

Direct Collocation for Quantum Optimal Control

Aaron Trowbridge,¹ Aditya Bhardwaj,² Kevin He,² David I. Schuster,^{2,3} and Zachary Manchester¹

Abstract—We present an adaptation of direct collocation – a trajectory optimization method commonly used in robotics and aerospace applications – to quantum optimal control (QOC); we refer to this method as Padé Integrator Collocation (PICO). This approach supports general nonlinear constraints on the states and controls, takes advantage of state-of-the-art large-scale nonlinear programming solvers, and has superior convergence properties compared to standard approaches like GRAPE and CRAB. PICO also allows for the formulation of novel free-time and minimum-time control problems – crucial for realizing high-performance quantum computers when the optimal pulse duration is not known *a priori*. We demonstrate PICO’s performance both in simulation and on hardware with a 3D circuit cavity quantum electrodynamics system.

Keywords—quantum optimal control, superconducting qubits, direct collocation, nonlinear programming, numerical methods

I. INTRODUCTION

The field of optimal control, which has its origins in aerospace engineering and robotics, has produced a large body of sophisticated methods for solving control problems fundamentally similar to the problems posed in quantum optimal control (QOC) [1–5]. However, many of these methods have yet to be adopted by those working on control problems for quantum systems. This work aims to bridge the gap between robotic control and quantum control and, in so doing, provide a new perspective to practitioners of QOC.

Current widely used QOC methods [6, 7] fall into the category of *indirect methods*, in that they do not include the states as decision variables. In this work, we introduce Padé Integrator Collocation (PICO) as an alternative, fully direct method. PICO is an adaptation of the direct collocation method [8] tailored to the quantum setting. With this method, we are able to leverage large-scale nonlinear programming solvers, specifically the interior point method IPOPT [9]. We demonstrate that PICO, due to being a direct method, improves upon existing indirect methods in several ways, and is able to produce state-of-the-art results in both simulation and on hardware.

Our contributions include:

- A novel formulation of QOC using direct collocation
- Structure-preserving *Padé integrators* that efficiently compute an approximation of the matrix exponential to simulate quantum dynamics
- An open-source implementation QuantumCollocation.jl [10]

We demonstrate state-of-the-art results, both with regard to fidelity and time optimality, on a set of simulated problems of

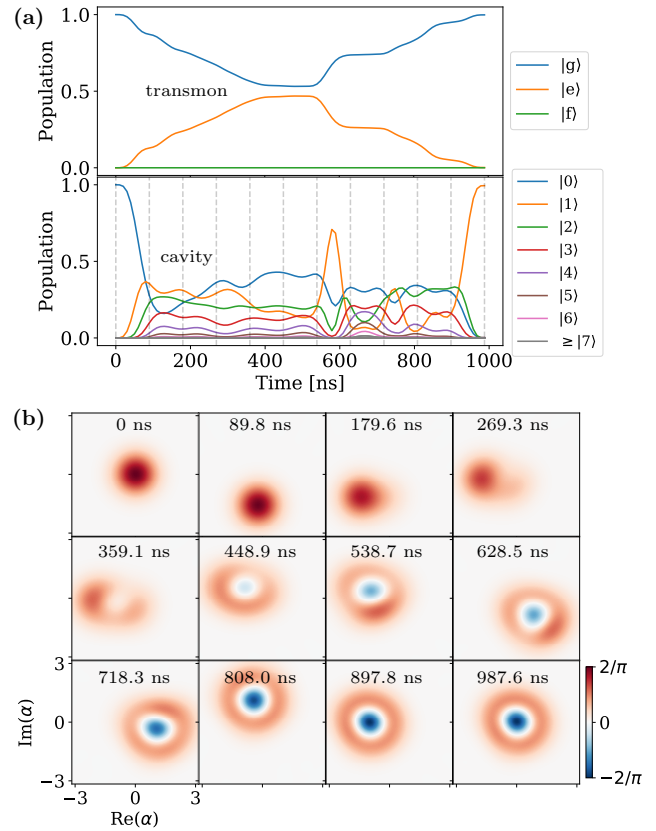


Fig. 1. **PICO pulse simulated state evolution for a target state of $|g1\rangle$.** (a) Time evolution of the separate transmon and cavity levels for the duration of the control pulse. In either case, the other system component is traced out. Cavity photon number populations increase in order, before being manipulated to end in the single photon state. For this pulse, the transmon $|f\rangle$ state and cavity photon numbers above 6 are not significantly populated. Dashed vertical lines indicate the times of the Wigner tomography slices in (e). (b) Time evolution of the cavity populations in phase space, shown as Wigner tomography plots. Starting in the vacuum 0 photon state, the state evolves under the control pulse into the final 1 photon state. Time slices go from left-right and up-down at the equally spaced time points indicated by the gray vertical lines in the cavity populations plot in (d).

increasing difficulty. We also show that PICO produces pulses with state-of-the-art performance on an experimental system.

This paper is structured as follows: First, in Sec. II, we review the problem of quantum optimal control and existing solution methods. Then, in Sec. III, we introduce our method, PICO. In Sec. IV we demonstrate PICO’s performance on three QOC problems of increasing difficulty. Finally, in Sec. V, we present the hardware results achieved with PICO.

¹Carnegie Mellon University, Robotics Institute

²University of Chicago, Department of Physics

³Stanford University, Department of Applied Physics

II. BACKGROUND

In this section, we review relevant prior work. First, we discuss the mathematical formulation of quantum optimal control. Then, we review gradient-based methods for solving QOC problems, followed by gradient-free methods. Finally, we review the direct collocation method commonly employed to solve trajectory optimization problems in aerospace and robotics applications.

