\rho, \mathbf{u})\rangle_{f(\mathbf{u};\boldsymbol{\theta})}. \tag{C2}$$

The new problem formulation optimizes with respect to an upper bound of the original one since $\langle J(\rho, \mathbf{u})\rangle_{f(\mathbf{u};\boldsymbol{\theta}^*)} \geqslant J(\rho, \mathbf{u}^*)$. Equality is achieved when all the probability mass of $f(\mathbf{u}; \boldsymbol{\theta}^*)$ is at $\mathbf{u}^*$. To facilitate the derivation, we additionally introduce a logarithmic transform and a shape function that is differentiable and nonincreasing, $S(\cdot) : \mathbb{R} \to \mathbb{R}_+$, which transforms our minimization problem into a maximization one:

$$\boldsymbol{\theta}^* = \underset{\boldsymbol{\theta}}{\text{argmax}} \log \int_{\mathcal{U}} S\big(J(\rho, \mathbf{u})\big) f(\mathbf{u}; \boldsymbol{\theta}) d\mathbf{u} \tag{C3}$$

$$= \log \langle S(J(\rho, \mathbf{u}))\rangle_{f(\mathbf{u};\boldsymbol{\theta}).} \tag{C4}$$

We perform gradient updates to update the parameters $\boldsymbol{\theta}$ of the distribution $f(\mathbf{u}; \boldsymbol{\theta})$, which is assumed to belong to the exponential family of distributions:

$$\nabla_{\boldsymbol{\theta}} \log \int_{\mathcal{U}} S(J(\rho, \mathbf{u})) f(\mathbf{u}; \boldsymbol{\theta}) d\mathbf{u} \tag{C5}$$

$$= \frac{\nabla_{\boldsymbol{\theta}} \int_{\mathcal{U}} S(J(\rho, \mathbf{u})) f(\mathbf{u}; \boldsymbol{\theta}) d\mathbf{u}}{\int_{\mathcal{U}} S(J(\rho, \mathbf{u})) f(\mathbf{u}; \boldsymbol{\theta}) d\mathbf{u}} \tag{C6}$$

$$= \frac{\int_{\mathcal{U}} S(J(\rho,\mathbf{u}))\nabla_{\boldsymbol{\theta}} f(\mathbf{u};\boldsymbol{\theta})d\mathbf{u}}{\int_{\mathcal{U}} S(J(\rho,\mathbf{u}))f(\mathbf{u};\boldsymbol{\theta})d\mathbf{u}}. \tag{C7}$$

Now we apply the log trick $\nabla_{\boldsymbol{\theta}} f(\mathbf{u};\boldsymbol{\theta}) = f(\mathbf{u};\boldsymbol{\theta})\nabla_{\boldsymbol{\theta}} \log f(\mathbf{u};\boldsymbol{\theta})$ to obtain:

$$\nabla_{\boldsymbol{\theta}} \log \int_{\mathcal{U}} S(J(\rho,\mathbf{u}))f(\mathbf{u};\boldsymbol{\theta})d\mathbf{u} \tag{C8}$$

$$= \frac{\int_{\mathcal{U}} S(J(\rho,\mathbf{u}))f(\mathbf{u};\boldsymbol{\theta})\nabla_{\boldsymbol{\theta}} \log f(\mathbf{u};\boldsymbol{\theta})d\mathbf{u}}{\int_{\mathcal{U}} S(J(\rho,\mathbf{u}))f(\mathbf{u};\boldsymbol{\theta})d\mathbf{u}} \tag{C9}$$

$$= \frac{\langle S(J(\rho,\mathbf{u}))\nabla_{\rho} \log f(\mathbf{u};\boldsymbol{\theta})\rangle_{f(\mathbf{u};\boldsymbol{\theta})}}{\langle S(J(\rho,\mathbf{u}))\rangle_{f(\mathbf{u};\boldsymbol{\theta})}}. \tag{C10}$$

