

Quantum multiobservable control

Raj Chakrabarti,^{*} Rebing Wu, and Herschel Rabitz

Department of Chemistry, Princeton University, Princeton, New Jersey 08544, USA

(Received 27 April 2008; published 30 June 2008)

We present deterministic algorithms for the simultaneous control of an arbitrary number of quantum observables. Unlike optimal control approaches based on cost function optimization, quantum multiobservable tracking control (MOTC) is capable of tracking predetermined homotopic trajectories to target expectation values in the space of multiobservables. The convergence of these algorithms is facilitated by the favorable critical topology of quantum control landscapes. Fundamental properties of quantum multiobservable control landscapes, including the MOTC Gramian matrix, are introduced. The effects of multiple control objectives on the structure and complexity of optimal fields are examined. With minor modifications, the techniques described herein can be applied to general quantum multiobjective control problems.

DOI: [10.1103/PhysRevA.77.063425](https://doi.org/10.1103/PhysRevA.77.063425)

PACS number(s): 32.80.Qk, 03.67.-a, 02.30.Yy

I. INTRODUCTION

The optimal control of quantum dynamics is receiving increasing attention due to widespread success in laboratory experiments and numerical simulations across a broad scope of systems. With these promising results it becomes imperative to understand the reasons for success and to develop more efficient algorithms that can increase objective yields, especially for more complex objectives.

Quantum optimal control (QC) problems studied to date fall into two major classes: (1) control of the expectation values of single quantum observables or pure states [1–4]; (2) control of quantum dynamical transformations [5,6]. The former has been implemented in both simulations and experiments, whereas the latter has been approached predominantly through numerical studies, due to the expense of quantum process tomography. An important third class of QC problems, which lies between the latter two problem types, is the control of arbitrary numbers of quantum observables. To date, few effective techniques—either experimental or numerical—have been reported for multiobservable quantum control [7,8].

The typical experimental approach to controlling single quantum observables is to maximize (or minimize) a control objective functional, such as the expectation value of an observable operator, using stochastic search algorithms. A common technique is to randomly sample single-observable expectation values at various points over the control landscape and use genetic algorithms (GAs) to update the control field [9]. However, the utility of such stochastic search algorithms needs careful consideration for more complex QC problems like multiobservable control. A recent work [8] examined the application of multiobjective evolutionary algorithms (MOEAs), which do not rely on a control objective functional, to a two-observable quantum optimal control problem. While the MOEAs displayed promising improvements over GAs, they scale more poorly than objective function-based algorithms [10]. Moreover, both GAs and MOEAs are limited in their efficiency by the fact that they do not make

use of prior knowledge pertaining to the structure of the search landscape. As such, there remains a need for the development of both numerical and experimental control strategies tailored for the problem of multiobservable quantum control.

Recently, significant strides have been made toward establishing a foundation for the systematic development of efficient optimal control algorithms for more complex QC problems based on the observation that the landscape traversed by search algorithms in the optimization of quantum controls is not arbitrarily complicated, but rather possesses an analytical structure originating in the geometry of quantum mechanics [11]. Prior work established important features of these landscapes, in particular their critical topologies [3,12]. The critical points of a control landscape correspond to locally optimal solutions to the control problem. The landscapes corresponding to problems 1 and 2 were shown to have a set of critical submanifolds of measure zero in the search domain, and to be essentially devoid of local traps, i.e., the vast majority of local suboptima are saddles, facilitating the convergence of local search algorithms.

The underlying monotonicity of quantum control landscapes has increased interest in deterministic quantum control optimization algorithms. A recent experimental study [13] demonstrated at least a twofold improvement in optimization efficiency through the use of local gradient algorithms rather than a GA. Such experimental gradient-based search algorithms, enabled by the favorable critical topology of QC landscapes, may in fact be essential for high-precision quantum multiobservable control. Still, any optimal control theory (OCT) strategy based on optimizing a control cost functional may not be ideal for multiobservable control problems, where the most effective combination of observable expectation values is not always clear at the outset. A more powerful deterministic algorithm would offer the ability to explore arbitrary trajectories across the control landscape to identify desirable solutions.

In order to address this need, we develop in this paper the general theory of quantum multiobservable tracking control (MOTC), a control strategy that seeks to drive a quantum system along homotopic paths to desired multiobservable expectation values. This algorithm is motivated by the so-called continuation methodology for multiobjective optimi-

^{*}rajchak@princeton.edu

zation [14], which is a vector optimization technique based on the principles of differential geometry that serves as an alternative to multiobjective stochastic optimization. The paper is organized as follows. In Sec. II, we derive analytical results characterizing the gradient flows of quantum multiobservable control cost functionals, examining the factors that govern the convergence of these flows, and showing that they follow paths (in the unitary group) that are highly Hamiltonian dependent. In Sec. III, we review the theory of unitary propagator tracking control, highlighting its algorithmic advantages compared to scalar objective optimization, as well as its stringent requirements for the regularity of control fields. In Secs. IV and V, we develop the related theory of multiobservable tracking control and introduce the MOTC Gramian matrix, which characterizes the ability of the search algorithm to move in arbitrary directions in multiobservable space from a given point on the control landscape. In Sec. VI, we describe the methods employed in the numerical implementation of multiobservable tracking and then, in Sec. VII, we present numerical illustrations of MOTC in the case of several model systems, examining the effect of multiple observable constraints on the structure and complexity of the optimal controls. Finally, in Sec. VIII, we draw general conclusions and discuss future directions.

II. QUANTUM MULTI-OBSERVABLE CONTROL GRADIENT FLOWS

A generic quantum optimal control problem can be written [11]

$$\max_{\varepsilon(t)} \Phi(U(T)) \quad (1)$$

where $U(T)$ is an implicit functional of $\varepsilon(t)$ via the Schrödinger equation for the unitary propagator,

$$\frac{dU(t)}{dt} = -\frac{i}{\hbar} H(\varepsilon(t)) U(t), U(0) = I_N.$$

Here H denotes the total Hamiltonian and $\varepsilon(t)$ is the time-dependent control field. Solutions to the optimal control problem correspond to $\frac{\delta\Phi}{\delta\varepsilon(t)} = 0$ for all $t \in [0, T]$. The objective function Φ can take various forms. The most common form of Φ is the expectation value of an observable of the system:

$$\Phi(U(T)) = \text{Tr}[U(T)\rho(0)U^\dagger(T)\Theta], \quad (2)$$

where $\rho(0)$ is the initial density matrix of the system and Θ is the Hermitian observable operator whose expectation value is to be maximized [15].

A natural objective function for multiobservable control is a positively weighted convex sum of individual observable objectives, i.e.,

$$\Phi_M(U) = \sum_{k=1}^m \alpha_k \Phi_k(U), \quad \alpha_k > 0, \quad (3)$$

where $\Phi_k(U) = \text{Tr}[U\rho(0)U^\dagger\Theta_k]$, $k=1, 2, \dots, m$. The goal of a multiobjective optimization problem may be to maximize the expectation values of all observables, i.e.,

$$\vec{\Phi}(U(T)) = \{\Phi_1(U(T)), \dots, \Phi_M(U(T))\}.$$

Alternatively, the goal may be to target observable expectation values χ_k , $k=1, \dots, m$, in which case the objective function (3) can be replaced by

$$\Phi'_M(U) = \sum_{k=1}^m \alpha_k |\Phi_k(U) - \chi_k|^2, \quad \alpha_k > 0. \quad (4)$$

In this section, we examine the factors that affect the efficiency of algorithms that optimize scalar objective functions of the form (2) or (3), with a specific focus on gradient algorithms. We begin by writing expressions for the gradients of these functionals. An infinitesimal functional change in the Hamiltonian $\delta H(t)$ produces an infinitesimal change in the dynamical propagator $U(T)$ as follows:

$$\delta U(T) = -\frac{i}{\hbar} \int_0^T U(T) U^\dagger(t) \delta H(t) U(t) dt. \quad (5)$$