A. Quantum Optimal Control

The control of quantum systems can be framed as an optimization problem over the space of time-dependent state trajectories subject to dynamics described by the Schrödinger equation. For a unitary operator, in our setting, this is given by:

$$\dot{U} = -iH(\mathbf{a}(t))U. \quad (1)$$

The dynamics are controllable via *drive* parameters $\mathbf{a}(t) \in \mathbf{R}^d$ in the Hamiltonian. For simplicity, we will limit ourselves to considering Hamiltonians of the form,

$$H(\mathbf{a}(t)) := H_0 + \sum_i a^i(t)H_i, \quad (2)$$

where $t \in [0, T]$, H_0 is the system's *drift* term, and H_i are referred to as the *drive* terms. Handling other types of Hamiltonians—e.g. those that are nonlinear in the controls—is also possible in our method.

QOC problems typically fall into three categories, corresponding to three different types of quantum objects: quantum states $|\psi(t)\rangle$, which satisfy $\partial_t |\psi\rangle = -iH |\psi\rangle$; unitary operators $U(t)$, which satisfy Eqn. (1); and density operators $\rho(t)$, which satisfy $\dot{\rho}(t) = -i[H, \rho]$.

QOC methods are agnostic to the form of the state and the dynamics, so we will primarily discuss unitary operators, as they are arguably the most general. In this case, given a desired gate U_{goal} , the objective we will be most concerned with is the unitary infidelity loss, defined as,

$$\ell(U) := 1 - \frac{1}{n} \left| \text{tr} \left(U_{\text{goal}}^\dagger U \right) \right|, \quad (3)$$

for $U \in SU(n)$.

We discretize the time interval $[0, T]$ into N time steps of size Δt ; the states and controls at each time step are denoted U_k and \mathbf{a}_k , respectively. One then solves the following optimization problem:

$$\underset{\mathbf{a}_{1:N-1}}{\text{minimize}} \quad J(\mathbf{a}_{1:N-1}) = \ell(U_N(\mathbf{a}_{1:N-1})), \quad (4)$$

where

$$U_N(\mathbf{a}_{1:N-1}) = \prod_{k=1}^{N-1} \exp(-iH(\mathbf{a}_k)\Delta t). \quad (5)$$

Current approaches for solving this problem fall into two categories: *gradient-based* methods and *basis function* methods. Both of these are *indirect* methods, as they treat the final state $U(T)$ as a function of the controls and minimize the objective $J(\mathbf{a}_{1:N-1})$ only over the controls $\mathbf{a}_{1:T-1}$, as opposed to considering both the states and controls as decision variables in what are known as *direct* methods [11].

B. Gradient-Based Methods

Gradient-based methods involve initializing the controls $\mathbf{a}_{1:T-1}$ with an initial guess, rolling out the state using Eqn. (5), and then iteratively updating the controls using a gradient descent algorithm; this type of approach is also referred to as a *shooting method*. Evaluating the objective $J(\mathbf{a}_{1:N-1})$, which requires a costly full rollout, is necessary at each iteration to compute a gradient and update the controls:

$$\mathbf{a} \leftarrow \mathbf{a} - \beta \nabla J(\mathbf{a}). \quad (6)$$

There are efficient ways to compute this gradient [12], but the approach is still limited by rollouts and other factors: The solution is dependent on the initial guess, gradient-based methods are prone to falling into local minima, and it is difficult to rigorously enforce constraints on the states.

The most popular gradient-based method is known as GRAPE (GRAdient Ascent Pulse Engineering) [6], which is available through the popular QuTiP python library. The company Q-CTRL also implements its own gradient-based optimization tool [13, 14], which is the industry standard, and is what we compare our results to in this paper.

C. Gradient-Free Methods

There are other approaches that do not require taking gradients (but still require objective evaluations). These approaches typically utilize a parameterized basis of functions to sufficiently reduce the number of decision variables so that gradient-free optimization algorithms — e.g. the Nelder-Mead simplex method — can be utilized. In the case of the popular CRAB algorithm [7], this is accomplished by utilizing a Fourier basis for each pulse component:

$$a(t) = 1 + \frac{\sum_{n=1}^{N_c} b_n \sin(\omega_n t) + c_n \cos(\omega_n t)}{\lambda(t)}. \quad (7)$$

Where N_c is the number of terms taken in the series expansion and $\lambda(t)$ is chosen to enforce the boundary conditions. The result is a smaller optimization problem:

$$\underset{b_{1:N_c}, c_{1:N_c}, \omega_{1:N_c}}{\text{minimize}} \quad J(b_{1:N_c}, c_{1:N_c}, \omega_{1:N_c}). \quad (8)$$

D. Trajectory Optimization and Direct Collocation

An alternative approach for solving trajectory optimization problems is *direct collocation* (DIRCOL) [8]. DIRCOL is a direct method that overcomes many of the limitations of indirect methods by including both the states and controls at each time step as decision variables, denoted by x_k and u_k respectively. In this formulation, objective evaluations are cheap, as they do not involve rollouts, and the dynamics are explicitly enforced as equality constraints between *knot points* (state and control samples) $z_k = (x_k, u_k)^\top$.

Enforcing the dynamics as constraints is a key property of DIRCOL that allows numerical solvers to temporarily violate these constraints during intermediate steps of the solution process *en route* to satisfying them at convergence. This *infeasible-start* capability, along with the ability to easily

enforce constraints on the state variables, is the source of many of the advantages of direct over indirect methods.