The exponential family distribution is given by:

$$f(\mathbf{u};\boldsymbol{\theta}) = h(\mathbf{u})(\boldsymbol{\theta}^{\top}T(\mathbf{u}) - A(\boldsymbol{\theta})), \tag{C11}$$

which is characterized by a set of natural parameters $\boldsymbol{\theta}$, sufficient statistics $T(\mathbf{u})$, base measure $h(\mathbf{u})$, and a log partition function $A(\boldsymbol{\theta})$. Thus, we have

$$\nabla_{\boldsymbol{\theta}} \log f(\mathbf{u};\boldsymbol{\theta}) = \nabla_{\boldsymbol{\theta}} \log[h(\mathbf{u})\exp(\boldsymbol{\theta}^{\top}T(\mathbf{u}) - A(\boldsymbol{\theta}))] \tag{C12}$$

$$= \nabla_{\boldsymbol{\theta}} \log h(\mathbf{u}) + \nabla_{\boldsymbol{\theta}}(\boldsymbol{\theta}^{\top}T(\mathbf{u}) - A(\boldsymbol{\theta})) \tag{C13}$$

$$= T(\mathbf{u}) - \nabla_{\boldsymbol{\theta}} A(\boldsymbol{\theta}). \tag{C14}$$

If one optimizes only over the mean of a Gaussian distribution, then one obtains:

$$\nabla_{\boldsymbol{\theta}} \log \langle S(J(\rho,\mathbf{u}))f(\mathbf{u};\boldsymbol{\theta})\rangle_{f(\mathbf{u};\boldsymbol{\theta})} \tag{C15}$$

$$= \frac{\langle S(J(\rho,\mathbf{u}))\Sigma^{-1}(\mathbf{u}-\mu)\rangle_{f(\mathbf{u};\boldsymbol{\theta})}}{\langle S(J(\rho,\mathbf{u}))\rangle_{f(\mathbf{u};\boldsymbol{\theta})}}, \tag{C16}$$

where μ is the mean and Σ is the variance. Thus, the gradient-ascent parameter update becomes:

$$\Sigma^{-1}\mu^{k+1} = \Sigma^{-1}\mu^{k} + \Sigma^{-1}\frac{\langle S(J(\rho,\mathbf{u}))(\mathbf{u}-\mu)\rangle_{f(\mathbf{u};\boldsymbol{\theta})}}{\langle S(J(\rho,\mathbf{u}))\rangle_{f(\mathbf{u};\boldsymbol{\theta})}} \tag{C17}$$

or, more simply,

$$\mu^{k+1} = \mu^{k} + \frac{\langle S(J(\rho,\mathbf{u}))(\mathbf{u}-\mu^{k})\rangle_{f(\mathbf{u};\boldsymbol{\theta})}}{\langle S(J(\rho,\mathbf{u}))\rangle_{f(\mathbf{u};\boldsymbol{\theta})}}. \tag{C18}$$

Note that in the cases where we have added a level of abstraction due to the inclusion of a parameterized policy network $\Phi(\rho_t;\varphi)$, the above derivation yields a parameter update,

$$\mu^{k+1} = \mu^{k} + \frac{\langle S(J(\rho,\mathbf{u}))(\varphi-\mu^{k})\rangle_{f(\varphi;\boldsymbol{\theta})}}{\langle S(J(\rho,\mathbf{u}))\rangle_{f(\varphi;\boldsymbol{\theta})}}, \tag{C19}$$

where in this case μ is the mean of a Gaussian distribution on the policy network parameters φ.

Due to the path integral nature of this derivation, the so-called QGASS approach is independent of the discretization scheme used to discretize the dynamics in Eq. (B1), as in the quantum trajectories literature. Furthermore, this approach may consider jump-diffusion dynamics, such as in the discrete measurement case [32].

