

PAPER

## Efficient optimization of state preparation in quantum networks using quantum trajectories

To cite this article: Michael H Goerz and Kurt Jacobs 2018 *Quantum Sci. Technol.* **3** 045005

View the [article online](#) for updates and enhancements.

### Related content

- [Controlling open quantum systems: tools, achievements, and limitations](#)  
Christiane P Koch
- [Optimal control theory for a unitary operation under dissipative evolution](#)  
Michael H Goerz, Daniel M Reich and Christiane P Koch
- [Quantum simulation of a quantum stochastic walk](#)  
Luke C G Govia, Bruno G Taketani, Peter K Schuhmacher et al.



**IOP | ebooks™**

Bringing you innovative digital publishing with leading voices to create your essential collection of books in STEM research.

Start exploring the collection - download the first chapter of every title for free.

# Quantum Science and Technology



## PAPER

# Efficient optimization of state preparation in quantum networks using quantum trajectories

RECEIVED  
13 January 2018

REVISED  
19 June 2018

ACCEPTED FOR PUBLICATION  
21 June 2018

PUBLISHED  
16 July 2018

Michael H Goerz<sup>1,2</sup>  and Kurt Jacobs<sup>2,3,4</sup>

<sup>1</sup> Edward L. Ginzton Laboratory, Stanford University, Stanford, CA 94305, United States of America

<sup>2</sup> U.S. Army Research Laboratory, Computational and Information Sciences Directorate, Adelphi, MD 20783, United States of America

<sup>3</sup> Department of Physics, University of Massachusetts at Boston, Boston, MA 02125, United States of America

<sup>4</sup> Hearne Institute for Theoretical Physics, Louisiana State University, Baton Rouge, LA 70803, United States of America

E-mail: [goerz@stanford.edu](mailto:goerz@stanford.edu)

**Keywords:** quantum optics, quantum networks, optimal control, numerical methods

## Abstract

The wavefunction Monte-Carlo method, also referred to as the use of ‘quantum jump trajectories’, allows efficient simulation of open systems by independently tracking the evolution of many pure-state ‘trajectories’. This method is ideally suited to simulation by modern, highly parallel computers. Here we show that Krotov’s method of numerical optimal control, unlike others, can be modified in a simple way so that it becomes fully parallel in the pure states without losing its effectiveness. This provides a highly efficient method for finding optimal control protocols for open quantum systems and networks. We apply this method to the problem of generating entangled states in a network consisting of systems coupled in a unidirectional chain. We show that due to the existence of a dark state subspace in the network, nearly optimal control protocols can be found for this problem by using only a single pure-state trajectory in the optimization, further increasing the efficiency.

## 1. Introduction

The control of simple quantum systems is central to the future realization of quantum technologies [1–4]. Such systems are usually open, meaning that they experience a significant source of noise from their environments [5, 6]. Systems that form the nodes in a quantum network, in which the nodes are connected together via traveling-wave fields are also necessarily open due to the interaction with the fields [6–9]. At the simplest level, open quantum systems and networks can be described by a first-order differential equation for the density matrix called a master equation. A useful feature of master equations is that the evolving density matrix for the system or systems can alternatively be written as the average over an ensemble of pure states in which the evolution of each state is given by an equation with a random (stochastic) component [6, 10–13]. For a  $d$ -dimensional system the representation of the density matrix scales as  $d^2$  while that of a pure-state scales only as  $d$ . For this reason the pure-state ensemble approach, or ‘quantum trajectory’ approach, can provide a significant reduction in numerical overhead. Here we use the trajectory method in which the stochastic element is a series of ‘jumps’ at random times, often referred to as the ‘quantum Monte-Carlo’ or ‘quantum jump trajectories’ method [14–16].

The ability to vary one or more parameters of the Hamiltonian of a system over time is a powerful tool for controlling the system. The problem of determining the time-dependence, or ‘pulse shapes’ required to realize a given evolution is highly non-trivial, however, and as a result numerical search methods, often referred to as ‘optimal control’ methods, are very useful for this purpose [17, 18]. To perform a numerical search requires simulating the evolution of a system or network many times under differing pulse shapes to sufficiently explore the ‘control space’ of possible pulses. The use of quantum trajectories for such simulations, especially as the size of a network increases, is thus desirable.

Here we examine the form that gradient-based numerical search methods take when written in terms of the ensemble of pure-state evolutions (the ‘quantum trajectories’) that simulate the master equation. We show that

unlike other methods, Krotov's method can be formulated in such a way that updates to the pulse shapes are obtained either from independent trajectories, or by quadratic cross-referencing of trajectories. We then show via a numerical example that Krotov's method remains effective even for a small number of trajectories, including a single trajectory.

In the second part of this work we apply the above 'Krotov trajectory method' to a problem of preparing a class of entangled states in a simple network. This network consists of a 'chain' of systems connected via a single unidirectional field. Each system, or 'node' in the network is a cavity containing a system with two stable levels, and for which the interaction with the cavity mode can be turned on and off via a local control field. One way in which these nodes can be realized is by having a three-level atom in each cavity, in which the three levels are in a lambda configuration. The two ground states provide the two stable levels, and the interaction between these two levels and the cavity mode is mediated by an off-resonant coupling to the excited state. Our problem may be thought of as a generalization of the configuration used by Cirac *et al* in their seminal work in which they showed how the state of a two-level system in a cavity could be transferred to a second two-level system in a second cavity via a unidirectional field [19]. They found that, so long as the coupling of each cavity to the traveling-wave field was the only source of dissipation, the coupling between the two-level systems and their respective cavities could be controlled in such a way that the excitation stored in the first two-level system could be emitted into the field, enter the second cavity, and be completely re-absorbed by the second two-level system, without being lost from the second cavity to the traveling field (that is, lost out of the 'open end' of the field that travels away from the cavities). Strictly speaking, to effect perfect transfer in this way requires an infinite time, since the transfer of the excitation to the field is an exponential decay, but near-unity fidelity can be obtained within a finite multiple of the decay time. One way to understand the protocol of Cirac *et al* is to say that there is a subspace of joint states of the two nodes that are 'dark' in the sense that no photons will be emitted from the systems when in this subspace [20–23]. The protocol is able to realize the transfer by evolving exclusively within this subspace. Protocols that employ the transfer method proposed by Cirac *et al* have been realized in a number of experiments [24–28].

Here we consider the preparation of dark states of  $N$  nodes in which a single-excitation, originally prepared in the first node, is shared equally between all the nodes. We find that by controlling only the local coupling between the nodes and the field link it is possible to prepare these multi-partite entangled states while remaining within the dark state subspace, thus realizing essentially perfect preparation (up to the infidelity imposed by the finite duration of the protocol and any sources of loss and decoherence additional to the field link). We also find that because the optimal protocols are able to enter the dark state subspace quickly on the time scale of the transfer, the majority of the numerical search can be performed with a single pure-state trajectory, thus greatly increasing the efficiency.

Our trajectory-based optimization method is applicable to essentially any control task in which the effect of the environment can be written as an average over pure-state trajectories. This is true for all Markovian master equations that have the Lindblad form, and a few non-Markovian master equations [29–35]. Our method is especially useful for situations in which there are dark state spaces in which the effect of the bath vanishes. This can be true for quantum networks, as in the example we explore here. Another scenario in which this holds is that of the storage and retrieval of a photon wave-packet from a collective dark state of an atomic ensemble [36–39].

The rest of this paper is structured as follows. In section 2 we first review the Monte-Carlo wavefunction method and the two preeminent gradient-based methods of quantum optimal control, gradient-ascent-pulse-engineering (GRAPE) and Krotov's method. Then, in section 2.3, we develop the central result of this paper, a trajectory-based variant of Krotov's method. We illustrate the method in a numerical case study in section 3. We first introduce the network model for a chain of cavities containing trapped atoms. Then, in sections 3.2–3.4, we present the result of applying optimal control to the creation of a dark state in that system, and analyze the convergence and the noise properties of the optimized pulses. Section 4 concludes.

