Robust Quantum Optimal Control

Thomas Propson*
Department of Physics, University of Chicago, Chicago, Illinois 60637, USA
(Dated: July 16, 2020)

This is a paper about robust quantum optimal control.

Introduction

(Existing work) The leading model of universal quantum computation is gate-based. There are analytic techniques to construct gates [**?** **?** **?** **?**]. Most methods focus on optimizing a few aspects of the gate, but not all experimentally relevant aspects. There is a growing literature on numerical techniques to construct gates [**?** **?**]. These methods formulate the quantum optimal control problem as unconstrained. They typically rely on zeroth-order or first-order optimizers. They are not sofisticated enough to handle all of the relevant constraints simultaneously. Although some analytic techniques exist to design pulses robust to decoherence, no numerical techniques have been presented to design pulses robust to decoherence (as far as the author is aware).

(This work) We employ the trajectory optimization literature to formulate the quantum optimal control problem as a constrained optimization problem. We study the quantum optimal control problem on the fluxonium. We outline experimentally realistic constraints and map them to the trajectory optimization framework. For the device we study we achieve a 2x increase in T_1 times. We present two methods for achieving robustness to system parameter deviations, and compare to existing dynamic decoupling methods. We find that our methods beat dynamic decoupling and mitigate dephasing by order X.

(Outline) First we formulate the quantum optimal control problem in the trajectory optimization framework. Then, we introduce the dynamics of the fluxonium device and outline experimental considerations relevant to gate construction. Next we outline a method for making the optimization T_1 aware. Finally, we present some methods for engineering robustness to decoherence and compare them to existing techniques.

QOC + AL-iLQR

(QOC Problem Statement) I have some initial configuration and I want to reach some target configuration, either single- or multi-state transfer, subject to the dynamics

$$\partial_t |\psi(t)\rangle = -\frac{i}{\hbar} H(t) |\psi(t)\rangle \tag{1}$$

Common numerical techniques include approximating the analytic unitary propagator solution, or employing explicit/implicit Runge-Kutta methods of the form

$$|\psi^{(k+1)}\rangle = |\psi^{(k)}\rangle + \partial_t |\psi^{(k)}\rangle \cdot dt \tag{2}$$

The interesting part is that your hamiltonian $H(t)$ has a time-dependent control parameter $u(t)$ that the experimentalist gets to control, e.g. flux threading a superconducting junction.

$$H(t) = H_0 + \sum_{i=1}^{m} a_m(u_m(t)) H_m \tag{3}$$

Most often $a_m(u) = u$ but any arbitrary dependence is allowed. The technique of adjusting $|\psi_N\rangle$ based on $\mathbf{u}(t)$ is the goal of the quantum optimal control optimizer. This domain is know as sensitivity analysis. GRAPE does

$$\min_{U:N-1\times m} 1 - |\langle \psi_N(U) | \psi_t \rangle|^2 \tag{4}$$

subject to the gradient descent update procedure

$$U' = U - \alpha \nabla_U (1 - |\langle \psi_N(U) | \psi_t \rangle|^2) \tag{5}$$

In general they add more cost functions than target fidelity.

(AL-iLQR Problem Statement) Trajectory optimization gives us gaurantees about our updates via Ricatti recursion and allows us to put constraints on

* tcpropson@uchicago.edu

our cost functions.

$$\min_{U:N-1\times m} \mathcal{L}_N(x_N, \lambda_N, \mu_N) + \sum_{k=1}^{N-1} \mathcal{L}_k(x_k, u_k, \lambda_k, \mu_k) \tag{6}$$

$$\mathcal{L}_k(x_k,u_k, \lambda_k, \mu_k) = (\lambda_k + \frac{1}{2} I_{\mu_k} c_k(x_k, u_k))^T c_k(x_k, u_k) + (x_k - x_f)^T Q (x_k - x_f) + u_k^T R u_k \tag{7}$$

The important point is that there is an update step (e.q. 17 from ALTRO paper) where we send $\lambda \to \infty$ and get all of the nice convergence properties. The weights are adjusted dynamically between iterations until all of our constraints are satisfied. The Markovian decision structure of the problem allows us to apply differentiable dynamic programming to garauntee that the update for each control U_k is opimal, as apposed to the greedy updates of first-order optimizers like the naive gradient descent.