Several of the above optimization problems contain two or three expectations, which is quite different than the above case wherein the parameter update was derived. To apply this parameter update to the two and three expectation cases above, one must simply redefine the shape function. In the cases of Eqs. (B5), (B6), (B8), and (B9), let the function $S(\cdot)$ be defined as

$$S(\cdot) := \langle \hat{S}(\cdot)\rangle_{Q}, \tag{C20}$$

where $\hat{S}$ is a standard shape function which is differentiable and nonincreasing. Thus, S is nonincreasing and positive semidefinite, so it may be treated as a shape function. This shape function may be substituted into Eq. (C19) to yield:

$$\mu^{k+1} = \mu^{k} + \frac{g\langle\langle \hat{S}(J(\rho,\mathbf{u}))\rangle_{Q}(\varphi-\mu)g\rangle_{f(\varphi;\boldsymbol{\theta})}}{g\langle\langle \hat{S}(J(\rho,\mathbf{u}))\rangle_{Q}g\rangle_{f(\varphi;\boldsymbol{\theta})}}. \tag{C21}$$

Similarly, for Eq. (B10), let the function $S(\cdot)$ be defined as:

$$S(\cdot) := g\langle\langle \hat{S}(\cdot)\rangle_{U}g\rangle_{Q}. \tag{C22}$$

In this case, $S(\cdot)$ is also nonincreasing and differentiable, so it may be treated as a shape function. This results in the parameter update:

$$\mu^{k+1} = \mu^{k} + \frac{g\langle g\langle\langle \hat{S}(J(\rho,\mathbf{u}))\rangle_{U}g\rangle_{Q}(\varphi-\mu)g\rangle_{f(\varphi;\boldsymbol{\theta})}}{g\langle g\langle\langle \hat{S}(J(\rho,\mathbf{u}))\rangle_{U}g\rangle_{g}\rangle_{f(\varphi;\boldsymbol{\theta})}}. \tag{C23}$$

The parameter update in Eq. (C19) can be connected to the information theoretic version of the model predicitive path integral (MPPI) algorithm for classical systems [81], as explored in Ref. [69]. The information theoretic MPPI algorithm applies an exponential shape function $S(y;\kappa) := \exp(-\kappa y)$ for $y, \kappa \in \mathbb{R}$, however other shape functions, such as the sigmoid function, are explored in Ref. [69].

The key difference between the QGASS approach compared to MPPI [81] is that MPPI requires one to perform importance sampling, which presents challenges when the diffusion process becomes degenerate, namely, the change of measures between the controlled and uncontrolled open quantum systems with continuous measurement requires inversion of an operator that is singular in a multitude of realizable experiments, such as the two-qubit system and the homodyne system.

In the context of Ref. [63], policies without explicit time dependence have been shown to effectively control a number of SPDE systems for reaching and stabiliziation tasks, however, these policies can fail for tracking tasks. Both of these approaches are algorithmically quite similar, and may have theoretic connections if one can connect the objective in GASS to an analogous free-energy relative entropy relationship. Aside from the differences in the resulting loss functional, another primary difference between the two approaches can be summarized by observing Eq. (C7), wherein one passes the gradient directly to the distribution $f(\varphi;\boldsymbol{\theta})$ and skips the implicit dependence of $S(J(\rho))$ on θ. This skipped gradient path enables one to bypass the potential discontinuities and nondifferentiability of J, however, in some sense ignores these contributions to the total gradient. These skipped