## 2. Optimal control via quantum trajectories

### 2.1. The Monte-Carlo wavefunction method

We consider the problem of initializing a quantum network from a well-defined separable state  $|\Psi(t=0)\rangle$  to a (generally entangled) state  $|\Psi_{\text{tgt}}(t=T)\rangle$ . The network is characterized by a Hamiltonian  $\hat{H} = \hat{H}_0 + \sum_i u_i(t)\hat{H}_i$  with one or more control fields  $u_i(t)$ , and a set of Lindblad operators  $\{\hat{L}_l\}$  to describe dissipative processes. In the Markov limit [5], the dynamical map  $\hat{\rho}(t) = \mathcal{E}(t, 0)\hat{\rho}(0)$  denotes the solution of the master equation

$$\frac{\partial \hat{\rho}}{\partial t} = -\frac{i}{\hbar}\mathcal{L}\hat{\rho} = -\frac{i}{\hbar}[\hat{H}(\{u_i(t)\}), \hat{\rho}] + \mathcal{L}_D\hat{\rho} \quad (1)$$

with the boundary condition  $\hat{\rho}_0 = |\Psi(0)\rangle\langle\Psi(0)|$  and the dissipator  $\mathcal{L}_D$ .

For a network consisting of more than a trivially small number of nodes, in general, the exponential growth of the Hilbert space will quickly render the numerical treatment of the system dynamics in Liouville space unfeasible. Significantly less resources are required to simulate the dynamics through quantum trajectories. In this case,  $\hat{\rho}(t)$  is approximated as an average of the pure states  $|\Psi_k(t)\rangle\langle\Psi_k(t)|$  resulting from  $M \rightarrow \infty$  trajectories indexed by  $k$ ,

$$\hat{\rho}(t) = \lim_{M \rightarrow \infty} \frac{1}{M} \sum_k |\Psi_k(t)\rangle\langle\Psi_k(t)|. \quad (2)$$

Each trajectory  $|\Psi_k(t)\rangle$  is a possible, statistically independent, pure-state realization of the system dynamics, taking into account dissipative processes. There are several ways to obtain the trajectories, corresponding to different physical measurements on the environment (e.g. the field into which the system decays), and governed by different stochastic equations of motion. Measurements of one or both of the quadratures of the field result in a quantum diffusion equation (QSDE) [13, 40], while photon counting results in quantum jumps (MCWF) [15, 16]. For the purposes of evaluating the density matrix via (2), the various kinds of trajectories are equivalent. We will focus on the quantum jump method because it is the most straightforward to realize numerically.

The *MCWF algorithm* for propagating the trajectory state  $|\Psi_k\rangle$  is [41]

1. define the non-Hermitian effective Hamiltonian

$$\hat{H}_{\text{eff}} = \hat{H} + \sum_l \hat{L}_l^\dagger \hat{L}_l \quad (3)$$

2. draw a random number  $r \in [0, 1)$

3. propagate until  $|\Psi_k(t)\rangle^2 = r$

4. apply an instantaneous quantum jump

$$|\Psi_k(t)\rangle \rightarrow \frac{\hat{L}_l |\Psi_k(t)\rangle}{\|\hat{L}_l |\Psi_k(t)\rangle\|},$$

choosing  $\hat{L}_l$  from the set of all Lindblad operators with probability

$$p(\hat{L}_l) = \langle\Psi_k(t)|\hat{L}_l^\dagger \hat{L}_l|\Psi_k(t)\rangle \quad (4)$$

5. draw a new random number  $r \in [0, 1)$  and continue the propagation

6. normalize any resulting  $|\Psi_k(t)\rangle$ .

In performing the propagation with discrete time steps, as is essential numerically, it is crucial to determine the jump times in step 3 with high precision (through interpolation and bisection), and in this way multiple instantaneous jumps can be included within one time step. Using this approach, relatively long time steps are possible. An alternative ‘first-order’ scheme [16] in which at most one jump may happen in each time step is simpler to implement, but requires small time steps to be accurate.

## 2.2. Optimal Control with GRAPE and Krotov’s method

We seek optimal solutions  $u_i(t)$  to minimize the error functional

$$J_T = 1 - \langle\langle\hat{\rho}(T)|\hat{P}_{\text{tgt}}\rangle\rangle, \quad (5)$$

for a fixed duration  $T$ , where  $\hat{P}_{\text{tgt}}$  is the projector onto the target state  $|\Psi_{\text{tgt}}\rangle$ . We use the notation  $\langle\langle a|b\rangle\rangle \equiv \text{tr}[a^\dagger b]$  for the Hilbert–Schmidt overlap of two operators.

There are well-established gradient-based methods of numerical optimal control to solve this problem. The two preeminent ones are GRAPE [42], usually as a quasi-Newton method in combination with the L-BFGS-B algorithm [43, 44], and Krotov’s method [45–47]. Both of these are *iterative*: they start from ‘guess’ fields  $u_i^{(0)}(t)$  and calculate updates  $\Delta u_i(t) = u_i^{(1)}(t) - u_i^{(0)}(t)$  that decrease the value of the target functional; for the next iteration,  $u_i^{(1)}(t)$  becomes the guess, and the procedure repeats until convergence is reached.

The GRAPE method starts with the assumption that  $u_i(t)$  with  $t \in [0, T]$  is discretized into  $n_t$  time steps, such that the time evolution up to a point  $t_j$  is

$$\hat{\rho}(t_j) = \mathcal{E}(t_j, 0) \hat{\rho}(0) = \left( \prod_{j'=j}^1 \mathcal{E}_{j'} \right) \hat{\rho}(0), \quad (6)$$

where  $\mathcal{E}_j \equiv \mathcal{E}(t_j, t_{j-1})$  denotes the dynamical map  $\hat{\rho}(t_{j-1}) \rightarrow \hat{\rho}(t_j)$ , with  $t_0 = 0$  and  $t_{n_t} = T$ . The indices in the product run backwards to account for time ordering. The control fields are approximated as constant within the

time interval,  $u_{ij} \equiv u_i([t_{j-1}, t_j])$ . Each time-local value  $u_{ij}^{(0)}$  is modified according to the gradient

$$\Delta u_{ij} \propto \frac{\partial J_T}{\partial u_{ij}} = - \left\langle \left\langle \hat{\rho}^{(0)}(t_j) \left| \frac{\partial \mathcal{E}_j}{\partial u_{ij}} \right| \hat{\rho}^{(0)}(t_{j-1}) \right\rangle \right\rangle, \quad (7)$$

where

$$\hat{\rho}^{(0)}(t_j) = \left( \prod_{j'=j+1}^{n_t} \mathcal{E}_{j'}^{(0)\dagger} \right) \hat{\rho}_{\text{tgt}} \quad (8)$$

is the target state backward-propagated with the conjugate Lindbladian. The superscript zero indicates the dynamics under the guess controls.