A trajectory optimization problem can be simply stated in the direct framework: We begin in an initial state, i.e. $x_1 = x_{\text{init}}$, and find a control sequence $u_{1:N-1}$ such that x_N minimizes an objective J consisting of a loss $\ell(x_N)$ measuring the distance between the final state x_N and the goal state x_{goal} . We may also include other objectives terms, including e.g. penalties on higher derivatives of the control pulse to encourage smoothness. It is common to enforce the dynamics constraints implicitly, i.e. as $f(z_k, z_{k+1}) = 0$. We can then write the DIRCOL problem as:

$$\begin{aligned} & \underset{z_{1:N}}{\text{minimize}} && J(z_{1:N}) \\ & \text{subject to} && f(z_k, z_{k+1}) = 0, \\ & && x_1 = x_{\text{init}}, \end{aligned} \quad (9)$$

which is a large sparse nonlinear program that can be efficiently solved with a nonlinear solver such as IPOPT.

III. PADÉ INTEGRATOR COLLOCATION

This section deals with formulating QOC problems as direct collocation trajectory optimization problems. We will focus on optimizing for $SU(n)$ gates. Using the loss $\ell(\cdot)$ defined in Eqn. (3) and the naive dynamics

$$f(U_{k+1}, U_k, \mathbf{a}_k, \Delta t) = U_{k+1} - \exp(-iH(\mathbf{a}_k)\Delta t) U_k, \quad (10)$$

where Δt is fixed. A simple DIRCOL formulation of this problem can be written as follows:

$$\begin{aligned} & \underset{U_{1:N}, \mathbf{a}_{1:N}}{\text{minimize}} && \ell(U_N) \\ & \text{subject to} && f(U_{k+1}, U_k, \mathbf{a}_k, \Delta t) = 0 \\ & && U_1 = I_n \end{aligned} \quad (11)$$

We detail several practical considerations: First, we will cover how to convert the complex-valued objects in Eqn. (11) to real values. Next, we will discuss a novel, efficient way to enforce the dynamics, which avoids costly evaluations of the matrix exponential present in (10). Finally, we will discuss a few extensions for achieving *smooth* and *time-optimal* solutions.

A. Isomorphic Formulation

To move between complex-valued quantum states and real-valued problem variables, we follow [12] and utilize an isomorphic representation for complex vectors and matrices. We use a *tilde* or the notation $\text{iso}(\cdot)$ to represent isomorphic representations. For a complex-valued vector $\psi \in \mathbf{C}^n$ and matrix $H \in \mathbf{C}^{n \times n}$, we then have, respectively,

$$\tilde{\psi} = \begin{pmatrix} \text{Re } \psi \\ \text{Im } \psi \end{pmatrix} \quad \text{and} \quad \tilde{H} = \begin{pmatrix} \text{Re } H & -\text{Im } H \\ \text{Im } H & \text{Re } H \end{pmatrix}, \quad (12)$$

where $\tilde{\psi} \in \mathbf{R}^{2n}$ and $\tilde{H} \in \mathbf{R}^{2n \times 2n}$.

Since the dynamics involve an evaluation of $\exp(-iH)$ — i.e. exponentiation of the *generator* — we also define:

$$G(H) := \text{iso}(-iH) = \begin{pmatrix} \text{Im } H & \text{Re } H \\ -\text{Re } H & \text{Im } H \end{pmatrix}. \quad (13)$$

And, since H is linear in \mathbf{a} and G is a linear function of H , we define:

$$G(\mathbf{a}) := G(H(\mathbf{a})) = G(H_0) + \sum_j a^j G(H_j). \quad (14)$$

B. Padé Integrators

The isomorphic dynamics are still in the form of Eqn. (10):

$$f(\tilde{U}_{k+1}, \tilde{U}_k, \mathbf{a}_k, \Delta t) = \tilde{U}_{k+1} - \exp(G(\mathbf{a}_k)\Delta t) \tilde{U}_k. \quad (15)$$

The matrix exponential in its present form is a costly operation that does not account for how matrix exponentials are numerically computed in practice. One approach to address this is to use the Padé approximant for the exponential [15], which approximates the matrix exponential as

$$\exp(A) \approx B^{-1}(A)F(A), \quad (16)$$

where $B(A)$ and $F(A)$ are truncated power series in A whose the coefficients can be chosen to match $\exp(A)$ up to some desired order.

We take advantage of this structure in PICO by recognizing that the leading matrix inverse is computationally expensive and not necessary to compute directly since we are enforcing the dynamics constraints implicitly. Thus, we can rewrite the dynamics using the Padé approximant as:

$$\mathbf{P}(\tilde{U}_{k+1}, \tilde{U}_k, \mathbf{a}_k, \Delta t) := B(\mathbf{a}_k, \Delta t)\tilde{U}_{k+1} - F(\mathbf{a}_k, \Delta t)\tilde{U}_k. \quad (17)$$

Since we solve most of our problems in a rotating frame without very fast dynamics, we find that the fourth order diagonal *Padé integrator*, denoted $\mathbf{P}^{(4)}$ is sufficient. It is defined by

$$B^{(4)}(\mathbf{a}, \Delta t) := I - \frac{\Delta t}{2}G(\mathbf{a}) + \frac{\Delta t^2}{12}G(\mathbf{a})^2 \quad (18)$$

and

$$F^{(4)}(\mathbf{a}, \Delta t) := I + \frac{\Delta t}{2}G(\mathbf{a}) + \frac{\Delta t^2}{12}G(\mathbf{a})^2. \quad (19)$$

Intuitively, B evolves the state backward a half step in time and F evolves the state forward a half step. Padé approximants have nice properties w.r.t. matrix Lie groups, namely that they are *structure-preserving* [16]). In practice, we find that any difference between rollouts under $\mathbf{P}^{(4)}$ and the matrix exponential is always of lower order than the infidelity of the optimal solution. For problems with particularly fast dynamics, higher-order Padé integrators can be used at very low additional computational cost.