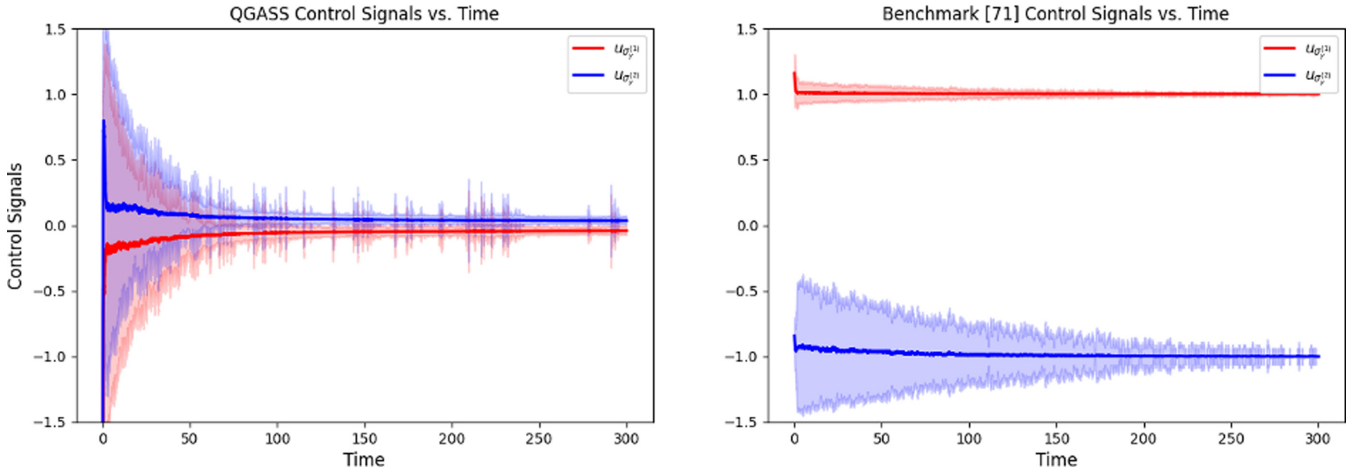

FIG. 4. Comparison of control signals provided by the (a) policy network trained by QGASS and (b) benchmark policy [71]. Control signals are zoomed in to depict the stabilization portion of the trained policy. Solid lines denote means and shaded regions denote 2-σ variances, both taken over 1000 sampled trajectories

connections may ignore important gradient information, however, they offer flexibility and maintain provable convergence and convergence rate characteristics [64].

APPENDIX D: STABILIZING CONTROL POLICIES FOR THE TWO-QUBIT EXPERIMENT

The primary results of this paper demonstrate strong performance of the policy trained by QGASS compared to the benchmark policy. As previously mentioned, this policy exhibits a bang-bang type control signal, where a strong impulsive signal is followed by a weaker stabilizing signal. These control signals are shown in Eq. (4).

This result is quite interesting, especially in the context of other optimal control approaches such as GRAPE, wherein control pulses are optimized via gradient ascent. In the limit, control pulses can be made arbitrarily similar to a bang-bang solution. In the case of policies trained by QGASS, these control pulses can effectively *react* to system measurement. The emergence of this sort of control solution is in part due to the control authority given to the policy, which can be seen as a relaxation of constraints on the energy added to the system relative to the system energy scale. The authors expect that the previously mentioned modifications to the cost functionals to penalize highly impulsive control signals will dramatically effect the resulting control policy solution, and in general can be tailored to the control problem and its constraints.

In contrast, the benchmark stabilizing control solution [71] is composed of conditions on the state, and ultimately injects more energy into the system than the QGASS solution, yet has a less impulsive solution, with consistent control effort that does not vanish like the QGASS solution in the prescribed time window.

[1] T. Schulte-Herbrüggen, A. Spörl, N. Khaneja, and S. J. Glaser, Phys. Rev. A **72**, 042331 (2005).

[2] A. Spörl, T. Schulte-Herbrüggen, S. J. Glaser, V. Bergholm, M. J. Storcz, J. Ferber, and F. K. Wilhelm, Phys. Rev. A **75**, 012302 (2007).

[3] J. Chan, T. M. Alegre, A. H. Safavi-Naeini, J. T. Hill, A. Krause, S. Gröblacher, M. Aspelmeyer, and O. Painter, Nature (London) **478**, 89 (2011).

[4] Y. Maday and G. Turinici, J. Chem. Phys. **118**, 8191 (2003).

[5] N. Khaneja, T. Reiss, B. Luy, and S. J. Glaser, J. Magn. Reson. **162**, 311 (2003).