Our Problem

(Fluxonium Device) In the two-level approximation we have

$$H/h = \omega_q \frac{\sigma_z}{2} + A(\Phi_{ext}) \frac{\sigma_x}{2} \tag{8}$$

This approximation is good up to the avoided crossing at $\Phi_{\text{ext}} = 0.35\Phi_0$. We get A by converting via $\langle g| \hat{\phi} |e\rangle$. We use the gates $X/2, Y/2, Z/2$. $X/2, Y/2$ are universal up to an arbitrary Z rotation.

(Constraints) We want constraints on our pulses. We want pulses start and end at zero for concatenation. We want pulses to have zero net flux to mitigate hysteresis in flux bias lines. We want the amplitude to be constrained $\delta\Phi_{\text{ext}} \sim 0.06\Phi_0$ so the two-level approximation stays valid. We want our state to obey normalization conditions, mitigating numerical error in simulaiton.

(T_1 and T_ϕ noise) Dissipation to the thermal bath via longitudinal relaxation (T_1) is an irreversible process that results in information loss. Pure dephasing (T_ϕ) is a reversible process. Dielectric loss, etc. contribute to T_1. $1/f$ flux noise, etc. contribute to T_ϕ There is a tradeoff between T_1 and T_ϕ. In the case of white noise we have that the sum of the noise weights W_1 and W_ϕ is constant [**?**]. In our case T_ϕ is saturated at the flux frustration point $\Phi_{\text{ext}} = 0.5\Phi_0$ T_ϕ decreases as we move away from the flux frustration point. Conversely, T_1 is at a minimum around the flux frustration point $T_1(0.5\Phi_0) = 0.315$ms, and increases away from the flux frustration point $T_1(0.43\Phi_0) = 4.3$ms. Given the nature of the decay processes and the tradeoff, we choose to optimize the flux bias to have high time-averaged T_1 times and employ some techniques to mitigate T_ϕ dephasing.

Robustness to T_1-type Noise

(Strategy) We seek to minimize the probability that the qubit relaxes longitudinally. To this end we augment the state vector x_k with the longitudinal relaxation probability

$$P_1(t_k) = \int_0^{t_k} \gamma_1(\Phi_{\text{ext}}(t)) dt \tag{9}$$

where $\gamma_1 = T_1^{-1}$. Setting the target longitudinal relaxation probability to 0 results in a quadratic cost at each knot point of the form $|P_1(t_k)|^2$.

The value $T_1(\Phi_{\text{ext}}(t_k))$ is obtained at each knot point by evaluating a spline interpolant fit to experimentally obtained data of the form $\{(\Phi_{\text{ext}}, T_1)\}$. Calculating T_1 directly from theoretical considerations requires many high-dimensional eigendecompositions, which is computationally expensive. Additionally, T_1 values are known to fluctuate greatly with laboratory temperatures [**?**]. Interpolating T_1 from experimental data increases the fridge truth of the simulation.

Furthermore, the probability of longitudinal relaxation is dependent on the gate duration t_N. We allow the optimizer to tune the gate duration by augmenting the control vector u_k with the time step between knot points Δt_k. Promoting Δt_k to a decision variable, rather than the number of knot points N, preserves the Markovian decision structure of the trajectory optimization problem. To ensure numerical integration accuracy is maintained we add a bound constraint at each knot point 5e-3 ns $\leq \Delta t_k \leq$ 2e-2 ns. Note that this bound constraint is allowed to be broken for intermediate iterations of the optimization, so we use the absolute value of the time step $|\Delta t_k|$ to ensure that it is non-negative.

(Results) We achieve a factor of 1.5 decrease in the probability of longitudinal relaxation from the analytic gates we benchmark against. We simulate the performance of the gates using the lindblad master equation (see Appendix X). We repeatedly apply the basis gates and measure the fidelity of the resulting state as a function of time as shown in Fig. 1.

Gate	Analytic P_1 (10^{-5})	QOC P_1 (10^{-5})	Speedup
Z/2	4.615	2.576	1.792
Y/2	2.826	1.888	1.496
X/2	9.562	5.080	1.882

Table I: Probability of longitudinal relaxation for each gate evaluated at the gate's duration.