The corresponding change in Φ is then given by

$$\delta\Phi = -\frac{i}{\hbar} \int_0^T \text{Tr}\{[\Theta(T), U^\dagger(t) \delta H(t) U(t)] \rho(0)\} dt,$$

where $\Theta(T) \equiv U^\dagger(T)\Theta U(T)$. In the special case of the electric dipole formulation, the Hamiltonian becomes

$$H(t) = H_0 - \mu\varepsilon(t) \quad (6)$$

where H_0 is the internal Hamiltonian of the system and μ is its electric dipole operator. In this situation, the gradient of Φ is [2]:

$$\frac{\delta\Phi}{\delta\varepsilon(t)} = -\frac{i}{\hbar} \text{Tr}\{[\Theta(T), \mu(t)] \rho(0)\}, \quad (7)$$

where $\mu(t) = U^\dagger(t)\mu U(t)$, and the gradient for Φ_M is

$$\frac{\delta\Phi_M}{\delta\varepsilon(t)} = -\frac{i}{\hbar} \sum_{k=1}^m \alpha_k \text{Tr}\{[\Theta_k(T), \mu(t)] \rho(0)\}. \quad (8)$$

The flow trajectories followed by these gradients are the solutions to the differential equation

$$\frac{\partial \varepsilon_s(t)}{\partial s} = \gamma \frac{\delta\Phi_{(M)}}{\delta\varepsilon_s(t)}, \quad (9)$$

where $s > 0$ is a continuous variable parametrizing the algorithmic time evolution of the search trajectory, and γ is an arbitrary positive constant that we will set to 1. Prior work [11] demonstrated that, under the assumption of linear independence of the elements of the time-evolved dipole operator $\mu(t)$, the landscape for the objective function (2) contains no local maxima, thus ensuring that the gradient flow (9) cannot be trapped. Furthermore, it was shown that the critical set of this objective function consists of submanifolds of Lebesgue measure zero in $\mathcal{U}(N)$ —indicating that the likelihood of encountering a suboptimal critical point is essentially null. Since Eq. (8) is the gradient of the expectation value of a single observable $\Theta_M = \sum_{k=1}^m \alpha_k \Theta_k$, it follows that its flow will also converge to the global optimum of the objective func-

tion and share the above favorable landscape features. These features indicate that gradient-based algorithms may also be effective for multiobservable control optimization.

The development of such deterministic search algorithms for quantum multiobservable control is especially important because of the aforementioned difficulties in sampling multiobjective landscapes with stochastic techniques. However, there are two characteristics of the simple gradient flows (9) that could be improved to render them more efficient in searching multiobservable control landscapes. First, the convergence rate of gradient flow control optimization is highly Hamiltonian dependent. To explicitly isolate the Hamiltonian-dependent contribution to the search dynamics, consider first the gradient flow of Φ_M on the unitary group $\mathcal{U}(N)$, which is given by [16]

$$\frac{dV_s}{ds} = \nabla \Phi_M(V_s) = \sum_{k=1}^m \alpha_k [\Theta_k, V_s \rho(0) V_s^\dagger] V_s \quad (10)$$

$$= [\Theta_M, V_s \rho(0) V_s^\dagger] V_s, \quad (11)$$

with $V \in \mathcal{U}(N)$. $\nabla \Phi_M(\cdot)$ denotes the gradient of the objective function on $\mathcal{U}(N)$, where the Riemannian metric is defined by the inner product

$$\langle X, Y \rangle \equiv \text{Tr}(X^\dagger Y)$$

for any X and Y in the tangent space $T_V \mathcal{U} \equiv \{VB|B^\dagger = -B\}$ of $\mathcal{U}(N)$ at V .

We are interested in the relationship between the paths followed by the gradient flow (9) on $\varepsilon(t)$ and that (10) on $\mathcal{U}(N)$. The gradient function on $\varepsilon_s(t)$ is related to the gradient on $\mathcal{U}(N)$ through

$$\frac{\delta \Phi_M}{\delta \varepsilon_s(t)} = \text{Tr} \left(\nabla \Phi_M(U_s(T)) \frac{\delta U_s^\dagger(T)}{\delta \varepsilon_s(t)} \right) \quad (12)$$

$$= \sum_{p,q=1}^N \frac{\partial \Phi_M}{\partial (U_s(T))_{pq}} \frac{\delta (U_s(T))_{pq}^*}{\delta \varepsilon_s(t)}. \quad (13)$$

Now suppose that we have the gradient flow on $\varepsilon_s(t)$ that follows (9) and let $U_s(T)$ be the projected trajectory on the unitary group $\mathcal{U}(N)$ of system propagators at time T , driven by $\varepsilon_s(t)$. The algorithmic time derivative of $U_s(T)$ is then

$$\frac{d(U_s(T))_{ij}}{ds} = \int_0^T \frac{\delta (U_s(T))_{ij}}{\delta \varepsilon_s(t)} \frac{\partial \varepsilon_s(t)}{\partial s} dt \quad (14)$$

which, combined with (9) and (11), gives

$$\frac{d(U_s(T))_{ij}}{ds} = \int_0^T \frac{\delta (U_s(T))_{ij}}{\delta \varepsilon_s(t)} \sum_{p,q=1}^N \frac{\partial \Phi_M}{\partial (U_s(T))_{pq}} \frac{\delta (U_s(T))_{pq}^*}{\delta \varepsilon_s(t)} dt. \quad (15)$$

To obtain expressions for the $\frac{\delta (U_s(T))_{ij}}{\delta \varepsilon_s(t)}$, note that in the electric dipole formulation Eq. (5) becomes

$$\delta U(T) = -\frac{i}{\hbar} \int_0^T U(T) U^\dagger(t) \mu U(t) \delta \varepsilon(t) dt. \quad (16)$$

Taking the derivative of this expression with respect to a functional change in the control field, we get

$$\frac{\delta U(T)}{\delta \varepsilon(t)} = -\frac{i}{\hbar} U(T) U^\dagger(t) \mu U(t) =: -\frac{i}{\hbar} U(T) \mu(t).$$

It is convenient to write Eq. (15) in vector form, replacing the $N \times N$ matrix $U_s(T)$ with the N^2 -dimensional vector \mathbf{u}_s :

$$\frac{d\mathbf{u}_s}{ds} = \left(\int_0^T \frac{\delta \mathbf{u}_s}{\delta \varepsilon_s(t)} \frac{\delta \mathbf{u}_s^\dagger}{\delta \varepsilon_s(t)} dt \right) \nabla \Phi_M(\mathbf{u}_s) =: F[\varepsilon_s(t)] \nabla \Phi_M(\mathbf{u}_s), \quad (17)$$

where the elements of the matrix F are

$$F_{ij,pq} = - \int_0^T (U_s(T) \mu_s(t))_{ij} (\mu_s(t) U_s^\dagger(T))_{pq} dt.$$

This relation implies that the variation of the propagator in $\mathcal{U}(N)$ caused by following the gradient flow in the space of control fields is Hamiltonian dependent, with the influence of the Hamiltonian completely contained in the N^2 -dimensional positive-semidefinite, symmetric matrix $F[\varepsilon_s(t)]$. We will make further use of this decomposition in the next section.

A second drawback to using gradient flows to search quantum multiobservable control landscapes is that their convergence rate depends on the properties of the observables Θ_k and the initial density matrix $\rho(0)$. This effect is purely kinematic and does not depend on the Hamiltonian. In [17] we explicitly integrate the kinematic gradient flow (11) in the special case that $\rho(0)$ is a pure state, and show that it follows a convoluted path in $\mathcal{U}(N)$. In general, the kinematic flow evolves on the interior of a polytope whose dimension (and the mean path length of the flow trajectory) rises with rank and eigenvalue nondegeneracy in $\rho(0)$ [17]. Moreover, it can be shown [17] that each term in the multiobservable gradient (8) can be expanded on a “natural” set of basis functions consisting of linear combinations of matrix elements of the time-evolved dipole operator $\mu(t)$; the dimension of this basis is

$$D = N^2 - (N-n)^2 - \sum_{i=1}^r n_i^2 = n(2N-n) - \sum_{i=1}^r n_i^2, \quad (18)$$

where n is the rank of $\rho(0)$ and n_i denotes the degeneracy of the i th distinct eigenvalue of $\rho(0)$ (out of the total r such distinct eigenvalues). D also increases with the rank and eigenvalue nondegeneracy of $\rho(0)$.