[6] N. Khaneja, T. Reiss, C. Kehlet, T. Schulte-Herbrüggen, and S. J. Glaser, J. Magn. Reson. **172**, 296 (2005).

[7] M. Shapiro and P. Brumer, *Quantum Control of Molecular Processes* (John Wiley & Sons, 2012), https://onlinelibrary.wiley.com/doi/book/10.1002/9783527639700.

[8] U. Boscain, G. Charlot, J.-P. Gauthier, S. Guérin, and H.-R. Jauslin, J. Math. Phys. **43**, 2107 (2002).

[9] U. Boscain, T. Chambrion, and G. Charlot, arXiv preprint quant-ph/0409022 (2004), https://www.aimsciences.org/article/doi/10.3934/dcdsb.2005.5.957.

[10] U. Boscain and P. Mason, J. Math. Phys. **47**, 062101 (2006).

[11] A. Carlini, A. Hosoya, T. Koike, and Y. Okudaira, Phys. Rev. Lett. **96**, 060503 (2006).

[12] P. Salamon, K. H. Hoffmann, and A. Tsirlin, Appl. Math. Lett. **25**, 1263 (2012).

[13] U. Boscain, F. Grönberg, R. Long, and H. Rabitz, J. Math. Phys. **55**, 062106 (2014).

[14] R. Romano, Phys. Rev. A **90**, 062302 (2014).

[15] F. Albertini and D. D'Alessandro, Automatica **74**, 55 (2016).

[16] T. Szakács, B. Amstrup, P. Gross, R. Kosloff, H. Rabitz, and A. Lörincz, Phys. Rev. A **50**, 2540 (1994).

[17] I. R. Sola, J. Santamaria, and D. J. Tannor, J. Phys. Chem. A **102**, 4301 (1998).

[18] A. Bartana, R. Kosloff, and D. J. Tannor, Chem. Phys. **267**, 195 (2001).

[19] C. P. Koch, J. P. Palao, R. Kosloff, and F. Masnou-Seeuws, Phys. Rev. A **70**, 013402 (2004).

[20] J. P. Palao, R. Kosloff, and C. P. Koch, Phys. Rev. A **77**, 063412 (2008).

[21] T. Caneva, M. Murphy, T. Calarco, R. Fazio, S. Montangero, V. Giovannetti, and G. E. Santoro, Phys. Rev. Lett. **103**, 240501 (2009).

[22] R. Eitan, M. Mundt, and D. J. Tannor, Phys. Rev. A **83**, 053426 (2011).

[23] P. Kumar, S. A. Malinovskaya, and V. S. Malinovsky, J. Phys. B: At., Mol. Opt. Phys. **44**, 154010 (2011).

[24] J. P. Palao, D. M. Reich, and C. P. Koch, Phys. Rev. A **88**, 053409 (2013).

[25] N. M., C. Koch, and D. Sugny, J. Mod. Opt. **61**, 857 (2014).

[26] G. Jäger, D. M. Reich, M. H. Goerz, C. P. Koch, and U. Hohenester, Phys. Rev. A **90**, 033628 (2014).

[27] D. Dong, C. Chen, T.-J. Tarn, A. Pechen, and H. Rabitz, IEEE Trans. Syst. Man Cybern. Part B: Cybern. **38**, 957 (2008).

[28] M. Y. Niu, S. Boixo, V. N. Smelyanskiy, and H. Neven, npj Quantum Inf. **5**, 1 (2019).

[29] J. Zhang, Y.-x. Liu, R.-B. Wu, K. Jacobs, and F. Nori, Phys. Rep. **679**, 1 (2017).

[30] A. C. Doherty, S. Habib, K. Jacobs, H. Mabuchi, and S. M. Tan, Phys. Rev. A **62**, 012105 (2000).

[31] K. Jacobs, Phys. Rev. A **67**, 030301(R) (2003).