Krotov's method takes a slightly different approach. Starting from time-continuous guess controls  $u_i^{(0)}(t)$ , it analytically constructs pulse updates  $\Delta u_i(t)$  that minimize an extended functional, commonly

$$J = J_T + \sum_i \frac{\lambda_i}{S_i(t)} \int_0^T [u_i^{(1)}(t) - u_i^{(0)}(t)]^2 dt, \quad (9)$$

where  $\lambda_i$  is an arbitrary weight determining the overall magnitude of  $\Delta u_i(t)$ , and  $S_i(t) \in [0, 1]$  is a shape function that may be used to enforce boundary conditions. As the optimization converges,  $\Delta u_i(t) \rightarrow 0$ , such that  $J$  and  $J_T$  become equivalent. The control problem is solved by the update equation

$$\Delta u_i(t) = \frac{S_i(t)}{\lambda_i} \text{Im} \left\langle \left\langle \hat{\rho}^{(0)}(t) \left| \frac{\partial \mathcal{L}}{\partial u_i(t)} \right| \hat{\rho}^{(1)}(t) \right\rangle \right\rangle, \quad (10)$$

where  $\hat{\rho}^{(0)}(t)$  is a state backward-propagated with the conjugate Lindbladian, see (8), with the boundary condition

$$\hat{\rho}(T) = - \frac{\partial J_T}{\partial \langle \langle \rho(T) |} = \hat{\rho}_{\text{tgt}}, \quad (11)$$

where we use a notation analogous to [47] for the derivative with respect to the formal co-state in inner products in Liouville space. Only now is the update  $\Delta u_i(t)$  discretized to a time grid ( $\Delta u_i(t) \rightarrow \Delta u_{ij}$ ,  $t \rightarrow t_j$ ), resulting in a formula that superficially resembles (7), with at least two crucial differences:

First, the forward-propagated state  $\rho(t)$  uses the updated controls  $u_i^{(1)}(t)$  in (10), and the guess controls  $u_i^{(0)}(t)$  in (7). This makes Krotov's method *sequential* (updates at later times depend on updates at earlier times), whereas GRAPE is *concurrent* (updates are independent).

Second, the local derivative term in each equation differs; the gradient derivative  $\partial \mathcal{E}_j / \partial u_{ij}$  can be evaluated similarly to  $\mathcal{E}_j$  itself, under the assumption that  $\mathcal{E}_j$  is a simple exponential [48]. In comparison,  $\partial \mathcal{L} / \partial u_{ij}$  is more straightforward to evaluate (and instantaneous). In full generality,  $\mathcal{L}$  in (10) is the right-hand side of the equation of motion for  $\hat{\rho}$ , whether or not this is a master equation in Lindblad form [47]. For the particular form (1) with a time-independent dissipator,  $\partial \mathcal{L} / \partial u_{ij} = [\hat{H}_i, \rho(t_j)]$ .

The backward-propagated state  $\hat{\rho}^{(0)}(t)$  is the same for Krotov's method and GRAPE only for the specific case of a direct state-to-state transfer, functional (5). More generally, in Krotov's method, the boundary condition is determined explicitly by (11), which for other optimization tasks (e.g., gate optimization) does not yield states identical to the target states (but usually, a linear combination of them). Even though for GRAPE, the boundary condition also implicitly depends on the choice of functional, for all optimization tasks in quantum control that we are aware of, the backwards propagation always starts from the target states. While we have assumed linear controls and a convex functional, Krotov's method can be also be extended to go beyond these assumptions [47].

### 2.3. Krotov's method for quantum trajectories

There are two possibilities for rewriting the optimization in terms of quantum trajectories. First, we can insert the expansion of  $\hat{\rho}$  (2) into the functional (5) to find

$$J_T = 1 - \lim_{M \rightarrow \infty} \frac{1}{M} \sum_{k=1}^M \underbrace{|\langle \Psi_k(T) | \Psi_{\text{tgt}} \rangle|^2}_{\equiv \tau_k}. \quad (12)$$

That is, the fidelity is simply the average of the fidelities from each trajectory. We seek a single control that simultaneously implements the state-to-state transition for every  $|\Psi_k\rangle$ .

We now wish to ask, given the form of  $J_T$ , how the update rules for GRAPE and Krotov's method might be written in terms of the individual trajectories. For GRAPE this is not so simple; the form of the update rule is

based on the fact that the evolution of  $\hat{\rho}$  in each time step is given by the application of a linear operator  $\mathcal{E}_j$ , and this is no-longer true for the trajectories. Second, in a given time step the evolution of a given trajectory may contain one or more jumps at random times, which further complicates the calculation of the derivative with respect to the controls  $u_{ij}$ .

In contrast, for Krotov's method the functional (12) leads to the update equation [46]

$$\Delta u_i(t) = \frac{S_i(t)}{M\lambda_i} \sum_{k=1}^M \underbrace{\text{Im} \langle \chi_k^{(0)}(t) | \hat{H}_i | \Psi_k^{(1)}(t) \rangle}_{\equiv \Delta u_{ik}(t)}, \quad (13)$$

where  $\chi_k^{(0)}(t)$  is backward-propagated with the boundary condition

$$\chi_k^{(0)}(T) = -\frac{\partial J_T}{\partial \langle \Psi_k |} = \tau_k^{(0)} | \Psi_{\text{tgt}} \rangle, \quad (14)$$

where  $\tau_k^{(0)}$  is  $\tau_k$  from (12) evaluated for the guess pulse. The crucial difference is that for GRAPE, the gradient is evaluated for the time evolution operator, whereas for Krotov's method, the gradient is with respect to the equation of motion. Only the latter is immediately compatible with the decomposition into quantum trajectories.

From a numerical perspective, the update equation (13) can be evaluated with high efficiency: the contributions to the update  $\Delta u_{ik}(t)$  from every trajectory are completely independent. If the numerical method is parallelized, so that different trajectories are simulated on different processors, and a message-passing-interface (MPI) is used for interprocessor communication, only these scalar values need to be communicated. We thus reap the full benefit of the MCWF method: reduction of the overall dimensionality by a factor of  $d$ , the dimension of the total Hilbert space. The required memory per compute node is reduced directly by that factor; the worst-case CPU time reduces by  $d^3$  (as the fundamental operation in the time propagation is a matrix-vector multiplication).

We can obtain an alternative formulation of the control problem in terms of quantum trajectories by leaving the functional (5) in terms of the density matrix, and instead expanding the update equation in terms of the trajectories. For GRAPE this approach has the same issues as before, whereas for Krotov's method this results in the update equation

$$\Delta u_i(t) = \frac{S_i(t)}{M^2\lambda_i} \sum_{k,k'=1}^M \text{Im} [\langle \xi_k^{(0)}(t) | \hat{H}_i | \Psi_{k'}^{(1)}(t) \rangle \langle \Psi_{k'}^{(1)}(t) | \xi_k^{(0)}(t) \rangle], \quad (15)$$

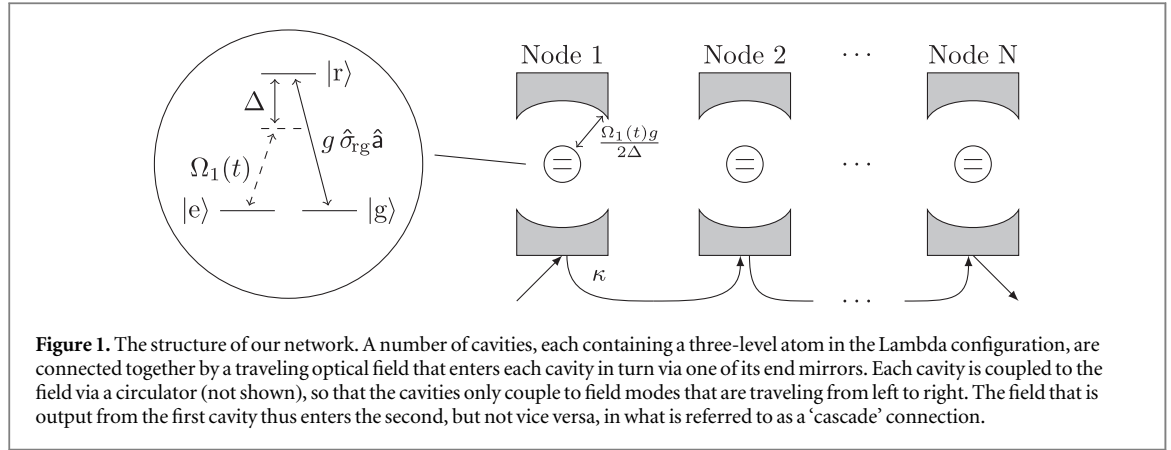
using the decomposition

$$\hat{\rho}^{(0)}(t) = \lim_{M \rightarrow \infty} \frac{1}{M} \sum_{k=1}^M |\xi_k(t)\rangle \langle \xi_k(t)| \quad (16)$$

for the backward-propagation with  $|\xi_k^{(0)}(T)\rangle = |\Psi_{\text{tgt}}\rangle$ . The above update now incorporates *cross-trajectory* information, including overlaps of the propagated states from each trajectory with those from every other trajectory. Compared to the full density matrix optimization, this alternative method maintains the saving in CPU time and memory that comes from evolving the trajectories independently, but does incur significantly more communication overhead.