Gradient flow control optimization is thus Hamiltonian dependent and decreases in efficiency for cases where $\rho(0)$ and the observables Θ_k are nondegenerate and of high rank. Despite these drawbacks to gradient flow sampling of multiobservable control landscapes, more sophisticated gradient-based algorithms may offer a significant advantage over stochastic search, due to the favorable critical topology of these landscapes. We consider these algorithms below.

III. UNITARY MATRIX FLOW TRACKING

The symmetric, positive semidefinite matrix $F[\varepsilon_s(t)]$ in Eq. (17) above indicates that the convergence time for local gradient-based OCT algorithms may vary greatly as a function of the Hamiltonian of the system. Given the decomposition of the gradient into Hamiltonian-dependent and Hamiltonian-independent parts, the natural question arises as to whether the Hamiltonian-dependent part can be suppressed to produce an algorithm whose convergence time will be dictated by that of the unitary gradient flow (or a suitable kinematic analog), irrespective of the system Hamiltonian.

In order for the projected flow from $\varepsilon_s(t)$ onto $U_s(T)$ to match the path followed by the gradient flow (10), the quantity $\frac{\partial \varepsilon_s(t)}{\partial s}$ that corresponds to movement in each step must satisfy a matrix integral equation

$$\frac{dU_s(T)}{ds} = \int_0^T \frac{\delta U_s(T)}{\delta \varepsilon_s(t)} \frac{\partial \varepsilon_s(t)}{\partial s} dt = \nabla \Phi_M(U_s(T)). \quad (19)$$

In the dipole formulation, this relation becomes

$$\int_0^T \mu_s(t) \frac{\partial \varepsilon_s(t)}{\partial s} dt = i\hbar U_s^\dagger(T) \nabla \Phi_M(U_s(T)), \quad (20)$$

where, as in the previous section, $\mu_s(t) \equiv U_s^\dagger(t) \mu U_s(t)$. This is a Fredholm integral equation of the first kind [18] for $\frac{\partial \varepsilon_s(t)}{\partial s}$, given $\varepsilon_s(t)$ at s and all $t \in [0, T]$. When Φ_M takes the form in Eq. (3), we have

$$\int_0^T \mu_s(t) \frac{\partial \varepsilon_s(t)}{\partial s} dt = -i\hbar \sum_{k=1}^m \alpha_k [\rho(0), U_s^\dagger(T) \Theta_k U_s(T)].$$

On the basis of eigenstates, the matrix integral equation (20) is written

$$\int_0^T \langle i | \mu_s(t) | j \rangle \frac{\partial \varepsilon_s(t)}{\partial s} dt = i\hbar \langle i | U_s^\dagger(T) \nabla \Phi_M(U_s(T)) | j \rangle. \quad (21)$$

The relation (21) is underspecified with respect to $\varepsilon_s(t)$, indicating that the integral equation possesses a family of solutions. To solve it, we must convert it to an explicit initial value problem for $\varepsilon_s(t)$ given $\varepsilon_0(t)$. We first expand the partial derivative as

$$\frac{\partial \varepsilon_s(t)}{\partial s} = \sum_{i,j=1}^N x_s^{ij} \langle i | \mu_s(t) | j \rangle + f_s(t) \quad (22)$$

on the basis of functions $\langle i | \mu_s(t) | j \rangle$, where the “free function” $f_s(t)$ contains the additional linearly independent degrees of freedom. Inserting this expansion into Eq. (21) produces

$$\begin{aligned} & \sum_{p,q=1}^N x_s^{pq} \int_0^T \langle i | \mu_s(t) | j \rangle \langle p | \mu_s(t) | q \rangle dt \\ & = \langle i | \Delta_s | j \rangle - \int_0^T f_s(t) \langle i | \mu_s(t) | j \rangle dt, \end{aligned}$$

where $\Delta_s = i\hbar U_s^\dagger(T) \nabla \Phi_M(U_s(T))$. For more general target

tracks Q_s in $\mathcal{U}(N)$, we have $\Delta_s = \frac{dQ_s}{ds}$. If we denote the Gramian matrix G_s as

$$(G_s)_{ij,pq} = \int_0^T \langle i | \mu_s(t) | j \rangle \langle p | \mu_s(t) | q \rangle dt, \quad (23)$$

we can solve for the coefficients $\vec{x} = (x_s^{11}, \dots, x_s^{1N}; \dots; x_s^{N1}, \dots, x_s^{NN})^T$ as

$$\vec{x} = G_s^{-1} v \left(\frac{dQ_s}{ds} - \int_0^T \mu_s(t') f_s(t') dt' \right),$$

where the operator $v(\cdot)$ vectorizes its matrix argument, provided that G_s is invertible. We then obtain the following initial value problem for $\varepsilon_s(t)$, the algorithmic evolution of the control field along the track:

$$\frac{\partial \varepsilon_s(t)}{\partial s} = f_s(t) + v \left(\frac{dQ_s}{ds} - \int_0^T \mu_s(t') f_s(t') dt' \right)^T G_s^{-1} v(\mu_s(t)). \quad (24)$$

Each “free function” f_s corresponds to a unique algorithmic step in $\varepsilon(\cdot)$; modulating this function allows for systematic exploration of the set of functions $\varepsilon_s(t)$ that are compatible with the gradient step on $\mathcal{U}(N)$ [19].

Solving this set of N^4 scalar differential equations requires that the $N^2 \times N^2$ matrix G_s defined by (23) is invertible, which is equivalent to the claim that the map between control fields and unitary propagators is locally surjective in a sufficiently small neighborhood of U_s . Control fields at which G_s is singular correspond to so-called *singular extremal* solutions to the control problem, in order to contrast them with *regular extremals* [20]. Even if G_s is invertible, it is possible that it is nearly singular, resulting in large numerical errors during the solution to the differential equation. It is convenient to assess the nearness to singularity of G_s by means of its condition number, namely, the ratio of its largest to its smallest singular value.

Although the gradient function $\frac{\partial \Phi_M}{\partial \varepsilon_s(t)}$ is always locally the direction of fastest increase in the objective function at $\varepsilon_s(t)$, the path $\varepsilon_s(t)$ (parametrized by s) derived from following this gradient has no universal (Hamiltonian-independent) global geometry, since Φ_M is not explicitly a function of $\varepsilon_s(t)$. It is known [15] that this path will not encounter any traps during the search, but beyond this, the geometry can be expected to be rugged. Unlike the dynamical gradient flow (9), the algorithmic flow (24) follows the gradient flow on $\mathcal{U}(N)$. This flow respects the geometric formulation of the optimal control objective function in terms of $U_s(T)$ rather than directly in terms of $\varepsilon_s(t)$. The functions $\mu_s(t)$ contain all relevant information about the quantum dynamics, whereas the functions $\frac{dQ_s}{ds}$ contain complete information about the geometry of the kinematic search space. The N^2 functions $\langle i | \mu_s(t) | j \rangle$, $1 \leq i, j \leq N$, are calculated during the evaluation of $\frac{\partial \Phi_M}{\partial \varepsilon_s(t)}$; hence, the computational overhead incurred by following this flow corresponds to that needed to compute the N^4 elements of G_s and invert the matrix, at each algorithmic time step.