[32] C. Sayrin, I. Dotsenko, X. Zhou, B. Peaudecerf, T. Rybarczyk, S. Gleyzes, P. Rouchon, M. Mirrahimi, H. Amini, M. Brune *et al.*, Nature (London) **477**, 73 (2011).

[33] D. Lu, K. Li, J. Li, H. Katiyar, A. J. Park, G. Feng, T. Xin, H. Li, G. Long, A. Brodutch *et al.*, npj Quantum Inf. **3**, 45 (2017).

[34] G. Cardona, A. Sarlette, and P. Rouchon, in *2018 IEEE Conference on Decision and Control (CDC)* (IEEE, 2018), pp. 6591–6596, https://ieeexplore.ieee.org/abstract/document/8618681.

[35] J. Mulero-Martínez and J. Molina-Vilaplana, J. Math. Phys. **61**, 102203 (2020).

[36] P. Warszawski, H. M. Wiseman, and A. C. Doherty, Phys. Rev. A **102**, 042210 (2020).

[37] W.-L. Ma, S. Puri, R. J. Schoelkopf, M. H. Devoret, S. Girvin, and L. Jiang, Sci. Bull. **66**, 1789 (2021).

[38] Y. Kashiwamura and N. Yamamoto, IFAC-PapersOnLine **50**, 11760 (2017).

[39] M. H. Goerz and K. Jacobs, Quantum Sci. Technol. **3**, 045005 (2018).

[40] M. Heuck, K. Jacobs, and D. R. Englund, Phys. Rev. Lett. **124**, 160501 (2020).

[41] S. Krastanov, K. Jacobs, D. R. Englund, and M. Heuck, Quantum Inform. **8**, 103 (2022).

[42] K. Jacobs, X. Wang, and H. M. Wiseman, New J. Phys. **16**, 073036 (2014).

[43] A. Balouchi and K. Jacobs, Quantum Sci. Technol. **2**, 025001 (2017).

[44] V. Belavkin, Phys. Lett. A **140**, 355 (1989).

[45] V. Belavkin, Rep. Math. Phys. **43**, A405 (1999).

[46] V. P. Belavkin, Found. Phys. **24**, 685 (1994).

[47] G. J. Milburn and D. F. Walls, Phys. Rev. A **28**, 2065 (1983).

[48] M. Brune, S. Haroche, V. Lefevre, J. M. Raimond, and N. Zagury, Phys. Rev. Lett. **65**, 976 (1990).

[49] Y. Takahashi, K. Honda, N. Tanaka, K. Toyoda, K. Ishikawa, and T. Yabuzaki, Phys. Rev. A **60**, 4974 (1999).

[50] A. Lupaşcu, S. Saito, T. Picot, P. De Groot, C. Harmans, and J. Mooij, Nat. Phys. **3**, 119 (2007).

[51] T. Nakajima, A. Noiri, J. Yoneda, M. R. Delbecq, P. Stano, T. Otsuka, K. Takeda, S. Amaha, G. Allison, K. Kawasaki *et al.*, Nat. Nanotechnol. **14**, 555 (2019).

[52] H. M. Wiseman and G. J. Milburn, Phys. Rev. Lett. **70**, 548 (1993).

[53] M. Abdelhafez, D. I. Schuster, and J. Koch, Phys. Rev. A **99**, 052327 (2019).

[54] C. Ahn, A. C. Doherty, and A. J. Landahl, Phys. Rev. A **65**, 042301 (2002).

[55] A. Barchielli and M. Gregoratti, *Quantum Trajectories and Measurements in Continuous Time: The Diffusive Case* (Springer, 2009), Vol. 782, https://link.springer.com/book/10.1007/978-3-642-01298-3.

[56] H. J. Kappen, J. Stat. Mech.: Theory Exp. (2005) P11011.

[57] H. J. Kappen and H. C. Ruiz, J. Stat. Phys. 162, 1244 (2016), ISSN 1572-9613, https://doi.org/10.1007/s10955-016-1446-7.

[58] G. Williams, A. Aldrich, and E. Theodorou, arXiv:1509.01149, https://arc.aiaa.org/doi/full/10.2514/1.G001921.