C. Smooth Solutions

Smooth control pulses are often desirable. To achieve this, we augment the states of the system with the first and second derivatives of the drive parameter so that the knot points are now $z_k = (\tilde{U}_k, \mathbf{a}_k, \dot{\mathbf{a}}_k, \ddot{\mathbf{a}}_k)^\top$ with $\ddot{\mathbf{a}}$ the new control variable. This procedure is equivalent to using a piecewise cubic spline

interpolation of the controls, which is convenient for interpolating trajectories. The new dynamics for this augmented problem are given by

$$f(z_k, z_{k+1}) = \begin{pmatrix} \mathbf{P}^{(n)}(\tilde{U}_{k+1}, \tilde{U}_k, \mathbf{a}_k, \Delta t) \\ \mathbf{a}_{k+1} - \mathbf{a}_k - \dot{\mathbf{a}}_k \cdot \Delta t \\ \dot{\mathbf{a}}_{k+1} - \dot{\mathbf{a}}_k - \ddot{\mathbf{a}}_k \cdot \Delta t \end{pmatrix}. \quad (20)$$

By adding quadratic regularization costs on \mathbf{a}_k , $\dot{\mathbf{a}}_k$, and $\ddot{\mathbf{a}}_k$, and possibly bounding constraints on the velocities $\dot{\mathbf{a}}_k$ or accelerations $\ddot{\mathbf{a}}_k$, we can easily control the smoothness of the pulse.

D. Time-Optimal Solutions

In our framework, it is possible, and very useful, to treat the timestep Δt as a decision variable: the knot points are augmented as, e.g., $z_k = (\tilde{U}_k, \mathbf{a}_k, \Delta t_k)^\top$. It is often necessary to add bound constraints, $\Delta t_{\min} < \Delta t_k < \Delta t_{\max}$, to prevent the solver from taking advantage of discretization errors. In practice, it also helps to constrain all the Δt_k s to be equal.

This augmentation adds extra freedom to the optimization problem, allowing the optimal duration of the pulse, which is often not known *a priori*, to be found by the solver. This is referred to as a *free-time* problem. Moreover, we can now add a cost term to the objective of the form $J_{\text{mintime}}(\Delta t_{1:N-1}) = \sum_k \Delta t_k$ and an inequality constraint $\mathcal{F}(U_N) \geq \bar{\mathcal{F}}$ on the final state fidelity which is rigorously enforceable in our method. This allows us to achieve *minimum-time* solutions for a chosen fidelity, which are helpful for realizing higher-fidelity quantum computations in the presence of decoherence.

IV. SIMULATION EXAMPLES

In this section, we describe the results of applying PICO to a set of three examples of increasing difficulty: a minimum-time single-qubit problem, a two-qubit problem, and a three-qubit problem. In all examples, we use a random initial guess for the controls and the geodesic on $SU(n)$ from the identity to the desired gate as a (dynamically infeasible) initial state trajectory. Except for the single-qubit example, all examples in this paper (including the hardware result) use the smooth solution problem formulation described in Sec. III-C. Code for all of these examples can be found in the QuantumCollocation.jl GitHub repository [10].

A. Single-Qubit Y-Gate Minimum Time Problem

As a first example, we consider a single-qubit system defined in QCTRL's user guide [17] (note that we ignore the noise factor and compare against the system ignoring robustness) with the Hamiltonian:

$$H(\alpha(t), \gamma(t)) = \frac{1}{2}(\gamma(t)\sigma_- + \gamma^*(t)\sigma_+) + \frac{\alpha(t)}{2}\sigma_z. \quad (21)$$

Where $\alpha \in \mathbf{R}$, $\gamma \in \mathbf{C}$, σ_+ and σ_- are the qubit ladder operators and σ_z is the Pauli z matrix. The goal is to find a time-optimal pulse that enacts a Y -gate (i.e. $U_{\text{goal}} = \sigma_y$) while enforcing $|\alpha| < \alpha_{\max} = 2\pi \times 0.1$ MHz and $|\gamma| < \gamma_{\max} = 2\pi \times 0.3$ MHz.

To achieve this goal we first solve the following *free-time* problem:

$$\begin{aligned} & \underset{z_{1:N}}{\text{minimize}} && J_0(z_{1:N}) = Q \cdot \ell(\tilde{U}_N) + R(\alpha_{1:N-1}, \gamma_{1:N-1}) \\ & \text{subject to} && \mathbf{P}^{(4)}(\tilde{U}_{k+1}, \tilde{U}_k, (\alpha_k, \gamma_k), \Delta t_k) = 0, \\ & && \tilde{U}_1 = I_{2n}, \\ & && |\alpha_k| < \alpha_{\max}, \\ & && |\gamma_k|^2 < \gamma_{\max}^2, \\ & && \Delta t_{\min} < \Delta t_k < \Delta t_{\max}, \end{aligned} \quad (22)$$

where $z_k = (\tilde{U}_k, \alpha_k, \gamma_k, \Delta t_k)$, $N = 100$, $Q = 200$ is a user-specified cost shaping parameter, and $R(\alpha_{1:N-1}, \gamma_{1:N-1})$ is a quadratic regularization objective on the magnitude of the controls (see code for details).