A multitude of other flows $\frac{dQ_s}{ds}$ could be substituted in the right-hand side of Eq. (21). Since we are interested in global optimality, we should choose a flow that follows the shortest possible path from the initial condition to a unitary matrix that maximizes the observable expectation value. It can be shown [15] that a continuous manifold of unitary matrices W maximizes $\Phi_{(M)}$. These unitary matrices can be determined numerically by standard optimization algorithms on the domain of unitary propagators [21]. The shortest length path in $\mathcal{U}(N)$ between the initial guess U_0 and an optimal W is then the geodesic path, which can be parametrized as

$$Q_s = U_0 \exp(iAs) \quad (25)$$

with $A = -i \log(W^\dagger U_0)$, where “log” denotes the complex matrix logarithm with eigenvalues chosen to lie on the principal branch $-\pi < \theta < \pi$. Thus, if we set $\frac{dQ_s}{ds} = A$, the tracking algorithm will attempt to follow the geodesic path.¹

Due to the nonlinearity of the integral equation (19), errors in tracking will inevitably occur, increasing the length of the search trajectory beyond that of the minimal geodesic path. These tracking errors may be ameliorated through the introduction of stabilization terms in Eq. (24) (see Sec. V) or higher-order functional derivatives in Eq. (20).

IV. MULTIOBSERVABLE HOMOTOPY TRACKING CONTROL

As a methodology for multiobservable control, unitary matrix tracking has an advantage over gradient search in that it can directly follow an optimal path in the space of unitary propagators, assuming the map $\varepsilon(t) \mapsto U(T)$ is surjective and the first-order formulation of the tracking Eq. (24) is sufficiently accurate. However, it cannot be implemented experimentally without expensive tomography measurements, and carries a computational overhead that scales exponentially with system size.

We now consider a related, *experimentally implementable* tracking control algorithm—multiobservable homotopy tracking control (MOTC)—that seeks to drive the expectation values of m observable operators $\Theta_1, \dots, \Theta_m$ along a predetermined path \mathbf{w}_s of $\vec{\Phi}$ to desired target values. These trajectories may correspond to expectation value paths mapping to the kinematic gradient flow (10), the geodesic (25), or any other path through multiobservable space. In particular, unlike the gradient flows (9), MOTC may be used to successively drive the expectation values of individual observables to their maxima, while constraining the others to fixed values.

The observables measured at each step can be assumed to be linearly independent without loss of generality. Denote the m scalar functions of algorithmic time (expectation value paths for each observable) by $\Phi_s^1, \dots, \Phi_s^m$. Then, the vector Fredholm integral equation for $\frac{\partial \varepsilon_s(t)}{\partial s}$ analogous to Eq. (19) is given by

¹In the case that the control system evolves on a subgroup of $\mathcal{U}(N)$, e.g., $SU(N)$, the geodesic on that subgroup can be tracked instead.

$$\frac{d\Phi_s^i}{ds} = \int_0^T \frac{\delta \Phi_s^i}{\delta \varepsilon_s(t)} \frac{\partial \varepsilon_s(t)}{\partial s} dt = \frac{dw_s^i}{ds}, \quad 1 \leq i \leq m, \quad (26)$$

where $\Phi_s^i = \text{Tr}[U_s(T)\rho(0)U_s^\dagger(T)\Theta_i]$. As in Sec. III, to solve for the algorithmic flow $\frac{\partial \varepsilon_s(t)}{\partial s}$ that tracks this path, it is necessary to expand it on a basis of independent functions. In this case, we make this expansion on the basis of the independent observable expectation value gradients:

$$\frac{\partial \varepsilon_s(t)}{\partial s} = \sum_{j=1}^m x_s^j \frac{\delta \Phi_s^j}{\delta \varepsilon_s(t)} + f_s(t), \quad (27)$$

where the free function is similar to that in Eq. (22). Inserting the expansion 27 into the integral equation, we have

$$\sum_{j=1}^m x_s^j \int_0^T \frac{\delta \Phi_s^i}{\delta \varepsilon_s(t)} \frac{\delta \Phi_s^j}{\delta \varepsilon_s(t)} dt = \frac{dw_s^i}{ds} - \int_0^T \frac{\delta \Phi_s^i}{\delta \varepsilon_s(t)} f_s(t) dt.$$

Defining the m -dimensional vector $\mathbf{a}_s(t) = (a_s^1(t), \dots, a_s^m(t))^T$ by

$$\begin{aligned} a_s^i(t) &= \frac{\delta \Phi_s^i}{\delta \varepsilon_s(t)} \\ &= -\frac{i}{\hbar} \text{Tr}[U_s^\dagger(T)\mu_s(t)\nabla\Phi^i(U_s(T))], \end{aligned} \quad (28)$$

$$= -\frac{i}{\hbar} \text{Tr}\{\rho(0)[U_s^\dagger(T)\Theta_i U_s(T), \mu_s(t)]\}, \quad (29)$$

and the MOTC Gramian matrix Γ as

$$(\Gamma_s)_{ij} = \int_0^T a_s^i(t) a_s^j(t) dt,$$

we can then solve for the expansion coefficients $\vec{x} = (x_s^1, \dots, x_s^m)^T$ as

$$\vec{x} = \Gamma_s^{-1} \left[\frac{d\mathbf{w}_s}{ds} - \int_0^T \mathbf{a}_s(t') f_s(t') dt' \right],$$

provided that Γ_s is invertible. Returning to the original expansion (27), we obtain the following nonsingular algebraic differential equation for the algorithmic flow of the control field:

$$\frac{\partial \varepsilon_s(t)}{\partial s} = f_s(t) + \left(\frac{d\mathbf{w}_s}{ds} - \int_0^T \mathbf{a}_s(t') f_s(t') dt' \right) \Gamma_s^{-1} \mathbf{a}_s(t). \quad (30)$$

In the special case where only one observable Θ is measured at each algorithmic step, this equation reduces to

$$\frac{\partial \varepsilon_s(t)}{\partial s} = f_s(t) + \frac{a_s(t)}{\gamma_s} \left(\frac{dP_s}{ds} - \int_0^T a_s(t') f_s(t') dt' \right), \quad (31)$$

where P_s is the desired track for $\langle \Theta(T) \rangle$, and $\gamma_s = \int_0^T a_s^2(t) dt$. We note that unitary matrix tracking can also be expressed as a special case of Eq. (30), if the functions Φ^i are taken to be the matrix elements of $U_s(T)$, i.e.,

$$\Phi^{(j-1)N+k}(U_s(T)) = \langle j|U_s(T)|k\rangle, \quad j, k = 1, \dots, N.$$

Then, according to Eq. (28), the N^2 -dimensional complex vector $\mathbf{a}_s(t) = v(\mu_s(t))$ is the vector form of the matrix $\mu_s(t)$:

$$a_s^{(j-1)N+k}(t) = \frac{1}{i\hbar} \langle j|\mu_s(t)|k\rangle, \quad j, k = 1, \dots, N.$$

With this choice, the $N^2 \times N^2$ Gramian matrix Γ is identical to G_s as defined by (23). Indeed, with Φ^i and a^i suitably defined, the MOTC tracking differential equation (30) provides a unified framework for continuation approaches [14] to generic quantum multiobjective control problems.

One disadvantage of multiobservable tracking, compared to local gradient optimization of the objective function Φ_M , is that it encounters singular control fields more frequently. In this case, singularities correspond to the situation that variation of control fields near the control under consideration is not sufficient to produce arbitrary variation of $\tilde{\Phi}(U(T))$ driven by this control. Nonetheless, the likelihood of the matrix Γ_s being ill conditioned—even at singular *extremal* control fields $\varepsilon(t)$, where G is singular—diminishes rapidly with $N^2 - m$ (Sec. VII).

Auxiliary penalties on $\varepsilon_s(t)$, such as practical experimental constraints on the total field fluence, act to decrease the degeneracy in the solutions $\frac{\partial \varepsilon_s(t)}{\partial s}$ to Eq. (26). It can be shown [22] that the choice

$$f_s(t) = -\frac{1}{\eta_s} \varepsilon_s(t) w(t) \quad (32)$$

for the free function in the tracking differential equations, where $w(t) > 0$ is an arbitrary weight function and the η_s term (typically constant) controls numerical instabilities, will determine the $\frac{\partial \varepsilon_s(t)}{\partial s}$ at each algorithmic time step s that minimizes fluence.

