Direct Collocation for Quantum Optimal Control

Aaron Trowbridge[1], Aditya Bhardwaj[2], and Kevin He[2]

[1]Robotics Exploration Lab, Carnegie Mellon University
[2]Schuster Lab, Stanford University

Abstract

We present an adaptation of the *direct collocation* trajectory optimization method for solving problems in quantum optimal control (QOC). This approach addresses several limitations of standard methods, including the ability to solve minimum time problems, a crucial objective for realizing high-performance quantum computers. We demonstrate that this approach leads to improved performance on simulated systems as well as on nascent hardware devices, compared to other existing methods. To the best of our knowledge, this is the first time that direct collocation, which is commonplace in the field of robotic control, has been applied to QOC.

Contents

1 Introduction

Controlling quantum systems is in principle the problem of optimizing over the space of quantum state trajectories given the ability to control, over an interval of time, certain terms in the time-dependent Hamiltonian describing the system. We will consider n-dimensional quantum systems with time-dependent Hamiltonians of the form

$$H(\mathbf{a}(t), t) = H_0 + \sum_i a^i(t) H_i, \tag{1}$$

where $t \in [0, T]$, $\mathbf{a}(t) \in \mathbb{R}^m$ is the control trajectory, referred to as the *pulse*, H_0 is the system's *drift* term, and H_i are the *drive* terms.

The field of optimal control, which has its origins in space systems and robotics, has produced, especially in recent years, a large body of sophisticated methods for solving control problems fundamentally identical to the problems posed in QOC. However, many of these methods have not seemed to have been adopted by those (primarily physicists) working on control problems for quantum systems. This work aims to bridge the gap between robotic control and quantum control, and in so doing to provide a new perspective on the problem of quantum control.

1.1 Types of Problems

There are typically three flavors of QOC problems, corresponding to three types of states:

1. *Pure quantum states* $\psi(t)$: Minimize the infidelity between the final state $\psi(T)$ and the goal state ψ_{goal}:

$$\ell(\psi(T)) = 1 - |\langle \psi(T) | \psi_{\text{goal}} \rangle|^2 \tag{2}$$

   Where $\psi(0) = \psi_{\text{init}}$ and $\psi(t)$ satisfies the Schröedinger equation $\dot{\psi} = -iH(\mathbf{a}(t), t)\psi$.

2. *Unitary operators* $U(t)$: Minimize the infidelity or trace distance between the final unitary $U(T)$ and the desired final unitary U_{goal}: respectively,

$$\ell(U(T)) = 1 - \left(\operatorname{tr}\sqrt{U(T)^\dagger U_{\text{goal}}}\right)^2 \quad \text{or} \quad \ell(U(T)) = \frac{1}{2}\|U(T) - U_{\text{goal}}\|_{\text{tr}}, \tag{3}$$

   Here $U(0) = I$ and $U(t)$ also satisfies the Schröedinger equation $\dot{U} = -iH(\mathbf{a}(t), t)U$.

3. *Mixed quantum states* or *density matrices* $\rho(t)$: Minimize the infidelity or trace distance between the final state $\rho(T)$ and the goal state ρ_{goal}: respectively,

$$\ell(\rho(T)) = 1 - \left(\operatorname{tr}\sqrt{\rho(T)\rho_{\text{goal}}}\right)^2 \quad \text{or} \quad \ell(\rho(T)) = \frac{1}{2}\|\rho(T) - \rho_{\text{goal}}\|_{\text{tr}}. \tag{4}$$

   Here $\rho(0) = \rho_{\text{init}}$ and $\rho(t)$ satisfies the von Neumann equation $\dot{\rho} = -i[H(\mathbf{a}(t), t), \rho]$.

Remark: In this paper, for simplicity, we will focus on the first type of problem, and just consider the case of optimizing a single pure quantum state $\psi(t)$ — all methods discussed apply to the other two types of problems as well.

2 Background

In practice, to do any type of trajectory optimization it is necessary to discretize the time interval $[0, T]$ into N time steps of size Δt, the states and controls at each time step are denoted ψ_k and $\mathbf{a}_k$, respectively. Typically the following optimization is then solved to find the optimal pulse:

$$\underset{\mathbf{a}_{1:T-1}}{\text{minimize}} \quad J(\mathbf{a}_{1:T-1}) = \ell(\psi_N(\mathbf{a}_{1:T-1})), \tag{5}$$

$$\text{subject to} \quad c(\mathbf{a}_k) \leq 0, \quad \forall k \tag{6}$$

where

$$\psi_{k+1} = \exp\left(-iH(\mathbf{a}_k)\Delta t\right)\ \psi_k \tag{7}$$

and $\psi_1 = \psi_{\text{init}}$. Here, during the time interval Δt the controls $\mathbf{a}_k$ are held constant, which is referred to as a *zero-order hold*, and allows the dynamics to exactly take the form above.

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 optimize only over the controls $\mathbf{a}_{1:T-1}$, as opposed to considering both the states and controls as decision variables in what is known as direct methods. The advantages of direct methods, over indirect methods, are the main point of this paper. Before getting to the details of direct methods we will first introduce these two current mainstream approaches to solving QOC problems.

2.1 Gradient-Based Methods

This method involves initializing the controls $\mathbf{a}_{1:T-1}$ to some initial guess, and then iteratively updating them using a gradient descent algorithm. At each iteration is necessary to *rollout* the system with the given controls (this is also referred to as a *shooting method*) to get ψ_N and evaluate the cost function $J(\mathbf{a})$ to be able to take a gradient of it and update the controls:

$$\mathbf{a} \leftarrow \mathbf{a} - \beta \nabla J(\mathbf{a}) \tag{8}$$

There are efficient ways to compute this gradient, but the approach is still limited in other ways. These include the fact that the solution is dependent on the initial guess, gradient-based methods are in general prone to falling into local minima, and it is not possible to enforce constraints on the states.

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

2.2 Gradient-Free Methods

For this class of approaches, the key idea is to reduce the number of decision variables enough so that gradient-free optimization algorithms—e.g. the Nelder-Mead downhill simplex method—can be utilized. In the case of the popular CRAB algorithm (also implemented in QuTiP and compared against later on) this is accomplished by parameterizing the pulse by a set of basis functions. The CRAB algorithm, specifically, utilizes a Fourier basis parameterization for each pulse component:

$$a(t) = 1 + \frac{\sum_{n=1}^{N_c} b_n \sin(\omega_n t) + c_n \cos(\omega_n t)}{\lambda(t)}. \tag{9}$$

Where N_c is the number of terms to *chop* the series off at and $\lambda(t)$ is chosen to enforce the boundary conditions. Standard gradient-free methods can then be used to solve the new optimization problem:

$$\underset{b_{1:N_c}, c_{1:N_c}, \omega_{1:N_c}}{\text{minimize}} \quad J(b_{1:N_c}, c_{1:N_c}, \omega_{1:N_c}). \tag{10}$$

3 Direct Collocation

3.1 State Transfer Problems

3.2 Free Time Problems

3.3 Minimum Time Problems

4 Quantum Control Problems

4.1 Isomorphic Problem Formulation

4.2 Padé Integrator Dynamics

4.3 Special Considerations

5 Experimental Results