Both trajectory formulations of Krotov's method, the 'independent trajectory' and 'cross-trajectory' formulations, guarantee monotonic convergence for  $M \rightarrow \infty$ . However, the numerical efficiency rests on the assumption that  $M$  can be kept reasonably small, ideally smaller than  $d$  (and certainly smaller than  $d^3$ ). For a significantly larger number of trajectories, the total numerical resources will approach or even surpass those required for a full density matrix propagation. Note that even if there is no benefit in the *total* numerical resources, the memory *per compute node* is still reduced by a factor of  $d$ .

For finite (small)  $M$ , there is no guarantee of monotonic convergence. This is especially true because the stochastic jumps are different between iterations, as well as between the forward and the backward-propagation within the same iteration. Moreover, we expect the instantaneous jumps in the trajectory to cause jumps also in the control update. These jumps will average out both with the number of trajectories and the number of control iterations, but they may slow down convergence and may introduce noise into the optimized control field. To analyze how strongly the resulting optimized control fields are affected by these two factors, and to what extent the cross-trajectory update equation might be more robust, we now consider a practical example.



### 3. Creation of an entangled dark state in a quantum network

#### 3.1. Network model

We consider a network in which a number of ‘nodes’ are connected via a traveling-wave field, depicted in figure 1. The nodes are connected to the field via circulators so that the field propagates from one node to the next in a single direction only, a configuration often referred to as a ‘cascade’ network. Each node consists of a cavity containing a single atom. The atom encodes a qubit in two degenerate states  $|g\rangle$ ,  $|e\rangle$ , driven via a Raman transition through an auxiliary level  $|r\rangle$ :  $|g\rangle \leftrightarrow |r\rangle$  couples through a time-dependent drive  $\Omega(t)$  with detuning  $\Delta$ , while  $|r\rangle \leftrightarrow |e\rangle$  is statically coupled with coupling strength  $g$ , see figure 1. For large  $\Delta$ , the level  $|r\rangle$  can be adiabatically eliminated [19], resulting in an effective two-level system whose coupling to the cavity is controlled via the drive  $\Omega(t)$ .

For a single node labeled  $(i)$ , the effective Hamiltonian in the rotating wave approximation then consists of the drift term  $\hat{H}_0$  and a driven Jaynes–Cummings term  $\hat{H}_d$ . With the cavity detuned from the driving field by  $g^2/\Delta$  to compensate for the Stark shift, we find

$$\hat{H}^{(i)} = \hat{H}_0^{(i)} + \hat{H}_d^{(i)}, \quad (17)$$

$$\hat{H}_0^{(i)} = -\frac{g^2}{\Delta} \hat{a}_i^\dagger \hat{a}_i + \frac{g^2}{\Delta} \hat{\Pi}_g^{(i)} \otimes \hat{a}_i^\dagger \hat{a}_i, \quad (18)$$

$$\hat{H}_d^{(i)} = -\frac{i\Omega_i(t)g}{2\Delta} (\hat{\sigma}_{e,g}^{(i)} \otimes \hat{a}_i - \text{c.c.}), \quad (19)$$

where  $\hat{a}_i$  is the cavity lowering operator,  $\hat{\Pi}_g$  is the projector on the qubit ground state, and  $\hat{\sigma}_{e,g}$  is the raising operator of the qubit. Leakage of photons out of the cavity is described by the Lindblad operator

$$\hat{L}^{(i)} = \sqrt{2\kappa} \hat{a}_i. \quad (20)$$

There is also spontaneous decay from level  $|r\rangle$  of the atom, but the decay is suppressed for large  $\Delta$  (the prerequisite for the adiabatic elimination that yields the effective two-level system), and can be neglected [19].

We now couple  $N$  cavities to a traveling-wave field via circulators, so that the field propagates from cavity to cavity in only one direction, as depicted in figure 1. The resulting master equation describing the evolution of all the cavities is obtained using input–output theory [8, 9, 49]. The explicit application of the theory is through the ‘SLH’ formalism of Gough and James [7, 50, 51]. While the calculation can be done by hand, it is tedious, and we instead use the QNET software package, which automates the SLH formalism<sup>5</sup>.

The total Hamiltonian for the entire network is

$$\hat{H} = \sum_{i=1}^N \hat{H}^{(i)} + \sum_{i,j>i}^N \hat{H}^{(i,j)}, \quad (21)$$

$$\hat{H}^{(i,j)} = i\kappa \hat{a}_i^\dagger \hat{a}_j + \text{c.c.} \quad (22)$$

Note that the connection via the field yields the additional static interaction Hamiltonian  $\hat{H}^{(i,j)}$  between *all* nodes of the network (not merely between nearest neighbors). The Lindblad operator for the entire network is

<sup>5</sup> The python package QNET allows the user to construct quantum input–output networks, and will calculate the master equation for any such network [52].

$$\hat{\mathcal{L}} = \sum_{i=1}^N \hat{\mathcal{L}}^{(i)}. \quad (23)$$

Thus, both the interaction between nodes and the total dissipation of the network are proportional to the decay rate  $\kappa$ . In order to create an entangled state quickly,  $\kappa$  (the interaction) must be large, which would seem to imply rapid decay of the network. However, the form of the Lindblad operator allows for the individual summands  $\hat{\mathcal{L}}_i$  to destructively interfere, canceling overall dissipation. If the dark state condition [20–23]

$$\forall t: \langle \Psi(t) | \hat{\mathcal{L}}^\dagger \hat{\mathcal{L}} | \Psi(t) \rangle = 0 \quad (24)$$

is fulfilled, an entangled state may be generated without loss of coherence. Nonetheless, we must consider the dynamics of a quantum network to be inherently highly dissipative, as the dark state condition is only fulfilled for carefully chosen control pulses that evolve the network exclusively within the dark state manifold.

In applying our numerical search method to the problem of generating an entangled dark state in the network we assume that the field link between cavities is the only significant source of loss. Since the entangled dark state exists only because the same field link is coupled to multiple systems, any sources of loss that are not similarly collective will necessarily limit the fidelity with which the entangled state can be prepared. Such losses include spontaneous emission from the atomic excited state (which is suppressed by the detuning), internal cavity losses, and loss *en route* between the cavities (as opposed to loss out of the open end of the field link, which is the collective loss). These losses will be negligible so long as the loss rates are small compared to the time it takes the protocol to prepare the entangled state. (This time scale is set by the rate at which the cavities decay out the open end of the field link, which is also the rate of the effective field-mediated coupling between the cavities.) We ignore all additional sources of loss in our analysis because (i) the effect of these loss sources cannot be reduced in any significant way by optimizing the control protocol (since they do not possess any associated dark states), and thus while this loss limits the achievable fidelity it is not expected to change the convergence properties of the numerical search method, and (ii) to prepare high-fidelity entangled states requires that any additional loss is small compared to that of the field link itself.