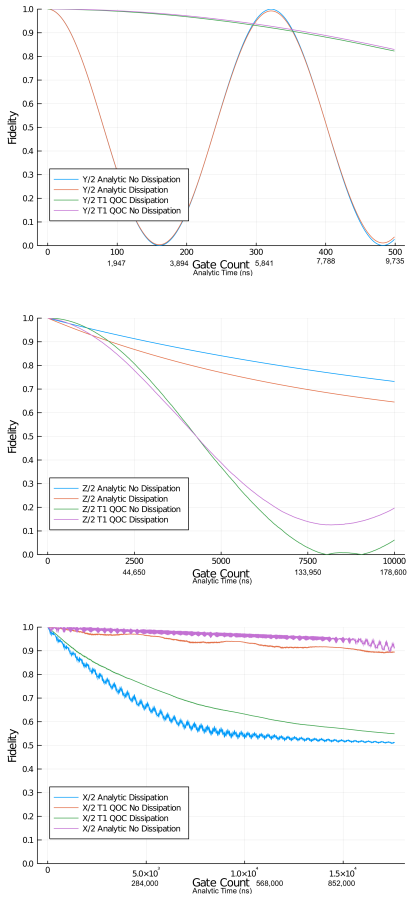

Figure 1: Master equation simulation with T_1 dissipation for all basis gates.

Robustness to T_ϕ-type Noise

(Strategy) System parameter deviations arise due to measurement error and noise injection. For example, we might measure ω_q inaccurately, which leads to a deviation from the the drive frequency ω_d. Further, we may have some temperature dependent gain in the lines leading to the device, or $1/f$ noise and $A(\Phi_{\text{ext}})$ comes out with some δA on the flux axis. Both of these effects lead to dephasing, uncertain dynamics, and thus erroneous computation.

Figure 2: fidelity v.s. ω for the $X/2$ gate via analytic, 3-sample, 2-derivative (maybe add 5-sample and 3-derivative)

Figure 3: fidelity at $\omega \pm \sigma_\omega$ v.s. time for the $X/2$ gate via analytic, 3-sample, 2-derivative (maybe add 5-sample and 3-derivative)

These problems are typically attacked with dynamic decoupling sequence where an erroneous rotation is compensated for with one that cancels the δ. However, we can do better than the first order variants of these pulses.

We propose two methods for engineering robustness to system parameter deviations. The first we call the derivative method. We use the intuition that making the final state insensitive to changes in the system parameter is encoded in the derivative $\partial_\omega^l |\psi_N\rangle$. In particular, we find that going to the second derivative is good. The first derivative is already small, especially for the δA time, because, near the end of optimization, the optimizer is unable to improve the final state by changing the flux bias. The dynamics for propgating the derivative of the state are found by differentiating the schroedinger equation dynamics with respect to the parameter of interest. The scheme boils down to propagating sequential derivatives of the parameter you want to be robust to ∂_ω^l and penalizing thier norm $|\partial_\omega^l|^2$.

The other scheme we call the sampling method. This method is well-known in the trajectory optimiztion. In this scheme we propgate an array of states whose dynamics differ in that the parameter of interest is altered slightly for each state. In the 3-point sampling method we propagate $\psi_\pm$, ψ where $\omega_\pm = \omega \pm \delta\omega$. Here δ is typically taken at standard deviations of the parameter of interest.

(Results) We compare the simplest schemes for the derivative and sampling methods to the CORPSE dynamic decoupling sequence. They perform X better by Y metric.

We also find that longer gate durations allow for greater robustness.

Figure 4: Master equation simulation with T_2 dissipation comparing the gates robust to $\delta\omega_q$ and δA against the analytic gate.

Discussion

We have proposed some schemes and they work well.

Ricatti Recursion

This will give the reader unfamiliar with trajectory optimization intuition for how the trajectory optimization update scheme works and why it is better than a more naive method.

Measurements

We measure T_1 using the standard experiment and T_2 using te Ramsey experiment. We fit with splines and the data looks like fig. 3 in Helin's paper [**?**]. We measure ω_q and σ_{ω_q} using X method.

ME Simulation

We model dissipation using the lindblad master equation and standard collapse operators for longitudinal relaxation and dephasing of a two-level system.

Derivative Method

$$\begin{aligned}\partial_\omega^2 \partial_t \psi = \frac{-i}{h}[2 \cdot \frac{\sigma_z}{2}\partial_\omega \psi \\ + (\omega\frac{\sigma_z}{2} + A\frac{\sigma_x}{2})\partial_\omega^2 \psi]\end{aligned} \tag{10}$$