Optimal Control Theory

Variational Quantum Algorithms · Module 7 — Control Theoretic View

Node 7.4

Position in Knowledge Graph

System: Variational Quantum Algorithms **Structural Domain:** Control-Theoretic View **Node:** 7.4

This is the **central technical** node of Module 7. Where 7.2 and 7.3 introduced observability and controllability as *structural* properties, optimal control theory is the framework for *choosing* the control input that best achieves a goal. It is the variational calculus of control systems, and it is what justifies casting VQA optimization as a control problem in the first place.

The connection is direct: **VQA parameter optimization is a discrete-time optimal control problem**. Every theorem and algorithm in optimal control theory has a VQA analogue. This node walks through the framework, the major algorithmic families (LQR, indirect methods, direct methods, GRAPE), and the specific quantum control variants (Krotov, CRAB, gradient ascent pulse engineering).

The thesis's view is that **the gap between optimal control theory and VQA optimization is mostly notational** — the techniques that have been developed for half a century in classical and quantum control are exactly the techniques VQA practitioners are slowly rediscovering. Module 7 makes the import explicit.

The Optimal Control Problem

The general setup:

- **State:** $\mathbf{x}(t) \in \mathcal{X}$, with dynamics $\dot{\mathbf{x}} = f(\mathbf{x}, \mathbf{u}, t)$.
- **Control:** $\mathbf{u}(t) \in \mathcal{U}$, to be chosen.
- **Initial condition:** $\mathbf{x}(0) = \mathbf{x}_0$.
- **Terminal time:** T (fixed or free).
- **Cost functional:**

$$J[\mathbf{u}] = \phi(\mathbf{x}(T)) + \int_0^T L(\mathbf{x}(t), \mathbf{u}(t), t)\, dt,$$

where ϕ is a **terminal cost** and L is a **running cost**.

Problem. Find $\mathbf{u}^*(t)$ that minimizes J, subject to the dynamics and any constraints on $\mathbf{x}, \mathbf{u}$.

This is the **canonical optimal control problem**. It generalizes the calculus of variations (where there is no control input separate from the state) and is the parent framework for everything in this node.

Linear Quadratic Regulator (LQR)

The simplest non-trivial case: linear dynamics, quadratic cost, no constraints.

$$\dot{\mathbf{x}} = A\mathbf{x} + B\mathbf{u}, \quad J = \mathbf{x}(T)^T S\mathbf{x}(T) + \int_0^T (\mathbf{x}^T Q\mathbf{x} + \mathbf{u}^T R\mathbf{u})\, dt.$$

Theorem. The optimal control is a linear feedback law $\mathbf{u}^*(t) = -K(t)\mathbf{x}(t)$, where $K(t) = R^{-1}B^T P(t)$ and $P(t)$ satisfies the **Riccati differential equation**

$$-\dot{P} = A^T P + PA - PBR^{-1}B^T P + Q, \quad P(T) = S.$$

Implications:

1. **The optimal control is linear in the state** — a clean closed-form result.
2. **The Riccati equation is solvable in closed form** for many cases — gives explicit control laws.
3. **The result generalizes** to discrete time, infinite horizon, and stochastic noise (Linear Quadratic Gaussian, LQG).

For VQAs: the analogue is "linear ansatz, quadratic cost." Both assumptions are wrong for typical VQAs (the ansatz is highly nonlinear in parameters; the cost is non-convex). So LQR doesn't apply directly. But it serves as the **theoretical limit case** — the easiest possible optimal control problem, against which to measure the difficulty of harder ones.