### 3.2. Optimized creation of an entangled dark state

We assume that at time zero, the network is initialized in the state  $|\Psi(t=0)\rangle = |eg\dots g\rangle$ . That is, the atom in the cavity of the leftmost node is in the excited state, while all other nodes (and all cavities) are in the ground state. The state of the cavities,  $|0\dots 0\rangle$  is implicit in our notation. We now seek the control fields  $\Omega_i(t)$  (one field for each node) that will bring the system into an entangled dark state at a fixed final time  $T$ ,  $|\Psi(t=T)\rangle = \frac{1}{\sqrt{N}}(|eg\dots g\rangle + |ge\dots g\rangle + \dots + |gg\dots e\rangle)$ . The cavity for all nodes again must be in the ground state at time  $T$ .

The optimization problem is solved by minimizing (5) using Krotov's method. We first do this optimization using the full density matrix, update (10), and then—for comparison—using a trajectory optimization, both for independent trajectories, functional (12) with update (13), and for the cross-trajectory method, update (15).

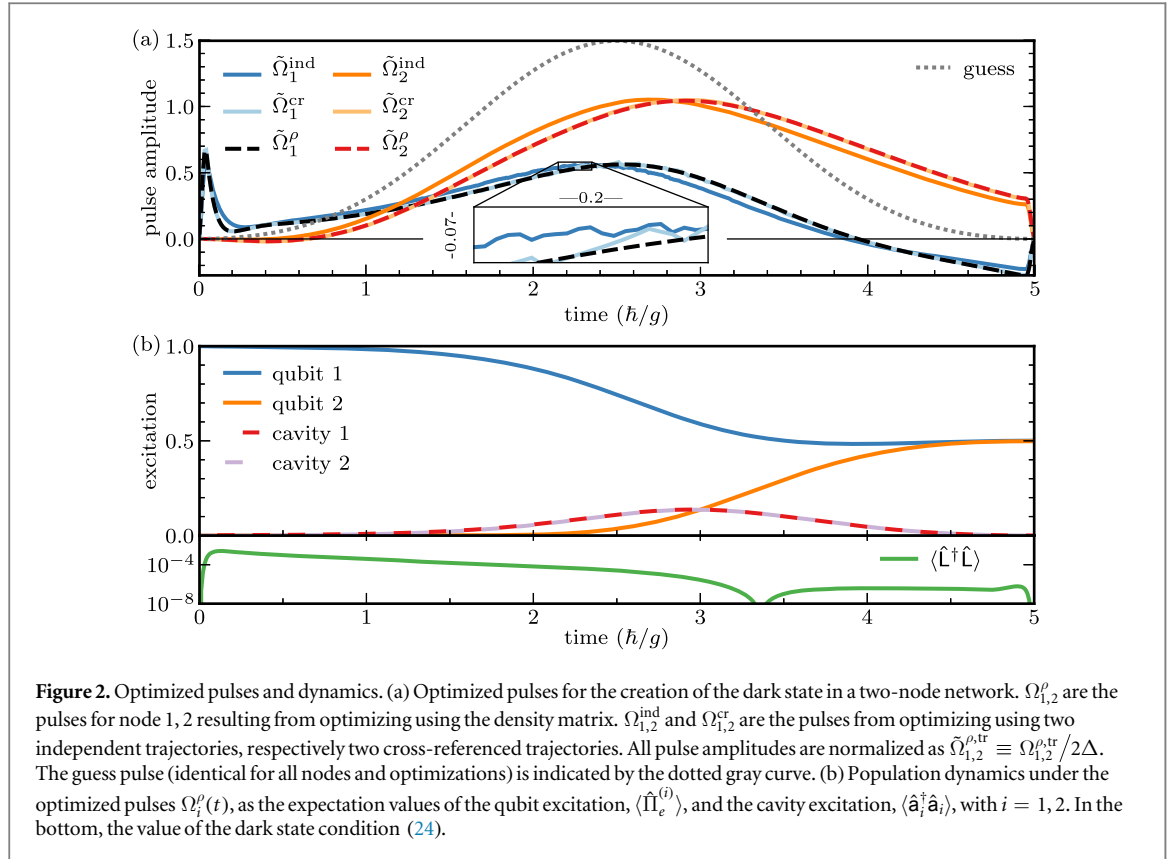
One reason for choosing this particular physical system as an example to illustrate our proposed optimization method is that the Hamiltonian (21) preserves the total excitation in the system. Thus, it is sufficient to model both the cavity and the atom at each node as a two-level-system. Moreover, when representing the state numerically, we need only include the single-excitation subspace along with the ground state—to account for the possibility of spontaneous decay (the loss of the excitation out the open end of the field that links the cavities). The effective dimension of the resulting Hilbert space is  $2N + 1$ , instead of the exponentially scaled  $4^N$ . This makes a full density matrix optimization feasible even for relatively large  $N$ , allowing us to compare against the trajectory optimizations. We consider first the simplest case of  $N = 2$  nodes, with control fields  $\Omega_1(t)$  and  $\Omega_2(t)$ . In this case, the dark state is the Bell state

$$|\Psi\rangle_{\text{tgt}} = \frac{1}{\sqrt{2}}(|eg\rangle + |ge\rangle). \quad (25)$$

This allows us to easily illustrate the characteristics of the optimized pulses and node dynamics. To ensure that our conclusions hold for larger networks, we then consider a network of 20 nodes. In principle, even larger networks would be numerically feasible. However (as we will see), the optimization becomes increasingly harder, due to the drastically larger search space resulting from  $N$  independent control pulses. When the number of nodes is significantly higher than 20, we do not find good control solutions within a reasonable number of iterations (even for the optimization that employs the full density matrix). We expect that it might be possible to sufficiently reduce the search space for larger networks by not treating all pulses as fully independent.

For full generality, we use a dimensionless Hamiltonian where energies are written in units of  $g$ . Consequently, time can be expressed in units of  $\hbar/g$ . For  $\Delta = 100g$  and  $\kappa = g$ , through systematical variation of  $T$ , we find that the two-node dark state can comfortably be realized for  $T = 5 \hbar/g$ . For 20 nodes, we correspondingly increase the process duration to  $T = 50 \hbar/g$ . In general, for a fixed number of nodes, the





**Figure 2.** Optimized pulses and dynamics. (a) Optimized pulses for the creation of the dark state in a two-node network.  $\Omega_{1,2}^\rho$  are the pulses for node 1, 2 resulting from optimizing using the density matrix.  $\Omega_{1,2}^{\text{ind}}$  and  $\Omega_{1,2}^{\text{cr}}$  are the pulses from optimizing using two independent trajectories, respectively two cross-referenced trajectories. All pulse amplitudes are normalized as  $\tilde{\Omega}_{1,2}^{\rho,\text{tr}} \equiv \Omega_{1,2}^{\rho,\text{tr}}/2\Delta$ . The guess pulse (identical for all nodes and optimizations) is indicated by the dotted gray curve. (b) Population dynamics under the optimized pulses  $\Omega_i^\rho(t)$ , as the expectation values of the qubit excitation,  $\langle \Pi_e^{(i)} \rangle$ , and the cavity excitation,  $\langle \hat{a}_i^\dagger \hat{a}_i \rangle$ , with  $i = 1, 2$ . In the bottom, the value of the dark state condition (24).

minimum time required to entangle the nodes of the network is proportional to the interaction Hamiltonian (22), that is, proportional to the value of  $\kappa$ .