With an initial solution to (22) found by PICO we then warm-start a second problem with a new objective $J = J_0 + D \sum_k \Delta t_k$, and an inequality constraint on the final fidelity, $\mathcal{F}(U_N) \leq \bar{\mathcal{F}}$, to prevent the fidelity from decreasing while we minimize the duration of the pulse. We used $D = 10^9$ and $1 - \bar{\mathcal{F}} = 5 \times 10^{-6}$. The final solution has an exponential rollout infidelity of 4.72×10^{-6} and a duration of 1.67 μs . As can be seen in Fig. 2, PICO has found the analytical *bang-bang* solution. Q-CTRL's proprietary method also finds a comparable solution, which is to be expected for this simple problem, but with PICO we are able to fix the fidelity and solve for the duration.

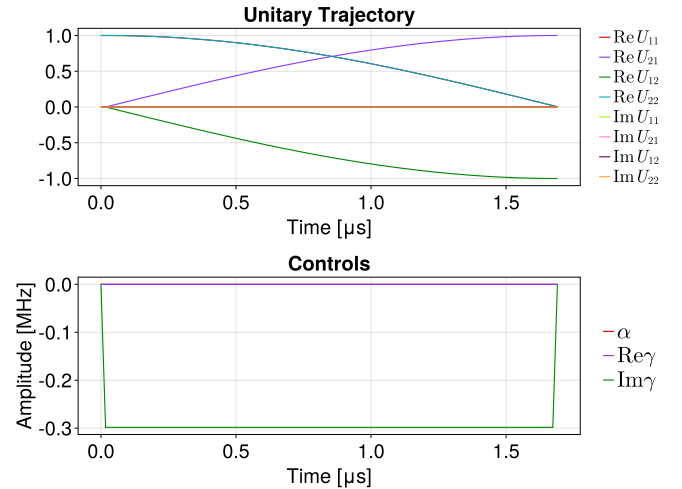


Fig. 2. **Single qubit Y-gate minimum time solution.** The top plot shows the evolution of the real and imaginary components of the unitary operator over the duration of the pulse. The bottom plot shows the pulse returned by PICO, where it is observed to have found the analytic *bang-bang* solution on its own (α and $\text{Re } \gamma$ were effectively zeroed out). The optimized pulse has a duration of 1.67 μs and a final rollout infidelity of 4.72×10^{-6} . We solved the same minimum time problem using Q-CTRL [17] which achieved a duration of 1.68 μs and infidelity of 1.44×10^{-5} as defined in Eqn. 3. Notably, unlike the PICO solution, this solution did not zero out α and $\text{Re } \gamma$.

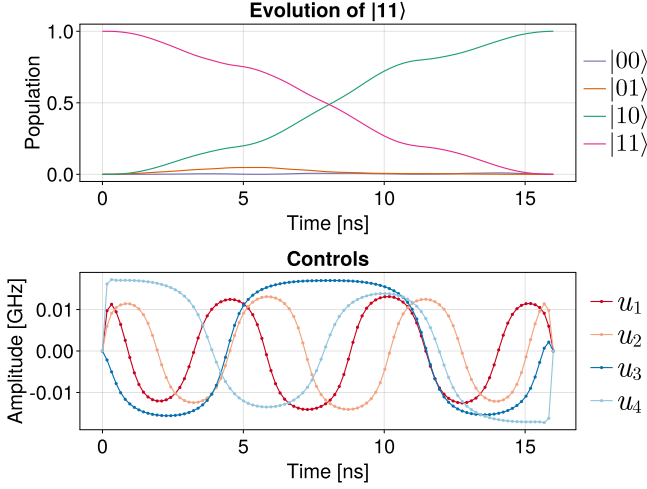


Fig. 3. **Two-qubit CNOT gate.** Optimized pulse of duration 16 ns and rollout infidelity of 2.25×10^{-8} . The top plot shows the evolution of the $|11\rangle$ state under the control pulse, while the bottom plot shows the pulse itself. The dots in the bottom plot correspond to the $N = 100$ knot points of the trajectory.

B. Two-Qubit CNOT Gate Problem

As a second example, we consider a two-qubit Hamiltonian in the rotating frame with direct qubit drives:

$$\begin{aligned}
 H(u(t)) = & g(\hat{a}^\dagger \hat{a})(\hat{b}^\dagger \hat{b}) \\
 & + u_1(t)(\hat{a} + \hat{a}^\dagger) + iu_2(t)(\hat{a} - \hat{a}^\dagger) \\
 & + u_3(t)(\hat{b} + \hat{b}^\dagger) + iu_4(t)(\hat{b} - \hat{b}^\dagger),
 \end{aligned} \quad (23)$$

where $g/2\pi = 100$ MHz and \hat{a} and \hat{b} are the annihilation operators of the first and second qubits, respectively. We optimize a CNOT gate using the free-time problem framework subject to the control bounds $|u_{i=1:4}(t)|/2\pi < 20$ MHz. The theoretical minimum CNOT gate time is given by $2\pi/g = 10$ ns, although this may not be achievable under our control bounds. Hence, as a starting point, we solve an initial free time problem with $N = 100$ knot points and an initial guess of $\Delta t_{k=1:N-1} = 0.1$ ns between knot points, subject to the constraint that all the Δt_k between knot points are equal and $0.09 < \Delta t_k < 0.17$ ns. This problem converges to a 15 ns solution with rollout infidelity 3.67×10^{-8} in just 64 solver iterations. Given this solution, we then solve another free-time problem with an initial guess of $\Delta t_k = 0.15$ and bounds $0.135 < \Delta t_k < 0.210$ ns. This problem converges to a 16 ns solution in around 500 solver iterations with a rollout infidelity of 2.25×10^{-8} as shown in Fig. 3.