V. ERROR CORRECTION

Errors can occur when tracking paths on $\mathcal{U}(N)$ or subspaces of its homogeneous spaces for two reasons. First, the algorithmic steps on these spaces will be first-order approximations to the true increments ($Q_{s_{k+1}} - Q_{s_k}$ or $\mathbf{w}_{s_{k+1}} - \mathbf{w}_{s_k}$) due to discretization error; this error will increase as a function of the curvature of the flow trajectory Q_s or \mathbf{w}_s at algorithmic time s_k . Second, the tracking integral equations are formulated in terms of only the first-order functional derivatives $\frac{\delta U_s(T)}{\delta \varepsilon_s(t)}$ or $\frac{\delta \tilde{\Phi}}{\delta \varepsilon_s(t)}$; the truncation error will depend on the magnitude of higher-order functional derivatives.

In numerical simulations, error-correction methods can be applied to account for these deviations from the track of interest. Generally, we choose an error-correction term \mathbf{c}_s that reflects the difference between the current value of the tracking vector and its target value, such that the tracking differential equation (30) becomes

$$\frac{\partial \varepsilon_s(t)}{\partial s} = f_s(t) + \left(\mathbf{c}_s + \frac{d\mathbf{w}_s}{ds} - \int_0^T \mathbf{a}_s(t') f_s(t') dt' \right)^T \Gamma_s^{-1} \mathbf{a}_s(t). \quad (33)$$

For unitary matrix tracking, one can follow the (minimal-length) geodesic from the real point $U_{s_k}(T)$ to the track point Q_{s_k} [19]. This correction can be implemented through the choice

$$\mathbf{c}_{s_k} = v \left(-\frac{t}{s_{k+1} - s_k} \log[Q_{s_k}^\dagger U_{s_k}(T)] \right)$$

in the discretized version of (33).

For general multiobservable expectation value tracking, the vector space within which $\tilde{\Phi}_s$ resides is not a Lie group, but rather a subspace of a homogeneous space, and consequently it is not as straightforward to apply error-correction algorithms that exploit the curved geometry of the manifold.² In this paper, we take as the correction term a simple scalar multiple of the difference between the actual value and the target track, i.e.,

$$\mathbf{c}_s = \beta(\mathbf{w}_s - \tilde{\Phi}_s), \quad \beta > 0.$$

Note that these methods can in principle also be implemented in an experimental setting. The use of Runge-Kutta (RK) integrators (see below) to solve the MOTC differential equation (33) with error correction often decreases tracking errors considerably, especially for problems involving integration over long s intervals. RK integration with error correction is the method of choice for numerical implementations of MOTC, although it is more difficult to apply experimentally.

VI. NUMERICAL IMPLEMENTATION

For illustrations of multiple observable tracking, a set of $m < N$ commuting operators $\Theta_1, \dots, \Theta_m$ was employed. Θ_1 was a randomly chosen nondegenerate diagonal matrix, and $\Theta_2, \dots, \Theta_m$ were sequential pure state projection operators in the canonical basis. The m -dimensional vector $\tilde{\Phi}_s$ was constructed by taking the trace of the product of each of these observable operators with the time-evolved density matrix. Numerical solution of the tracking differential equations (30) and (31) was carried out as follows. The electric field $\varepsilon_s(t)$ was stored as a $p \times q$ matrix, where p and q are the number of discretization steps of the algorithmic time parameter s and the dynamical time t , respectively (i.e., for each algorithmic step s_k , the field was represented as a q -vector for the purpose of computations). Starting from an initial guess $\varepsilon_{s_0}(t)$ for the control field, the Schrödinger equation was integrated over the interval $[0, T]$ by propagating over each time step, producing the local propagator

²A Riemannian metric on the space of density matrices may be defined [25], but its application to multiobservable error correction when $m < N^2 - 1$ is nontrivial.

$$U(t_{j+1}, t_j) = \exp[-iH_{s_k}(t_j)T/(q-1)].$$

The propagation toolkit [23] was used for this purpose. Local propagators were precalculated via diagonalization of the Hamiltonian matrix (at a cost of N^3), exponentiation of the diagonal elements, and left (right) multiplication of the resulting matrix by the matrix of eigenvectors (transpose of the matrix of eigenvectors). This approach is generally faster than computing the matrix exponential directly. Alternatively, a Runge-Kutta integrator [24] can be employed for time propagation.

The time propagators

$$U(t_j, 0) = U(t_j, t_{j-1}) \cdots U(t_1, 0)$$

computed in step 1 were then used to calculate the time-evolved dipole operators $\mu(t_j) = U^\dagger(t_j, 0)\mu U(t_j, 0)$, which can be represented as a q -dimensional vector of $N \times N$ Hermitian matrices. The Γ_{s_k} and vector \mathbf{a}_{s_k} were then computed by time integration of the dipole functions with an appropriate choice of the function $f_s(t)$ described above; in the present work, f_s was set to either 0 or the expression in (32). For tracking of unitary flows—either the kinematic gradient flow (11) or the geodesic flow (25)—the next point Q_{s_k} on the target unitary track was calculated numerically through $Q_{s_k} = Q_{s_{k-1}} \exp(-i\Delta_{s_k} ds)$, using a matrix exponential routine.

Next, the control field $\varepsilon_{s_k}(t)$ was updated to $\varepsilon_{s_{k+1}}(t)$. This step required inversion of the $N^2 \times N^2$ matrix G_{s_k} or $m \times m$ matrix Γ_{s_k} , which was carried out using lower triangular-upper triangular (LU) decomposition. The quantities $\Gamma_{s_k}^{-1}$, \mathbf{a}_{s_k} , and \mathbf{c}_{s_k} were used to compute the q -dimensional vector $\frac{\partial \varepsilon_{s_k}(t)}{\partial s}$. One of two approaches was used to update the field: (i) a simple linear propagation scheme, i.e.,

$$\varepsilon_{s_{k+1}}(t) = \varepsilon_{s_k}(t) + (s_{k+1} - s_k) \frac{\partial \varepsilon_{s_k}(t)}{\partial s} \Big|_{s=s_k},$$

or (ii) a fourth- or fifth-order Runge-Kutta integrator. The updated control field $\varepsilon_{s_{k+1}}(t)$ was then again used to propagate the Schrödinger equation.

Adaptive control of step size was used to accelerate convergence for both the gradient flow and MOTC algorithms. For the gradient flow, the maximum of the objective function Φ or Φ_M along the direction of the gradient at step s was located by first bracketing the maximum in that direction by parabolic extrapolation, followed by application of Brent's method for inverse parabolic interpolation to pinpoint the maximum. For both MOTC and the gradient flow, the Fehlberg embedded Runge-Kutta integrator (ASRK5) with Cash-Karp parameters [24] was used, which embeds a fourth-order integrator within a fifth-order integrator, and compares the differences between the fourth- and fifth-order estimates to assess truncation error and adjust Δs . To compare the efficiency of MOTC and gradient control algorithms, the minimum step size and error tolerance level in ASRK5 were set to the largest values that permitted stable integration of at least one of the algorithms.

VII. EXAMPLES

As an illustrative example of multiobservable control, we examine the problem of targeting the vector of m observable expectation values associated with the unitary propagator U to which the kinematic gradient of a single observable converges. We seek to track the multiobservable path \mathbf{w}_s of expectation values that corresponds to the geodesic between U_0 and W , i.e., $w_s^k = \text{Tr}\{\rho(0)\exp(-iAs)U_0^\dagger \Theta_k U_0 \exp(iAs)\}$, with $A = -i \ln(W^\dagger U_0)$, $1 \leq k \leq m$, for various numbers of observables m . Although this is a simple incarnation of multiobservable quantum control, it is well suited for illustration of its basic principles, since it deals with a universally applicable control objective, and allows for a systematic study of the effects of the imposition of additional observable objectives on the search dynamics and optimal control fields.