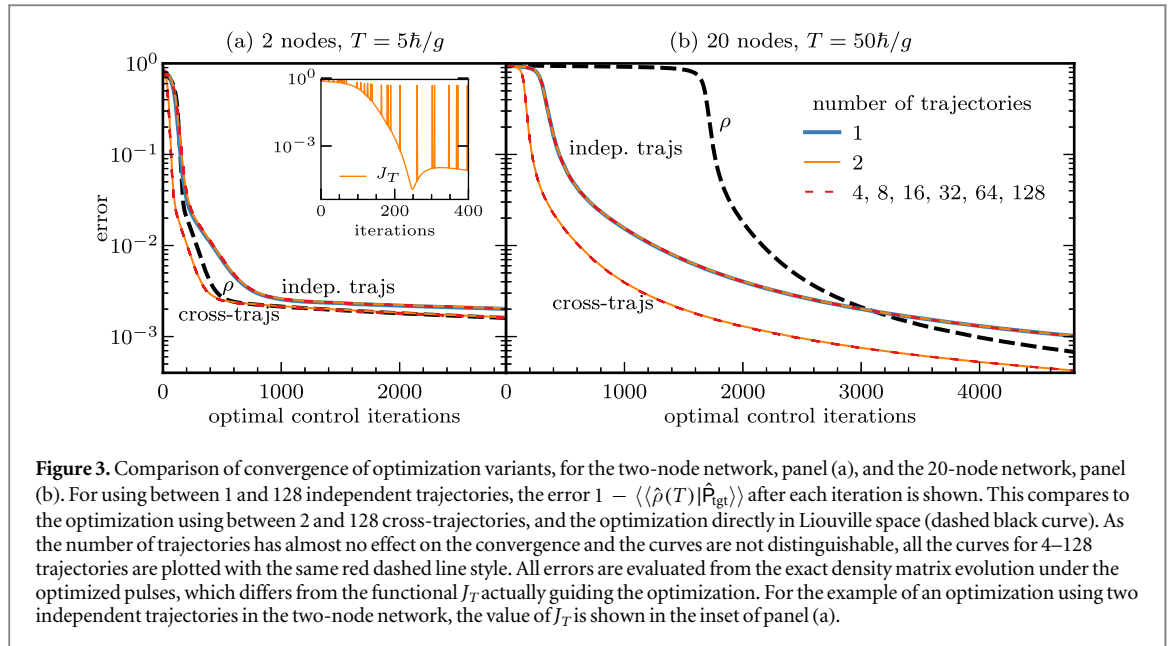
Figure 2 shows the result of an optimization for two nodes starting from a simple Blackman-shaped guess pulse, see the dotted gray line in panel (a). For a direct density matrix optimization using Krotov's method, that is the minimization of (5) with update (10), after 5000 iterations an error of  $J_T = 1.3 \times 10^{-3}$  is achieved. The optimized pulses  $\Omega_1(t)$  and  $\Omega_2(t)$  for the first and second node are shown in figure 2(a). We show the pulses resulting from the full density matrix optimization (superscript ' $\rho$ '; black, red), from two independent trajectories (superscript 'ind'; dark blue, dark orange), and from two cross-referenced trajectories (superscript 'cr'; light blue, light orange). The corresponding dynamics for the pulses resulting from the density matrix optimization are in panel (b), showing the smooth transition from the initial state  $|eg\rangle$  to the superposition of  $|eg\rangle$  and  $|ge\rangle$ : the excitation of the first and second qubit, the plotted expectation values  $\langle \Pi_e^{(1)} \rangle$  and  $\langle \Pi_e^{(2)} \rangle$ , correspond directly to the population of  $|eg\rangle$  and  $|ge\rangle$ , as the Hamiltonian preserves excitations ( $\langle ee|ee\rangle \equiv 0$ ).

As the interaction between the nodes is mediated by photons being emitted from cavity 1 into cavity 2, both cavities are necessarily excited during the transition (see the light and dark red dashed lines in figure 2(b)). Nonetheless, the dynamics are (nearly) coherent, due to the system being in a dark state (see the green line for  $\langle \hat{L}^\dagger \hat{L} \rangle$  in the bottom of figure 2(b), which is close to zero). The two cavities are phase-shifted by  $\pi$ , that is  $\langle \hat{a}_1 \rangle \approx -\langle \hat{a}_2 \rangle$ , resulting in destructive interference for the total dissipator.

As we have not taken into account any other dissipation channels (such as spontaneous decay of the qubit), the deviation from the dark state condition (24), is the only factor that limits the fidelity of the transfer. By continuing the optimization beyond 5000 iterations, the error could be made arbitrarily small in principle.

Optimizing with independent trajectories, yields very similar results, corresponding to the optimized pulses shown in figure 2(a) (dark orange, dark blue) being close but not identical to the pulses obtained from direct density matrix optimization (red, black). For the cross-trajectory method, the dynamics and optimized pulses (light colored) are virtually indistinguishable from the density matrix optimization.

In general, running both optimization methods with differing numbers of trajectories, and the density matrix method, all yield comparable errors. Nevertheless there are differences in the rate of convergence, as well as in certain details of the pulse features. As shown in the zoomed inset of figure 2(a), there is some noise in the optimized pulses obtained from trajectory optimization, compared to the perfectly smooth result of the density matrix optimization. This noise originates from the discontinuous jumps in the trajectories. The optimized pulses and dynamics for the 20-node network (not shown) have the same qualitative characteristics. In the



following, we will analyze how both the convergence rate and the noise in the optimized pulses depend on the choice of optimization method and the number of trajectories.

### 3.3. Convergence of trajectory optimization

Figure 3 shows the convergence, as measured by the error (5), over the number of iterations in the control procedure. In each iteration, the pulse is updated according to the appropriate update equation. Panel (a) shows the results for the two-node network, while panel (b) shows the result for the 20-node network.

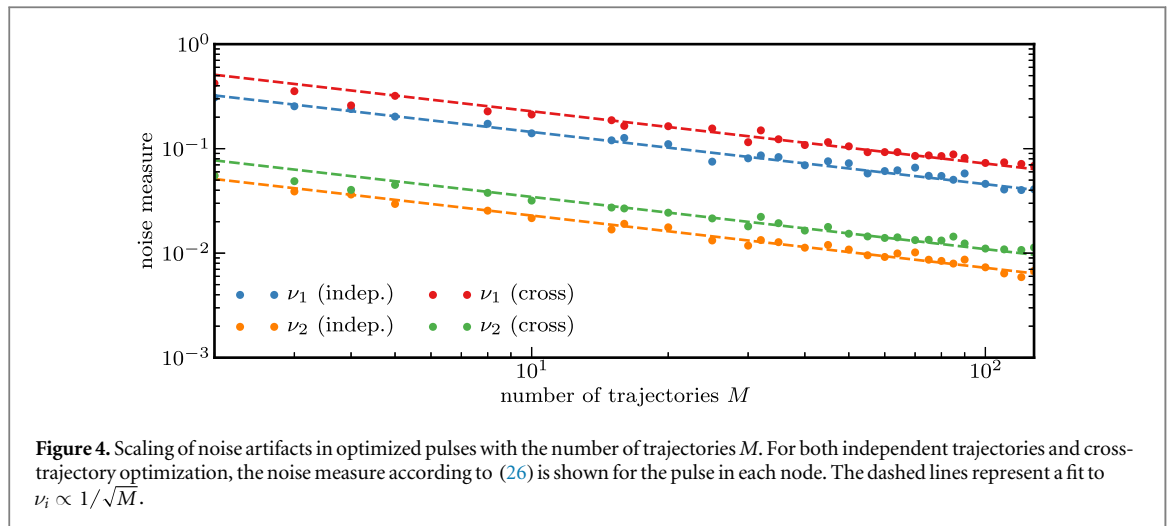
The convergence for the full density matrix optimization, update (10), is shown as the black dashed line. It compares to the convergence for using 1–128 independent trajectories (‘indep. trajs’), update (13), and 2–128 cross-trajectories (‘cross-trajs’), update (15).