Indirect Methods: The Hamiltonian Formulation

For nonlinear problems, the standard approach is via the **Pontryagin minimum principle** (covered in detail in node 7.5). The idea: introduce a **costate** $\boldsymbol{\lambda}(t)$ (Lagrange multiplier for the dynamics constraint) and form the control Hamiltonian

$$H(\mathbf{x}, \mathbf{u}, \boldsymbol{\lambda}, t) = L(\mathbf{x}, \mathbf{u}, t) + \boldsymbol{\lambda}^T f(\mathbf{x}, \mathbf{u}, t).$$

The optimal trajectory satisfies:

$$\dot{\mathbf{x}} = \partial H/\partial \boldsymbol{\lambda}, \quad \dot{\boldsymbol{\lambda}} = -\partial H/\partial \mathbf{x}, \quad \mathbf{u}^* = \arg\min_{\mathbf{u}} H.$$

This is a **two-point boundary value problem** (the state is given at $t = 0$, the costate at $t = T$) — harder to solve numerically than an initial-value problem, but in principle gives the exact optimum.

Algorithms: shooting methods, multiple shooting, collocation. All require careful initialization because the Hamiltonian formulation is sensitive to numerical issues.

For VQAs, the indirect approach corresponds to writing down the parameter-shift gradient conditions and solving them as a system of nonlinear equations. Practically, this is rarely done — gradient descent (a direct method) is simpler.

Direct Methods: Discretize First, Optimize Second

The alternative: discretize the problem in time and turn it into a finite-dimensional optimization.

Procedure: 1. Discretize time into N intervals: $t_k = k\Delta t$. 2. Approximate the state and control as piecewise functions: $\mathbf{x}_k = \mathbf{x}(t_k), \mathbf{u}_k = \mathbf{u}(t_k)$. 3. Discretize the dynamics: $\mathbf{x}_{k+1} = \mathbf{x}_k + \Delta t \cdot f(\mathbf{x}_k, \mathbf{u}_k)$ (Euler) or higher-order schemes. 4. Discretize the cost: $J \approx \phi(\mathbf{x}_N) + \Delta t \sum_k L(\mathbf{x}_k, \mathbf{u}_k)$. 5. Optimize over $\{\mathbf{u}_0, \ldots, \mathbf{u}_{N-1}\}$ directly, treating the dynamics as constraints (or implicit functions).

This is a **constrained nonlinear program** with Nm variables (where m is the control dimension). Solvable by gradient methods, sequential quadratic programming, or interior point.

For VQAs: this is exactly what VQA optimization is. The control intervals are the gates; the control variables are the parameters; the dynamics are the unitary evolution; the cost is the final expectation value. The "discretize first, optimize second" approach is the unstated framework of all VQA papers.

Quantum Optimal Control: GRAPE

The dominant algorithm for *continuous-pulse* quantum optimal control: **Gradient Ascent Pulse Engineering** (Khaneja et al. 2005).

Setup. Time evolution under a Hamiltonian $H(t) = H_0 + \sum_j u_j(t) H_j$. Discretize $u_j(t)$ into N piecewise-constant intervals. The total unitary is

$$U_{\text{total}} = \prod_{k=0}^{N-1} \exp\left(-i\Delta t \left(H_0 + \sum_j u_{j,k} H_j\right)\right).$$

Goal. Find $\{u_{j,k}\}$ that maximizes the fidelity $|\mathrm{Tr}(U_T^\dagger U_{\text{total}})|^2/d^2$ to a target unitary U_T, or that prepares a target state.

GRAPE update rule. Compute the gradient of the fidelity with respect to each $u_{j,k}$ using the analytical formula

$$\frac{\partial F}{\partial u_{j,k}} = -i\Delta t \cdot \mathrm{Re}\left[\mathrm{Tr}\left(\rho^\dagger H_j U_{\text{total}}\right)\right] + O(\Delta t^2),$$

where ρ is a reference state, then take a gradient step.

Compared to VQA parameter-shift: GRAPE uses an *approximate* gradient (the $O(\Delta t^2)$ error term) but is much faster per evaluation (no parameter-shift, just one forward simulation). For continuous quantum control, this tradeoff is worth it. For VQAs (where parameter-shift is exact and shot noise is the bottleneck), the tradeoff is different.

The link: GRAPE on a fine discretization is equivalent to VQA optimization with an HEA-like ansatz. The same algorithm, just at a different level of abstraction.

Krotov And CRAB

Two other major quantum control algorithms.

Krotov

A **monotonic** algorithm: each iteration is guaranteed to decrease the cost (no step size to tune). Works by iteratively solving a coupled system of forward (state) and backward (costate) equations.