The examples below employ an 11-dimensional Hamiltonian of the form (6), with

$$H_0 = \text{diag}\{0.1, 0.2, \dots, 1.1\}, \quad (34)$$

$$\mu_{ij} = \begin{cases} 1, & i = j, \\ 0.15, & |i - j| = 1, \\ 0.08, & |i - j| = 2, \\ 0 & \text{otherwise.} \end{cases} \quad (35)$$

We first assess the properties of the Gramian matrices G and Γ for such systems.

A. MOTC Gramian matrix estimation

An important consideration in any deterministic multiobservable quantum control optimization is the singularity of control fields applied to the quantum system, where the singularity of the mapping $\varepsilon(t) \mapsto U(T)$ (assessed through the Gramian G) must be distinguished from that of the mapping $\varepsilon(t) \mapsto \vec{\Phi}(U(T))$ (assessed through the Gramian Γ). Nonsingularity of the latter corresponds to the ability to move, through an infinitesimal change of the control field $\varepsilon_s(t) \rightarrow \varepsilon_{s+ds}(t)$, between two infinitesimally close vectors of multiple observable expectation value vectors, $\vec{\Phi} \rightarrow \vec{\Phi} + d\vec{\Phi}$. The Gramian G depends only on H_0 , μ , and T , whereas Γ additionally depends on the eigenvalue spectra of $\rho(0)$ and Θ_k . Experimentally or numerically, an ill-conditioned Gramian matrix will result in large tracking errors for $\varepsilon_s(t)$.

The requirements for a control field to be regular for the mapping $\varepsilon(t) \mapsto U(T)$ are, in general, more stringent than those for the mapping $\varepsilon(t) \mapsto \vec{\Phi}(U(T))$, since multiobservable control requires control of a subset of the parameters of $U(T)$. However, if $\rho(0)$ is rank deficient, the multiobservable Gramian matrix Γ can become more ill conditioned than G , since certain paths $U_s(T)$ cannot be accessed. Figure 1 compares the condition number distributions for G and Γ for various $\rho(0)$, where the number of observables $m=10$, for randomly sampled fields $\varepsilon(t)$ of the form

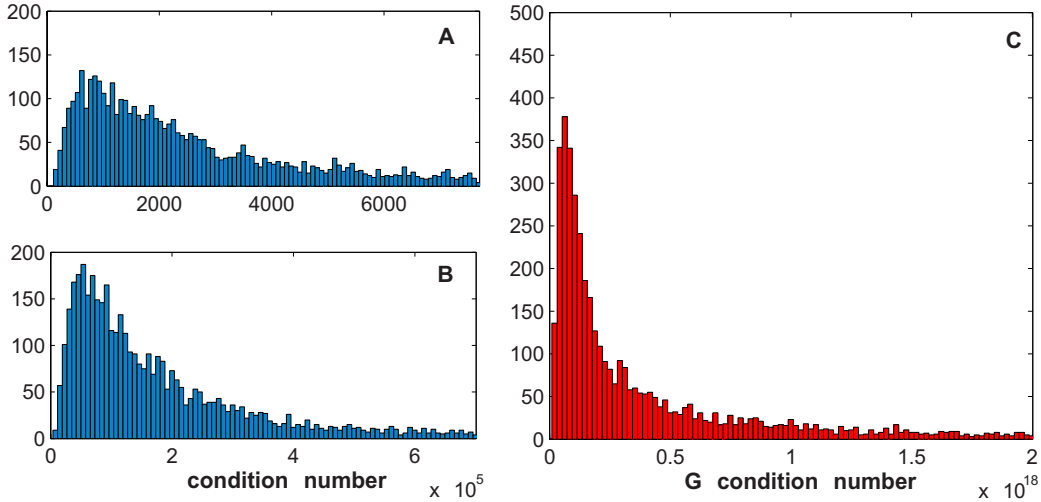


FIG. 1. (Color online) Gramian matrix condition number distributions for the 11-level system in Sec. VII. The amplitudes and phases of the modes of the control field $\varepsilon(t)$ were sampled randomly from the uniform distributions $(0,1]$ and $(0,2\pi]$, respectively, with the mode frequencies tuned to the transition frequencies of the system. MOTC Γ matrix with ten observables and $\rho(0)$ (a) full rank and nondegenerate (thermal) and (b) a pure state. (c) Unitary tracking G matrix.

$$\varepsilon(t) = \sum_{i=1}^N \sum_{j=i+1}^N A_{ij} \sin(\omega_{ij}t + \phi_{ij}), \quad 0 \leq t \leq T, \quad (36)$$

where $\omega_{ij} = |E_i - E_j|$ denote the transition frequencies between energy levels E_i, E_j of H_0 , ϕ_{ij} denotes a phase sampled uniformly within the range $(0, 2\pi]$, and A_{ij} denotes a mode amplitude sampled uniformly within the range $(0, 1]$. The final time T was chosen to be sufficiently large to achieve full controllability over $\mathcal{U}(N)$ at $t=T$ [11].

The absolute magnitudes of the condition numbers for a given $\rho(0)$ spectrum depends on the Hamiltonian, with random, dense H_0 , μ matrices generally exhibiting more well-conditioned Gramian matrix distributions. Typically, a Gramian condition number $C > 10^8$ results in large numerical errors upon inversion, and would be expected to compromise the accuracy of tracking [due to the sparseness of control field increments $\delta\varepsilon(t)$ that are capable of driving the system to the corresponding state].

For small numbers of observables m , rank deficiency in $\rho(0)$ does not shift the condition number distribution for G toward substantially higher values, compared to that of Γ , since the number of parameters of $\rho(T)$ that must be controlled is small. For example, the pure and thermal mixed states in Fig. 1 both have well-conditioned Gramian matrices (Γ), which permit accurate multiobservable tracking (see below).

For a quantum ensemble in thermal equilibrium with a bath at temperature T_e , the eigenvalues are determined by the Boltzmann distribution, i.e.,

$$\lambda_i = \frac{\exp(-E_i/kT_e)}{\sum_{i=1}^N \exp(-E_i/kT_e)}, \quad i = 1, \dots, N,$$

and MOTC can be carried out without the additional overhead of density matrix estimation. Through its effect on the

eigenvalues of $\rho(0)$, the spectrum of H_0 (in particular, the energy-level spacings) plays an important additional role in determining the Gramian matrix condition number distribution for thermally prepared states.

The regularity of control fields generally becomes harder to achieve for larger Hilbert space dimension. For example, for Hamiltonians of the form (6), the mean G condition number rose from $O(10^{18})$ for $N=11$ to $O(10^{19})$ for $N=19$, and the Γ condition number for full rank ρ rose from $O(10^5)$ to $O(10^6)$ for tracking of one complete measurement basis ($m=10$ and 18 , respectively).

The distributions above provide rough estimates for probabilities of encountering singularities during the implementation of the different forms of MOTC examined in the following sections. However, the actual changes in Γ that occur along an optimization trajectory will not be identical to those obtained by random sampling from these distributions, since the frequencies ω of successive control fields in MOTC are not independent random variables.³ As such, the fields $\varepsilon_s(t)$ sampled during MOTC tracking will typically be more regular than those in the distributions displayed above, although the trends will be similar.

B. Multiobservable tracking

According to Eqs. (7) and (18), the efficiency of gradient-flow-based observable maximization decreases with higher rank and nondegeneracy in $\rho(0)$. In these cases, the gradient flow typically follows a longer path in both $\mathcal{U}(N)$ and $\varepsilon(t)$. On the other hand, the unitary path can be constrained more effectively by MOTC in such cases, with the most stringent control possible for full-rank, nondegenerate $\rho(0)$. Figure 2

³The maximum change in Γ (and hence its condition number C) that can occur along an optimization trajectory is bounded due to the fact that the norm of the gradient $\delta\Phi_k/\delta\varepsilon(t)$ is bounded by the norm of the dipole operator $\|\mu\|$ [2].