We first observe that the optimization for the 20-node network is *much* harder than the optimization for 2 nodes, requiring several thousand iterations to reach errors below  $10^{-2}$ . In particular, the density matrix optimization has a long initial plateau before starting to converge. This behavior is exacerbated for more nodes, which is why we have limited ourselves to considering at most a 20-node network. Remarkably, the use of a trajectory optimization performs significantly better in this initial phase, largely avoiding the plateau. We presume this is because the dissipative jumps affect the trajectories much more strongly than the smooth decay affects the density matrix, leading to a steeper gradient. Both the two-node network and the 20-node network have the same qualitative convergence characteristics; the larger number of nodes simply ‘stretches’ them out.

The convergence of the cross-trajectory optimization consistently outperforms the optimization using independent trajectories. Asymptotically, the independent trajectories eventually lag behind the full density matrix optimization, but only slightly so. The cross-trajectory optimization asymptotically reaches the same error as the full density matrix optimization (clearly for the two-node network, but we may extrapolate this claim to hold also for 20 nodes).

A second remarkable observation is that the number of trajectories has almost no effect on the convergence. Thus, even a single trajectory, or two cross-referenced trajectories, can yield a satisfactory result, a significant reduction of numerical effort compared to the use of full density matrices. It is worth pointing out that as long as the number of trajectories does not exceed the number of CPU cores available on a compute node, it would be easy to parallelize the method not using a message-passing interface, but instead using a shared-memory approach (e.g. OpenMP). In this case, the additional numerical overhead of the cross-trajectory optimization relative to independent trajectories largely disappears.

For the trajectory optimizations, the error (10) that is shown in figure 3 is not identical to the functional  $J_T$  that the optimization directly aims to minimize (12). Because of the finite number of trajectories and the randomness inherent in the quantum jumps,  $J_T$  itself does not converge monotonically. An example for two independent trajectories is shown in the inset of figure 3. The loss of a photon from the cavity out of the open end of the field link (that is, a quantum jump) in a given iteration results in a total loss (error 1.0), as visible in the spikes in  $J_T$ . In addition, the lower envelope ( $J_T$  in the absence of photon loss) shows a local minimum around 250 iterations, owing to the two competing requirements of the optimization: achieving the desired target state (in the absence of photon loss), and minimizing decay (the dark state condition). For cross-trajectory



optimization,  $J_T$  behaves similarly (not shown). The non-monotonic convergence of  $J_T$  should not be held against the trajectory optimization method: as shown in figure 3 the actual physical error does decrease monotonically.

### 3.4. Jump noise

The use of trajectory optimization results in some noise in the control fields, see the inset in figure 2(a). To analyze how the magnitude of this noise scales with the number of trajectories, we quantify the noise by comparing the optimized control field  $\Omega_i(t)$  to a smoothed pulse  $\Omega_i^{\text{smooth}}(t)$

$$\nu_i = \int_0^T |\Omega_i(t) - \Omega_i^{\text{smooth}}(t)| dt. \quad (26)$$

For the purpose of this analysis, a good noise filter should generate an  $\Omega_i^{\text{smooth}}(t)$  that deviates from  $\Omega_i(t)$  as little as possible, while achieving a well-defined degree of smoothness. A standard Savitzky–Golay filter [53] is well-suited to solve this problem [54].

We find that the noise as defined in (26) asymptotically scales with the number of trajectories  $M$  as  $\nu_i \propto 1/\sqrt{M}$ . This is illustrated in figure 4, which compares the noise measure for both pulses of the two-node network, and for independent- and cross-trajectory optimization, as a function of the number of trajectories  $M$ . The results shown here use a five-point cubic convolute in the Savitzky–Golay filter; this choice of window length and polynomial order only affects the proportionality factor, i.e. the y-intercept in figure 4.

For independent trajectories, the  $1/\sqrt{M}$  scaling holds for all values of  $M$ . The pulse  $\Omega_1$  is noisier than the pulse  $\Omega_2$ , which is due to the fact that  $\Omega_2$  is closer to the original guess pulse than  $\Omega_1$ , see figure 2(a)—any noise induced by a jump is proportional to the magnitude of the pulse update in a given iteration.

The noise for the cross-trajectory optimization is systematically slightly higher than that for independent trajectories. We may conjecture that this is because the update (15) is quadratic in the (discontinuous) states, while the update (13) is only linear. However, for a small number of cross-trajectories ( $< \approx 5$ ), the noise is smaller than the asymptotic  $1/\sqrt{M}$  scaling, closer to the independent trajectories.

In any case, for this specific example, the noise has little effect on the fidelity, as we can see from figure 3: the optimization result is robust with respect to the level of noise introduced by the numerical procedure. If this were not that case, and if smooth pulses are required, one may also incorporate the smoothing directly into the optimization scheme by applying a smoothing filter, or fitting to a parametrized curve, after each iteration (or after some fixed number of iterations). The pulses resulting from the optimization in the 20-node network (not shown) have noise characteristics compatible with the above discussion.

## 4. Conclusion

We have formulated two variants of Krotov’s method of optimal control based on Monte-Carlo quantum jump trajectories for implementing state-to-state transitions. The fact that it is possible to apply Krotov’s method directly to stochastic trajectories is due to the fact that, unlike other methods, Krotov’s method does not require the calculation of the derivative of the propagator (which for stochastic trajectories does not exist). The first variant considers  $M$  independent trajectories, and optimizes the target in each trajectory, averaging pulse updates from the trajectories. It is easily parallelized to  $M$  processes communicating with a MPI. The second

‘cross-trajectory’ variant obtains an update by cross-referencing  $M$  trajectories into a coarse approximation of the density matrix evolution. When parallelized using MPI, this produces considerable overhead. However, for small  $M$ , a shared-memory parallelization model is efficient.

In any case, the use of trajectories provides a considerable numerical advantage for systems that are strongly dissipative, but for which simulating the full density matrix is not feasible due to the large dimension of the Hilbert space. In many cases, the memory to store a density matrix of size  $d^2$  is not available, while a pure-state of dimension  $d$  can easily be stored and propagated. This is typically the case for quantum networks, and we have illustrated the method for a simple example of creating an entangled dark state for a network of 20 cascaded cavities in which each contains a trapped atom.

We find that the use of trajectories in the optimization yields results comparable to the full density matrix optimization. In the case of cross-trajectory optimization, the convergence of the optimization even outperforms the density matrix optimization.

Remarkably, the number of trajectories has almost no influence in the convergence or the error that is achieved. The number of trajectories does have an influence in the amount of noise artifacts in the optimized pulses that result from the discontinuous (jump) dynamics. These scale as  $1/\sqrt{M}$  in the number of trajectories. At the same time, they have an only negligible effect on the dynamics and the resulting error, and could also be removed by applying a smoothing filter to the optimized pulses.

Thus, we can conclude that a relatively small number of trajectories (even just a single trajectory) is sufficient to optimize a state-to-state transition in a typical dissipative system. Further, this trajectory approach is especially efficient when the optimal protocol evolves the system within a dark (dissipationless) subspace, a situation which arises, for example, both in quantum networks and atomic ensembles [36]. The method naturally extends to optimization problems beyond a simple state preparation, e.g. the realization of quantum gates [55], or any other optimization that is easily expressed in terms of the density matrix. We have presented our optimization method and the example of the creation of a dark state in the context of the master equation in Lindblad form. However, the method extends to non-Markovian dynamics as well. It has been shown that Krotov’s method can be applied beyond the Markov approximation and leads to an update equation that is structurally identical to (10) [56–58], which can then be rewritten in terms of quantum trajectories. Also, instead of a quantum jump propagation, another trajectory method such as quantum state diffusion [40] could be used. In all cases, the feasibility of the optimization using Krotov’s method results from the fact that no analytic derivative of the inherently non-analytical propagator is required.