Strengths: monotonic convergence, no step size, theoretically clean. **Weaknesses:** slower per iteration than GRAPE, harder to implement, requires the full Hamiltonian to be known.

CRAB (Chopped Random Basis)

Parameterizes the control $u_j(t)$ in a *low-dimensional* basis (e.g., truncated Fourier series) and optimizes over the basis coefficients.

Strengths: few parameters, robust to local minima, easy to implement. **Weaknesses:** limited expressivity (can only represent functions in the chosen basis).

Connection to VQAs: CRAB is essentially "VQA with a smart parameterization." The choice of basis is the analogue of ansatz design. CRAB has been successful in experimental quantum control settings, and its lessons (smart parameterization, low-dimensional optimization) translate directly to ansatz design for VQAs.

The VQA Optimal Control View

Putting it all together, here is the **VQA optimal control view**:

VQA concept	Optimal control concept
Parameters $\boldsymbol{\theta}$	Discretized control input $\{u_{j,k}\}$
Ansatz $U(\boldsymbol{\theta})$	Discretized propagator $U_{\text{total}}(\mathbf{u})$
Cost $C(\boldsymbol{\theta})$	Terminal cost $\phi(\mathbf{x}(T))$
Parameter-shift gradient	Exact GRAPE-like gradient
Optimizer (Adam, etc.)	Outer loop of GRAPE/Krotov
Ansatz design	Choice of control parameterization (CRAB basis)
Barren plateau	Vanishing gradient in the control optimization
Symmetry restriction	Constrained optimization in the control space

Implication: every VQA optimization technique has a quantum optimal control analogue, and vice versa. The fields developed in parallel without much cross-talk, and the bridge is exactly this node.

The thesis emphasizes the bridge because it gives access to:

1. **Convergence theorems** from optimal control (e.g., Krotov's monotonicity).
2. **Robustness techniques** from robust optimal control (H_∞, μ-synthesis).
3. **Time-optimal results** from optimal control (Khaneja-Brockett-Glaser for two-qubit gates).
4. **Adaptive control schemes** from adaptive optimal control (Module 10).

Specific Theorems Worth Knowing

A few theorems from optimal control with direct VQA application:

1. Pontryagin minimum principle (covered in 7.5). Necessary conditions for optimality. Applies to any optimal control problem with smooth dynamics and cost.

2. Hamilton-Jacobi-Bellman equation. A PDE for the value function (the optimal cost-to-go). Sufficient conditions for optimality. Solving it directly is usually intractable, but it justifies dynamic programming approaches.

3. Bellman's principle of optimality. "An optimal policy has the property that whatever the initial state and decision are, the remaining decisions must constitute an optimal policy with regard to the state resulting from the first decision." The basis of dynamic programming.

4. Lyapunov stability. Conditions for the control system to converge to a desired state. For VQAs, gives criteria for when an optimizer is guaranteed to converge.

5. Separation principle. For LQG problems, the optimal controller can be decomposed into a state estimator and a feedback law. Not directly applicable to nonlinear quantum problems but inspires the architecture of adaptive VQAs.

These theorems are all in standard optimal control textbooks (e.g., Lewis-Vrabie-Syrmos, Bertsekas). Most VQA practitioners don't read them, which is a missed opportunity.

Limitations Of The Optimal Control Framing

The optimal control view has limitations:

1. Stochasticity. Standard optimal control assumes deterministic dynamics. VQAs have stochastic measurements, which makes them stochastic optimal control problems — a more complex framework.

2. High dimensionality. Optimal control techniques are designed for low-dimensional state spaces ($\mathcal{X} = \mathbb{R}^n$ for small n). The quantum state space has dimension 2^n, exponentially large.

3. Discrete nature. VQAs have discrete circuit structure (gates), not continuous controls. Some optimal control results don't translate cleanly.

4. Non-convexity. Optimal control results that assume convexity (LQR, certain Riccati decompositions) don't apply.

5. Hardware-specific constraints. Real VQA controls are constrained by gate fidelities, connectivity, and timing in ways that classical optimal control doesn't model.