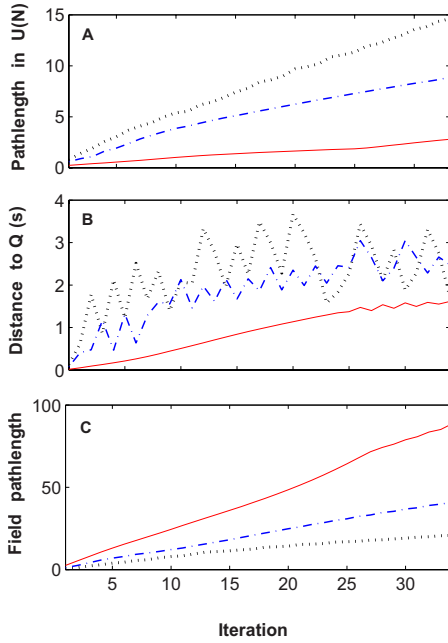


FIG. 2. (Color online) Comparison of MOTC of two (dotted), four (dashed), and ten (solid) observables from a single measurement basis, for 11-level system (34). $\rho(0)$ was of rank 7 with non-degenerate eigenvalues. The target was a unitary propagator that maximized the expectation value of the first observable. (a) Total distance (Frobenius norm) in $\mathcal{U}(N)$ traversed by the dynamical propagator $U(T)$ versus algorithmic step; (b) distance between target Q_s and actual $U_s(T)$ tracks as function of algorithmic step; (c) total distance (Euclidean norm) in $L^2(\mathbb{R})$ between the current field $\varepsilon_s(\cdot)$ and the original field $\varepsilon_0(\cdot)$ along the optimization trajectory.

compares the path lengths in $\mathcal{U}(N)$ of MOTC optimization trajectories in such a case with two, four, and ten observable expectation values tracked along the geodesic to an optimal W precomputed by following the kinematic gradient flow $\frac{dV_s}{ds} = [\Theta_1, V_s \rho(0) V_s^\dagger] V_s$.⁴ The Hamiltonian (34) was employed for these simulations. Across all cases, the Gramian Γ_s was well conditioned at each step of MOTC optimization, as predicted by the Γ condition number distributions above.

As can be seen from the figure, the path length in $\mathcal{U}(N)$ decreases progressively with increasing m ; this path length is in all cases smaller than that of the gradient flow (data not shown). For small m , there are significant stochastic fluctuations in the unitary step size per iteration, which are smoothed out for larger m . By contrast, the Euclidean path length traversed in the space of control fields $\varepsilon(\cdot)$ [assessed in terms of the Euclidean distance $|\varepsilon_{s_1}(\cdot) - \varepsilon_{s_2}(\cdot)|^2 \equiv \sum_i |\varepsilon_{s_1}(t_i) - \varepsilon_{s_2}(t_i)|^2$, where s_1, s_2 label successive points along the optimization trajectory] increases systematically with m , with the smallest change in $\varepsilon(\cdot)$ along the optimization trajectory occurring for the gradient and the greatest change occurring for $m=10$. Almost universally, imposing additional observable tracks increases the field path

⁴Note that, although the tracked unitary path Q_s is significantly shorter than that followed by the dynamical gradient, still shorter paths could be identified by minimizing the distance from W to U_0 while constraining $\langle \Theta \rangle$ to remain at its maximal value.

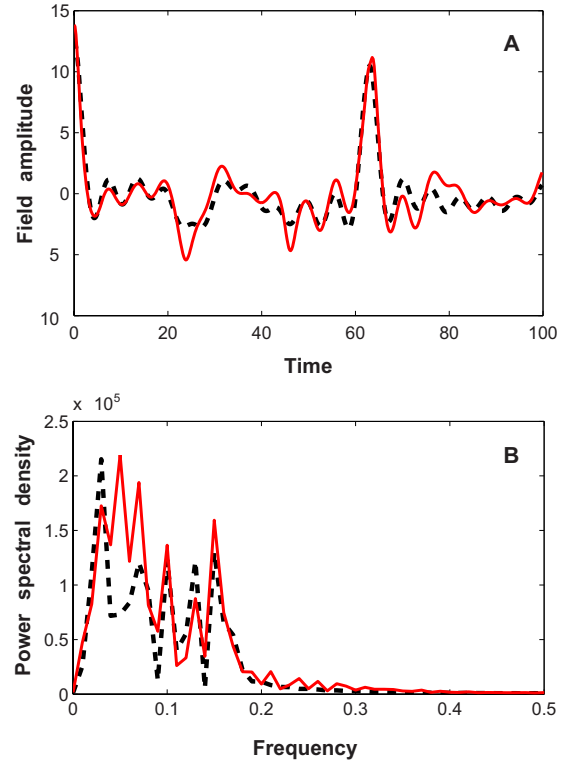


FIG. 3. (Color online) Comparison of optimal fields obtained via $m=2$ (dashed) and 10 (solid) MOTC search algorithms that maximize a single observable Θ . $\rho(0)$ was a thermal mixed state. (a) Optimal control fields for two- and ten-observable tracking; (b) Fourier power spectra of the optimal control fields. Note the increase in high-frequency modes due to the imposition of additional observable objectives. $\varepsilon_0(t)$ was of the form (36).

length and distance between $\varepsilon_0(\cdot)$ and $\varepsilon_s(\cdot)$. Since the MOTC field update (30) is based on projections the multiobservable gradient (8), each additional observable tracked causes the optimization trajectory to deviate further from that followed by the gradient.

Figure 3 depicts the optimal control fields identified by $m=2$ and 10 MOTC algorithms. Note that, even in cases such as this where the imposition of multiple observable expectation value constraints drives the system to a final unitary propagator W that is closer to U_0 , the optimal fields are invariably more complex, with the Fourier spectra displaying higher-frequency modes for additional observables. The dissimilarity of these fields from the initial guess indicates that optimal fields for multiobservable quantum control cannot be identified by heuristic reasoning based solely on, for example, the spectrum of the Hamiltonian.

One disadvantage of tracking-based optimization of observable expectation values, compared to steepest ascent search, is that more precise measurements of the gradient are required to remain on the desired track. Tracking paths for additional observables, where the auxiliary tracks are chosen so as to constrain the unitary propagator to more a uniform path, can confer additional stability and robustness to observable-tracking-based optimization.

Errors in tracking that occur due to breakdown of the first-order MOTC approximation may be stabilized by the

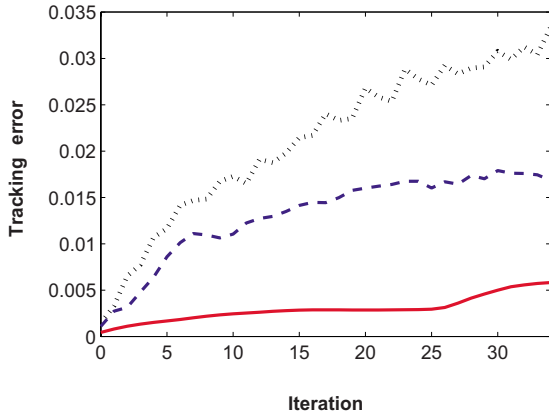


FIG. 4. (Color online) Observable tracking errors for the control problems displayed in Fig. 2 over the optimization trajectory. Dotted, two observables; dashed, four observables; solid, ten observables.

use of additional observables because it is less likely that the first-order approximation will break down simultaneously for all observables. Since more auxiliary observables can be tracked when $\rho(0)$ has more independent parameters, this stabilization method is most effective when $\rho(0)$ is full rank. Thus, in multiobservable tracking, a greater rank of $\rho(0)$ can increase the maximal stability of the algorithm, as well as the diversity of possible multiobservable expectation value targets and the freedom to follow arbitrary paths toward those targets. Figure 4 compares the tracking errors for the MOTC variants depicted in Fig. 2.