This example highlights two of the compelling convergence properties of PICO: 1) dynamically infeasible initial guesses and intermediate dynamics constraint violations have the potential to enable fast convergence and 2) the favorable tail-convergence properties afforded by direct methods result in many 9s of fidelity in far fewer iterations than indirect methods.

C. Three-Qubit SWAP Gate Problem

In this example, we consider the three-qubit Hamiltonian

$$H(u(t)) = 2\pi H_0 + \sum_{j=1}^3 u^j(t) \hat{a}_j + u^j(t)^* \hat{a}_j^\dagger, \quad (24)$$

where $u(t) \in \mathbb{C}^3$, \hat{a}_j is the annihilation operator on the j -th qubit, and the drift term is,

$$\begin{aligned}
 H_0 = & \sum_{j=1}^3 (\omega_j - \omega_d) \hat{a}_j^\dagger \hat{a}_j - \frac{\xi}{2} \hat{a}_j^\dagger \hat{a}_j^\dagger \hat{a}_j \hat{a}_j \\
 & + J_{12} (\hat{a}_1^\dagger \hat{a}_2 + \hat{a}_1 \hat{a}_2^\dagger) + J_{23} (\hat{a}_2^\dagger \hat{a}_3 + \hat{a}_2 \hat{a}_3^\dagger),
 \end{aligned} \quad (25)$$

where (in units of GHz) $\omega_1 = 5.18$, $\omega_2 = 5.12$, $\omega_3 = 5.06$, $\omega_d = 5.12$, $\xi = 0.34$, and $J_{12} = J_{23} = 5.0 \times 10^{-3}$. We used 500 timesteps, where $T = 200 \mu\text{s}$, $\Delta t = T/500$, $\Delta t_{\min} = 0.5\Delta t$, and $\Delta t_{\max} = \Delta t$. We also enforced box constraints of 0.04 GHz on $|\text{Re } u^j|/2\pi$ and $|\text{Im } u^j|/2\pi$. The goal is to enact a 1 – 3 SWAP gate, i.e.

$$\begin{aligned}
 U_{\text{goal}} = & |0\rangle\langle 0| \otimes I \otimes |0\rangle\langle 0| \\
 & + |1\rangle\langle 0| \otimes I \otimes |0\rangle\langle 1| \\
 & + |0\rangle\langle 1| \otimes I \otimes |1\rangle\langle 0| \\
 & + |1\rangle\langle 1| \otimes I \otimes |1\rangle\langle 1|.
 \end{aligned} \quad (26)$$

This problem was solved using the smooth solution dynamics in (20) and knot points $z_k = (\tilde{U}_k, u_k, \dot{u}_k, \ddot{u}_k, \Delta t_k)^\top$. The real and imaginary components of \ddot{u}_k were constrained to have absolute values less than 0.05 to enforce smoothness. The solution is shown in Fig. 4; code for this solution can be found in the QuantumCollocation.jl GitHub repository.

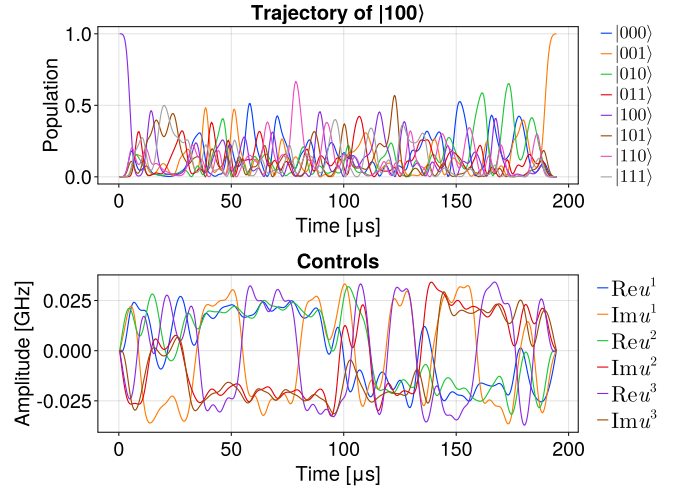


Fig. 4. **Three-qubit 1-3 SWAP gate pulse.** The top plot shows the populations of a $|100\rangle$ state, evolved via the exponential, with the controls optimized for the target 1-3 SWAP gate. The bottom plot shows the controls. This pulse achieved solver infidelity of 3.4×10^{-5} and exponential rollout infidelity of 8.6×10^{-5} with a duration of 194.56 μs .