## Acknowledgments

The network model used in this paper was derived using the Python QNET package [52]. The propagation and optimization were done using the Fortran QDYN library [59]. The pulse smoothing was realized through SciPy’s signal processing toolbox [60]. This research was supported in part by ASD(R&E) under their Quantum Science and Engineering Program (QSEP), and by the Army High Performance Computing Research Center (AHPCRC) (sponsored by the US Army Research Laboratory under contract No. W911NF-07-2-0027).

## ORCID iDs

Michael H Goerz  <https://orcid.org/0000-0003-2839-9976>

## References

- [1] Blume-Kohout R, Caves C M and Deutsch I H 2002 *Found. Phys.* **32** 1641
- [2] Giovannetti V, Lloyd S and Maccone L 2011 *Nat. Photon.* **5** 222
- [3] Stephens A M 2014 *Phys. Rev. A* **89** 022321
- [4] Roy A and DiVincenzo D P 2017 Topological quantum computing *Topological Matter: Topological Insulators, Skyrmions and Majoranas, Lecture notes of the 48th IFF Spring School 2017* ed S Bluegel *et al* p 278
- [5] Breuer H P and Petruccione F 2007 *The Theory of Open Quantum Systems* (Oxford: Oxford University Press)
- [6] Jacobs K 2014 *Quantum Measurement Theory and Its Applications* (Cambridge: Cambridge University Press)
- [7] Combes J, Kerckhoff J and Sarovar M 2017 *Adv. Phys. X* **2** 784
- [8] Gardiner C W and Collett M J 1985 *Phys. Rev. A* **31** 3761
- [9] Gardiner C W and Zoller P 2004 *Quantum Noise* (Berlin: Springer)
- [10] Carmichael H 1993 *An Open Systems Approach to Quantum Optics* (Berlin: Springer)
- [11] Diosi L 1986 *Phys. Lett. A* **114** 451
- [12] Wiseman H M 1996 *Quantum Semiclass. Opt.* **8** 205
- [13] Wiseman H M and Milburn G J 1993 *Phys. Rev. A* **47** 642
- [14] Hegerfeldt G C and Wilser T S 1992 Ensemble or individual system, collapse or no collapse: a description of a single radiating atom *Proc. 2nd Int. Wigner Symp., Classical and Quantum Systems* ed H D Doebner *et al* (Singapore: World Scientific) pp 104–15

- [15] Dum R, Zoller P and Ritsch H 1992 *Phys. Rev. A* **45** 4879
- [16] Mølmer K, Castin Y and Dalibard J 1993 *J. Opt. Soc. Am. B* **10** 524
- [17] Glaser S J *et al* 2015 *Eur. Phys. J. D* **69** 279
- [18] Koch C P 2016 *J. Phys.: Condens. Matter* **28** 213001
- [19] Cirac J I, Zoller P, Kimble H J and Mabuchi H 1997 *Phys. Rev. Lett.* **78** 3221
- [20] Kraus B, Büchler H P, Diehl S, Kantian A, Micheli A and Zoller P 2008 *Phys. Rev. A* **78** 042307
- [21] Clark S, Peng A, Gu M and Parkins S 2003 *Phys. Rev. Lett.* **91** 177901
- [22] Cho J, Bose S and Kim M S 2011 *Phys. Rev. Lett.* **106** 020504
- [23] Stannigel K, Rabl P and Zoller P 2012 *New J. Phys.* **14** 063014
- [24] Ritter S, Nölleke C, Hahn C, Reiserer A, Neuzner A, Uphoff M, Mücke M, Figueroa E, Bochmann J and Rempe G 2012 *Nature* **484** 195
- [25] Yin Y *et al* 2013 *Phys. Rev. Lett.* **110** 107001
- [26] Wenner J *et al* 2014 *Phys. Rev. Lett.* **112** 210501
- [27] Axline C J, Burkhardt L D, Pfaff W, Zhang K, Chou P, Campagne-Ibarcq P, Reinhold P, Frunzio L, Girvin S M and Jiang L 2018 *Nature Phys.* (<https://doi.org/10.1038/s41567-018-0115-y>)
- [28] Kurpiers P, Magnard T, Walter B, Royer M, Pechal J, Heinsoo Y, Salath'e A, Akin S, Storz J and Besse C 2018 *Nature* **558** 264
- [29] Breuer H P, Kappler B and Petruccione F 1999 *Phys. Rev. A* **59** 1633–43
- [30] Strunz W T, Diósi L and Gisin N 1999 *Phys. Rev. Lett.* **82** 1801–5
- [31] Strunz W T, Diósi L, Gisin N and Yu T 1999 *Phys. Rev. Lett.* **83** 4909–13
- [32] Gambetta J and Wiseman H M 2002 *Phys. Rev. A* **66** 012108
- [33] Gambetta J, Askerud T and Wiseman H M 2004 *Phys. Rev. A* **69** 052104
- [34] Piilo J, Maniscalco S, Härkönen K and Suominen K A 2008 *Phys. Rev. Lett.* **100** 180402
- [35] Wiseman H M and Gambetta J M 2008 *Phys. Rev. Lett.* **101** 140401
- [36] Fleischhauer M and Lukin M D 2000 *Phys. Rev. Lett.* **84** 5094–7
- [37] Liu C, Dutton Z, Behroozi C H and Hau L V 2001 *Nature* **409** 490
- [38] Novikova I, Gorshkov A V, Phillips D F, Sørensen A S, Lukin M D and Walsworth R L 2007 *Phys. Rev. Lett.* **98** 243602
- [39] Gorshkov A V, André A, Fleischhauer M, Sørensen A S and Lukin M D 2007 *Phys. Rev. Lett.* **98** 123601
- [40] Percival I 1998 *Quantum State Diffusion* (Cambridge: Cambridge University Press)
- [41] Plenio M B and Knight P L 1998 *Rev. Mod. Phys.* **70** 101
- [42] Khaneja N, Reiss T, Kehlet C, Schulte-Herbrüggen T and Glaser S J 2005 *J. Magn. Reson.* **172** 296
- [43] Byrd R H, Lu P, Nocedal J and Zhu C 1995 *SIAM J. Sci. Comput.* **16** 1190
- [44] Zhu C, Byrd R H, Lu P and Nocedal J 1997 *ACM Trans. Math. Softw.* **23** 550
- [45] Krotov V F 1996 *Global Methods in Optimal Control* (New York: Dekker)
- [46] Palao J P and Kosloff R 2003 *Phys. Rev. A* **68** 062308
- [47] Reich D M, Ndong M and Koch C P 2012 *J. Chem. Phys.* **136** 104103
- [48] de Fouquières P, Schirmer S, Glaser S and Kuprov I 2011 *J. Magn. Reson.* **212** 412
- [49] Gardiner C W 1993 *Phys. Rev. Lett.* **70** 2269
- [50] Gough J and James M 2009 *Commun. Math. Phys.* **287** 1109
- [51] Gough J and James M R 2009 *IEEE Trans. Autom. Control* **54** 2530
- [52] QNET 2018 <https://github.com/mabuchilab/QNET>
- [53] Savitzky A and Golay M J E 1964 *Anal. Chem.* **36** 1627
- [54] Press W H, Teukolsky S A, Vetterling W T and Flannery B P 1993 *Numerical Recipes in FORTRAN; The Art of Scientific Computing* (New York: Cambridge University Press)
- [55] Goerz M H, Reich D M and Koch C P 2014 *New J. Phys.* **16** 055012
- [56] Ohtsuki Y 2003 *J. Chem. Phys.* **119** 661
- [57] Hwang B and Goan H S 2012 *Phys. Rev. A* **85** 032321
- [58] Tai J S, Lin K T and Goan H S 2014 *Phys. Rev. A* **89** 062310
- [59] QDYN 2017 <https://qdyn-library.net>
- [60] Jones E *et al* SciPy: open source scientific tools for Python <http://scipy.org/>