[59] G. Williams, A. Aldrich, and E. A. Theodorou, J. Guid. Control. Dyn. **40**, 344 (2017).

[60] M. A. Pereira and Z. Wang, in *Robotics: Science and Systems* (2019), http://m.roboticsproceedings.org/rss15/p70.html.

[61] Z. Wang, K. Lee, M. A. Pereira, I. Exarchos, and E. A. Theodorou, in *2019 IEEE 58th Conference on Decision and Control (CDC)* (IEEE, 2019), pp. 6807–6814, http://www.roboticsproceedings.org/rss16/p049.html.

[62] E. N. Evans, A. P. Kendall, G. I. Boutselis, and E. A. Theodorou, in *Proceedings of Robotics: Science and Systems, Corvalis, Oregon, USA* (2020), http://www.roboticsproceedings.org/rss16/p049.html.

[63] E. N. Evans, A. P. Kendall, and E. A. Theodorou, Auton. Rob. **46**, 283 (2022).

[64] E. Zhou and J. Hu, IEEE Trans. Autom. Control **59**, 1818 (2014).

[65] E. Zhou and S. Bhatnagar, INFORMS J. Comput. **30**, 154 (2018).

[66] X. Chen, E. Zhou, and J. Hu, IISE Trans. **50**, 789 (2018).

[67] H. Zhu, J. Hale, and E. Zhou, J. Global Optim. **70**, 783 (2018).

[68] G. I. Boutselis, Z. Wang, and E. A. Theodorou, in *2020 IEEE International Conference on Robotics and Automation (ICRA)* (IEEE, 2020), pp. 2522–2528, https://doi.org/10.1109/ICRA40945.2020.9197284.

[69] Z. Wang, O. So, K. Lee, and E. A. Theodorou, in *Learning for Dynamics and Control* (PMLR, 2021), pp. 510–522, https://proceedings.mlr.press/v144/wang21b.html.

[70] I. Exarchos, M. A. Pereira, Z. Wang, and E. A. Theodorou, International Conference on Learning Representations (2020), https://openreview.net/forum?id=Iw4ZGwenbXf.

[71] M. Mirrahimi and R. Van Handel, SIAM J. Control Optim. **46**, 445 (2007).

[72] J. R. Johansson, P. D. Nation, and F. Nori, Comput. Phys. Commun. **183**, 1760 (2012).

[73] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga *et al.*, Adv. Neural Inf. Process. Syst. **32**, 8026 (2019).

[74] Y. A. LeCun, L. Bottou, G. B. Orr, and K.-R. Müller, in *Neural networks: Tricks of the trade* (Springer, 2012), pp. 9–48, https://doi.org/10.1007/978-3-642-35289-8_3.

[75] O. V. Morzhin and A. N. Pechen, Russ. Math. Surv. **74**, 851 (2019).
[76] K. Jacobs, arXiv:1304.0819.
[77] www.smartscholarship.org.
[78] W. Verstraelen and M. Wouters, Appl. Sci. **8**, 1427 (2018).
[79] Z. Wang, O. So, J. Gibson, B. Vlahov, M. Gandhi, G.-H. Liu ,and E. A. Theodorou, Rob.: Sci. Syst. (2018), http://www.roboticsproceedings.org/rss17/p073.html.
[80] R. T. Q. Chen, Y. Rubanova, J. Bettencourt, and D. K. Duvenaud, Advances in Neural Information Processing Systems 31 (2018), https://proceedings.neurips.cc/paper/2018/hash/69386f6bb1dfed68692a24c8686939b9-Abstract.html.
[81] G. Williams, N. Wagener, B. Goldfain, P. Drews, J. M. Rehg, B. Boots, and E. A. Theodorou, *International Conference on Robotics and Automation (ICRA)* (IEEE, 2017), pp. 1714–1721, https://ieeexplore.ieee.org/abstract/document/7989202. .