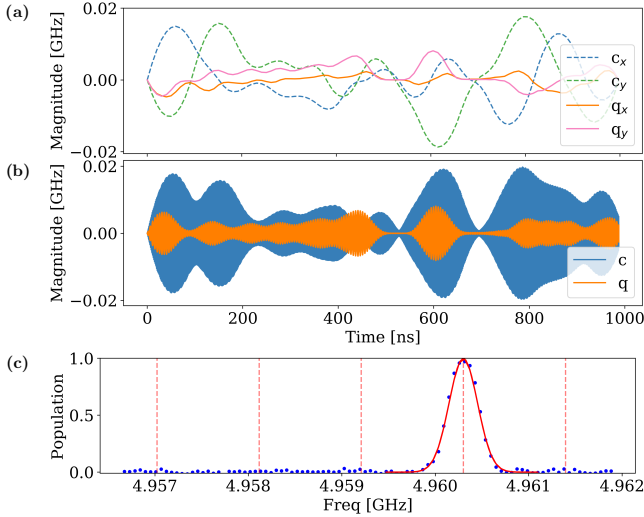


Fig. 5. **PICO pulse hardware results for a target state of $|g1\rangle$.** (a) Initial controls $u_j(t)$ are solved in a rotating frame at the qubit and cavity frequencies, such that the dispersive shift χ term dominates the Hamiltonian. Each of the cavity (c) and qubit (q) pulses are split into the real (x) and imaginary (y) quadratures to allow for effectively complex controls while still only solving with real numbers. (b) The quadratures of the cavity and qubit control pulses are recombined in the lab frame before being sent to the experiment control instrumentation. This also involves modulating at the cavity or qubit frequency. (c) Resolved qubit spectroscopy of the resulting state from the control pulses. Dashed lines indicated locations of photon-number resolved peaks so that, from right to left, they correspond to 0, 1, \dots , 4 photons. The final system state matches the target $|g1\rangle$ with fidelity 0.988 ± 0.011 .

V. HARDWARE RESULTS

In applying PICO to a hardware system a few other details were needed. Specifically, we were interested in a state transfer instead of a unitary gate. Additionally, in order to simulate the bosonic oscillator component of our hardware system, we truncated its infinite-dimensional Hilbert space. To ensure PICO did not take advantage of the artificial nonlinearity created by this truncation, we prevented the population of the highest levels in our truncated model by imposing high L_1 regularization costs on them in the problem objective. The result is a success in both simulation, as can be seen in Fig. 1, and in the outcome of the hardware experiment, as described below.

A. Experimental Outcome

Control pulses were tested on a 3D superconducting circuit-cavity QED platform that features a nonlinear transmon qubit coupled to a bosonic oscillator cavity and a readout cavity [18]. This system implements the following Hamiltonian,

$$\begin{aligned}
 H(u_q(t), u_c(t)) = & \omega_q \hat{b}^\dagger \hat{b} + \frac{\alpha}{2} \hat{b}^\dagger \hat{b} (\hat{b}^\dagger \hat{b} - 1) + \chi \hat{a}^\dagger \hat{a} \hat{b}^\dagger \hat{b} \\
 & + \omega_c \hat{a}^\dagger \hat{a} + \frac{k}{2} \hat{a}^\dagger \hat{a} (\hat{a}^\dagger \hat{a} - 1) \\
 & + u_q(t) (\hat{b} + \hat{b}^\dagger) + u_c(t) (\hat{a} + \hat{a}^\dagger),
 \end{aligned} \tag{27}$$

where \hat{b} is the annihilation operator for the transmon, $\omega_q = 2\pi \times 4.96$ GHz is the qubit $|g\rangle - |e\rangle$ transition frequency, $\alpha =$

$2\pi \times -0.143$ GHz is the anharmonicity, \hat{a} is the annihilation operator of the cavity, $\omega_c = 2\pi \times 6.22$ GHz is the cavity frequency, $\chi = 2\pi \times -1.13$ MHz is the dispersive interaction term, and $k = 2\pi \times 4.2$ kHz is the self-Kerr of the cavity. Controls $u_q(t)$ and $u_c(t)$ are time-dependent input pulses that determine the dynamics of the system. Pulses were generated by modeling three levels of the transmon and 14 levels of the cavity with L_1 costs on the highest levels.

To measure the performance of a set of control pulses, we perform photon-number-resolved qubit spectroscopy. Due to the χ dispersive shift term, individual-cavity photon-number populations change the qubit frequency and form distinguishable peaks [19]. As an example, we test solutions for state preparation of $|g1\rangle$ (transmon in the ground state, one photon in the cavity), and obtain an experimental fidelity of 0.988 ± 0.011 , as shown in Fig. 5, with the error obtained from simulating the experiment process. This fidelity is in line with state-of-the-art results [20, 21], and is close to the simulated fidelity under decoherence of 0.997. We attribute the discrepancy to slight inaccuracies or fluctuations in experimental system parameters, as well as quantization errors in the control electronics. Additional simulations showing the expected evolution and state populations at different time slices for the duration of the pulse are shown in Fig. 1 in the form of reduced traces on the transmon and cavity and Wigner tomography on the cavity.

VI. CONCLUSIONS AND FUTURE WORK

We have introduced PICO, a direct collocation method for quantum optimal control. By treating both the states and controls as decision variables, in contrast to indirect methods, PICO is able to achieve exceptional performance by leveraging state-of-the-art sparse nonlinear programming solvers like IPOPT. As a direct method, which can handle general nonlinear constraints on both the states and controls, PICO demonstrably outperforms existing indirect methods. In particular, it can solve minimum-time problems with constraints on the final state fidelity, allowing it to minimize pulse durations without sacrificing performance. PICO's other capabilities — e.g. a free-time problem formulation that allows the solver to find the optimal pulse duration — and experimental results, both in simulation and on hardware, show that is a powerful approach. PICO is open-source and available via a registered Julia package: QuantumCollocation.jl.

There are a number of exciting directions for future work. One avenue is to develop a custom nonlinear solver that takes advantage of the $SU(n)$ structure inherent in our dynamics: solvers, such as IPOPT, use techniques [22, 23, 9] that could be specialized for this setting. Another direction, which we view as crucial to hardware applications, is utilizing *iterative learning control* [24] combined with PICO to correct model-mismatch errors.