These limitations mean the import is *partial* — not every theorem applies. But the framework, vocabulary, and most of the algorithmic ideas do apply, and the exceptions are usually known and discussed.

Structural Role

This node is the **algorithmic core** of Module 7. It introduces optimal control theory in its full generality, surveys the major algorithmic families (LQR, indirect, direct, GRAPE, Krotov, CRAB), and establishes the explicit mapping from optimal control concepts to VQA concepts.

It is the prerequisite for node 7.5 (Pontryagin minimum principle, the necessary conditions) and 7.6 (feedback control, the closed-loop generalization).

Relations to Other Concepts

- **Pontryagin et al. (1962)** — *The Mathematical Theory of Optimal Processes*, the foundational text.
- **Khaneja et al. (2005)** — GRAPE algorithm.
- **Krotov (1995)** — monotonic optimal control method.
- **Doria et al. (2011)** — CRAB algorithm.
- **Bellman (1957)** — *Dynamic Programming*, the principle of optimality.

Worked Example — VQA As GRAPE

Consider a 4-qubit VQA with HEA ansatz, depth 5, total parameters $P = 60$. Cost: ground-state energy of an H_2 Hamiltonian.

As an optimal control problem:

- **State:** $|\psi(t)\rangle \in \mathbb{C}^{16}$, evolving under the discretized circuit.
- **Control variables:** the 60 parameters, partitioned into time intervals (one parameter per gate).
- **Dynamics:** unitary evolution under the gate Hamiltonians.
- **Terminal cost:** $\phi = \langle\psi(T)|H_{H_2}|\psi(T)\rangle$.
- **Running cost:** zero (no path penalty).

As a GRAPE problem: with continuous-time control, this would be solved by computing the gradient of the fidelity at each time interval and gradient-stepping. The discretized version is exactly the parameter-shift VQA optimizer.

Differences from standard GRAPE:

1. The "time" is discrete (gates) rather than continuous (pulses).
2. The gradient is exact (parameter shift) rather than approximate (GRAPE finite difference).
3. The cost is estimated from finite shots, not computed exactly.

Differences captured by the same framework: all the GRAPE convergence theory, monotonicity guarantees, basin-hopping initialization, and warm-start strategies apply with minor modifications.

This is the kind of analysis the optimal control framing makes natural and the optimization-only framing misses.

Visualization

Pending. A flow diagram showing the optimal control problem (state, control, dynamics, cost, optimizer loop) with VQA labels in parentheses. Caption: "The same diagram, two vocabularies."

Exercises

1. For an LQR problem with $A = 0, B = 1, Q = 1, R = 1, S = 0$, solve the Riccati equation and write down the optimal feedback law.
2. The GRAPE gradient formula has an $O(\Delta t^2)$ error. Why is this acceptable for continuous control but not for VQA parameter-shift?
3. (VQA) For a QAOA-style ansatz with cost and mixer Hamiltonians, write the optimal control formulation explicitly. What is the running cost?

Research Extensions

- **VQA optimization via Krotov** — applying monotonic methods to VQA loss functions.
- **CRAB-style basis design for VQA** — choosing parameterization bases that improve trainability.
- **Stochastic optimal control for VQAs** — formal extensions to handle shot noise.
- **HJB equation for VQA value function** — partial dynamic-programming approaches.

Cross-links to Other Courses

- **Calculus of Variations** — the mathematical heritage of optimal control.
- **Pontryagin Minimum Principle** — the necessary conditions, in node 7.5.
- **Quantum Optimal Control** — GRAPE, Krotov, CRAB.
- **Module 6 (Optimization Dynamics)** — the optimization-only view.
- **Module 7.5 (Pontryagin)** — the next node.

Variational Quantum Algorithms · Module 7 — Control Theoretic View · Node 7.4
Concepts: control-theory, cost-function, constrained-optimization, variational-principle
Prerequisites: 7.1, 7.2, 7.3
Enables: 7.5, 7.6

hbar.university — own.your.cognition