The relative efficiency of MOTC and gradient flow observable optimization can be assessed by employing ordinary differential equation (ODE) integrators with adaptive step-size. For this purpose, the Fehlberg embedded Runge-Kutta (ASRK5) method was used. Although it is somewhat more difficult to implement RK integrators experimentally, they provide a measure of the maximum step size that can be taken along a given optimization trajectory without incurring errors above a specified tolerance. Figure 5 compares the number of ASRK5 iterations needed to solve the above maximization problem by integrating (a) the gradient flow ODE (9) and (b) the MOTC flow (30) with w_s set to the ten-observable expectation value track corresponding to the geodesic between U_0 and W . Another adaptive step-size algorithm for improving optimization efficiency, a line search (Sec. VI), was also used in the case of the gradient (data not shown; whereas either ODE can be integrated via ASRK5, only the gradient flow can be simply implemented via a line search routine). The substantial decrease in the number of required iterations in the case of tracking algorithms, indicative of enhanced global optimality of the path they follow in $L^2(\mathbb{R})$, is displayed in the figure.

VIII. DISCUSSION

We have presented a class of deterministic algorithms for the optimal control of multiple quantum observables, based on tracking multiobservable expectation value paths to desired end points. These algorithms leverage the previously

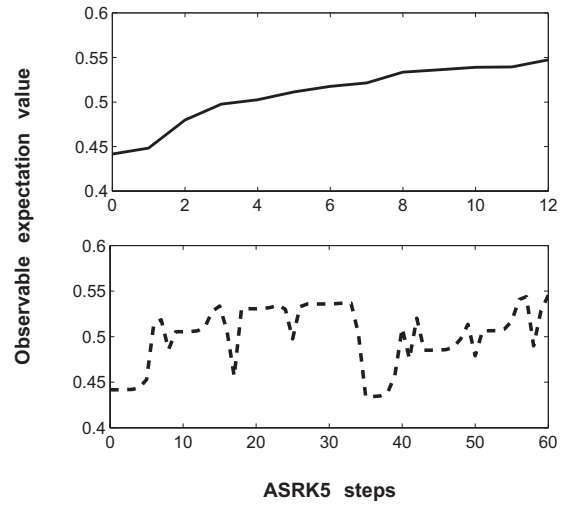


FIG. 5. Algorithmic efficiency of observable control algorithms assessed through variation of step size. Top: Adaptive step size Runge-Kutta (ASRK5) MOTC integration (ten observables); bottom: ASRK5 gradient flow integration, both for the observable maximization problem in Fig. 2. Expectation value refers to the first observable.

reported simple critical topology of quantum control landscapes [11], which indicates that gradient-based algorithms will rarely encounter singularities. An additional feature of multiobservable control landscapes that governs the ability of local search algorithms to follow arbitrary paths in multiobservable space, the MOTC Gramian matrix, has been identified and its properties investigated numerically for selected problems. Error-correction methods have been described that should facilitate the experimental implementation of these algorithms in the presence of noise. Moreover, a general MOTC framework has been presented that extends beyond the specific problem of multiobservable control to encompass more general quantum multiobjective optimization problems. Extensions to problems involving objectives in multiple quantum systems [8] are straightforward.

The performance of MOTC algorithms has been compared to that of local gradient flow algorithms based on scalar objective functions. Although the $\epsilon(t)$ gradient flow is always the locally optimal path, its projected path in $\mathcal{U}(N)$ is generally much longer than those that can be tracked by global MOTC algorithms. Even for single-observable control problems, the latter often require fewer iterations for convergence.

Features of the optimal control fields have been compared for various numbers of controlled observables. Application of multiple-observable expectation value constraints has been shown to generally increase the complexity of optimal control fields on the level set of a quantum control landscape [11] through the introduction of higher-frequency field modes beyond those present in the transition frequency spectrum of the internal Hamiltonian.

Two MOTC applications of particular importance are (A) the simultaneous maximization of the expectation values of a set of observables and (B) the preparation of arbitrary mixed states. (A) corresponds to the problem of identifying Pareto optimal solutions to the multiobservable control problem,

i.e., control fields $\varepsilon(t)$ such that all other fields have a lower value for at least one of the objective functions Φ_k , or else have the same value for all objectives. Whereas in generic multiobjective problems, the Pareto frontier is very difficult to sample due to its irregular structure, the simple landscape topology and geometry of quantum optimal control problems enables highly efficient methods for Pareto front exploration based on MOTC.

Problem (B) requires the control of a large number of quantum observables (up to N^2-1 , for Hilbert space dimension N). For such tasks, the eigenvalue spectrum of the initial quantum state plays an important role in determining the

maximum possible optimization efficiency. In these cases, it is usually possible to accelerate MOTC by choosing an optimal set of observable operators to be measured at each step. In this paper, we have focused on developing the formalism of MOTC and comparing the algorithmic properties governing the efficiency of MOTC with those of gradient flow and unitary matrix tracking algorithms. In a follow-up work, we will examine strategies for the efficient experimental implementation of MOTC, as well as methods for combining quantum state estimation with MOTC for effective Pareto front sampling.

-
- [1] A. Assion, T. Baumert, M. Bergt, T. Brixner, and B. Kiefer, *Science* **282**, 919 (1998).
- [2] T. S. Ho and H. Rabitz, *J. Photochem. Photobiol., A* **180**, 226 (2006).
- [3] H. Rabitz, M. Hsieh, and C. Rosenthal, *Science* **303**, 1998 (2004).
- [4] A. P. Peirce, M. A. Dahleh, and H. Rabitz, *Phys. Rev. A* **37**, 4950 (1988).
- [5] J. P. Palao and R. Kosloff, *Phys. Rev. Lett.* **89**, 188301 (2002).
- [6] J. P. Palao and R. Kosloff, *Phys. Rev. A* **68**, 062308 (2003).
- [7] S. M. Weber, F. Sauer, M. Plewicky, A. Merli, L. Wöste, and A. Lindinger, *J. Mod. Opt.* **54**, 2659 (2007).
- [8] L. Bonacina, J. Extermann, A. Rondi, V. Boutou, and J. P. Wolf, *Phys. Rev. A* **76**, 023408 (2007).
- [9] R. Levis, G. Menkir, and H. Rabitz, *Science* **292**, 709 (2001).
- [10] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, *IEEE Trans. Evol. Comput.* **6**, 182 (2002).
- [11] R. Chakrabarti and H. Rabitz, *Int. Rev. Phys. Chem.* **26**, 671 (2007).
- [12] R. Wu, A. Pechen, H. Rabitz, M. Hsieh, and B. Tsou, *J. Math. Phys.* **49**, 022108 (2008).
- [13] J. Roslund and H. Rabitz (unpublished).
- [14] C. Hillermeier, *Nonlinear Multiobjective Optimization: A Generalized Homotopy Approach* (Birkhauser, Basel, 2001).
- [15] H. Rabitz, M. Hsieh, and C. Rosenthal, *J. Chem. Phys.* **124**, 204107 (2006).
- [16] R. Wu, H. Rabitz, and M. Hsieh, *J. Phys. A* **41**, 015006 (2008).
- [17] See EPAPS Document No. E-PLRAAN-77-044806 for further details on the algorithmic efficiency of OCT gradient flow algorithms. For more information on EPAPS, see <http://www.aip.org/pubservs/epaps.html>.
- [18] M. Bertero, C. D. Mol, and E. R. Pike, *Inverse Probl.* **1**, 301 (1985).
- [19] J. Dominy and H. Rabitz, *J. Phys. A* **41**, 205305 (2008).
- [20] B. Bonnard and M. Chyba, *Singular Trajectories and Their Role in Control Theory* (Springer, Berlin, 2003).
- [21] R. Brockett, *Linear Algebr. Appl.* **146**, 79 (1991).
- [22] A. Rothman, T. S. Ho, and H. Rabitz, *Phys. Rev. A* **72**, 023416 (2005).
- [23] F. Yip, D. Mazziotti, and H. Rabitz, *J. Chem. Phys.* **118**, 8168 (2003).
- [24] W. Press, S. Teukolsky, W. Vetterling, and B. Flannery, *Numerical recipes in C++* (Cambridge University Press, Cambridge, U.K., 2002).
- [25] A. Uhlmann, *Rep. Math. Phys.* **9**, 273 (1976).