REFERENCES

- [1] Z. Manchester and S. Kuindersma, “Robust direct trajectory optimization using approximate invariant funnels,” *Autonomous Robots*, vol. 43, no. 2, pp. 375–387, 2018.
- [2] T. A. Howell, B. E. Jackson, and Z. Manchester, “Altro: A fast solver for constrained trajectory optimization,” in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2019, pp. 7674–7679.
- [3] T. A. Howell, C. Fu, and Z. Manchester, “Direct policy optimization using deterministic sampling and collocation,” 2023.
- [4] B. E. Jackson, K. Tracy, and Z. Manchester, “Planning with attitude,” *IEEE Robotics and Automation Letters*, vol. 6, no. 3, pp. 5658–5664, 2021.
- [5] R. Tedrake, *Underactuated Robotics*, 2023. [Online]. Available: <https://underactuated.csail.mit.edu>
- [6] N. Khaneja, T. Reiss, C. Kehlet, T. Schulte-Herbrüggen, and S. J. Glaser, “Optimal control of coupled spin dynamics: design of nmr pulse sequences by gradient ascent algorithms,” *Journal of Magnetic Resonance*, vol. 172, no. 2, pp. 296–305, 2005. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S1090780704003696>
- [7] T. Caneva, T. Calarco, and S. Montangero, “Chopped random-basis quantum optimization,” *Physical Review A*, vol. 84, no. 2, aug 2011. [Online]. Available: <https://doi.org/10.1103/PhysRevA.84.022326>
- [8] C. Hargraves and S. Paris, “Direct trajectory optimization using nonlinear programming and collocation,” *AIAA J. Guidance*, vol. 10, pp. 338–342, 07 1987.
- [9] A. Wächter and L. T. Biegler, “On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming,” *Mathematical programming*, vol. 106, pp. 25–57, 2006.
- [10] A. Trowbridge and A. Bhardwaj, “QuantumCollocation.jl,” Feb. 2023. [Online]. Available: <https://github.com/aarontrowbridge/QuantumCollocation.jl>
- [11] J. T. Betts, “Survey of Numerical Methods for Trajectory Optimization,” *Journal of Guidance, Control, and Dynamics*, vol. 21, no. 2, pp. 193–207, Mar. 1998. [Online]. Available: <http://arc.aiaa.org/doi/10.2514/2.4231>
- [12] N. Leung, M. Abdelhafez, J. Koch, and D. Schuster, “Speedup for quantum optimal control from automatic differentiation based on graphics processing units,” *Phys. Rev. A*, vol. 95, p. 042318, Apr 2017. [Online]. Available: <https://link.aps.org/doi/10.1103/PhysRevA.95.042318>
- [13] H. Ball, M. J. Biercuk, A. R. R. Carvalho, J. Chen, M. Hush, L. A. D. Castro, L. Li, P. J. Liebermann, H. J. Slatyer, C. Edmunds, V. Frey, C. Hempel, and A. Milne, “Software tools for quantum control: improving quantum computer performance through noise and error suppression,” *Quantum Science and Technology*, vol. 6, no. 4, p. 044011, 2021. [Online]. Available: <https://doi.org/10.1088/2058-9565/abdca6>
- [14] Q-CTRL, “Boulder Opal,” <https://q-ctrl.com/boulder-opal>, 2023, [Online].
- [15] C. B. Moler and C. V. Loan, “Nineteen dubious ways to compute the exponential of a matrix, twenty-five years later,” *SIAM Rev.*, vol. 45, pp. 3–49, 1978.
- [16] J. Cardoso and F. Silva Leite, “Theoretical and numerical considerations about padé approximants for the matrix logarithm,” *Linear Algebra and its Applications*, vol. 330, no. 1, pp. 31–42, 2001. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0024379501002518>
- [17] Q-CTRL, “Boulder Opal documentation: How to find time-optimal controls,” [/boulder-opal/application-notes/designing-robust-configurable-parallel-gates-for-large-trapped-ion-arrays](https://boulder-opal.com/application-notes/designing-robust-configurable-parallel-gates-for-large-trapped-ion-arrays), 2022, [Online; accessed 29-April-2023].
- [18] S. Chakram, K. He, A. Dixit, A. Oriani, R. Naik, N. Leung, H. Kwon, W. Ma, L. Jiang, and D. Schuster, “Multimode photon blockade,” *Nature Physics*, vol. 18, no. 1, pp. 879–884, 2022.
- [19] D. Schuster, A. Houck, J. Schreier, A. Wallraff, J. Gambetta, A. Blais, L. Frunzio, J. Majer, B. Johnson, M. Devoret *et al.*, “Resolving photon number states in a superconducting circuit,” *Nature*, vol. 445, no. 7127, pp. 515–518, 2007.
- [20] A. Eickbush, V. Sivak, A. Ding, S. Elder, S. Jha, J. Venkatraman, B. Royer, S. Girvin, R. Schoelkopf, and M. Devoret, “Fast universal control of an oscillator with weak dispersive coupling to a qubit,” *Nature Physics*, vol. 18, no. 1, pp. 1464–1469, 2022.
- [21] V. Sivak, A. Eickbusch, H. Liu, B. Royer, I. Tsioutsios, and M. Devoret, “Model-free quantum control with reinforcement learning,” *Physical Review X*, vol. 12, no. 1, p. 011059, 2022.
- [22] J. Nocedal and S. J. Wright, *Numerical optimization*. Springer, 1999.
- [23] S. Boyd, S. P. Boyd, and L. Vandenberghe, *Convex optimization*. Cambridge university press, 2004.
- [24] D. A. Bristow, M. Tharayil, and A. G. Alleyne, “A survey of iterative learning control,” *IEEE control systems magazine*, vol. 26, no. 3, pp. 96–114, 2006.