VQA for Quantum Control

Variational Quantum Algorithms · Module 10 — Vqa As Adaptive Control Systems

Node 10.4

Position in Knowledge Graph

System: Variational Quantum Algorithms **Structural Domain:** VQA as Adaptive Control Systems **Node:** 10.4

Module 7 (Control-Theoretic View) established that a VQA *is* a control system — the optimizer navigates a parameter landscape under noise and constraints, much like a classical controller navigating a plant. This node inverts the relationship: instead of using control theory to *explain* VQAs, it uses VQAs as a *tool* for solving quantum control problems.

Quantum control is the engineering problem of designing time-dependent Hamiltonians or pulse sequences to achieve a target operation: prepare a specific state, implement a specific gate, transport a specific excitation, etc. This is a classical optimal control problem in a quantum context — the target is an element of a Hilbert space or unitary group, but the optimization is over classical control functions.

The key insight for this node: **the same variational machinery that solves chemistry or combinatorial problems can also solve quantum control problems**. You parameterize the control pulse, define a cost that measures distance to the target, and optimize. The VQA is then a solver for quantum control.

This is not a reinterpretation — it's a direct application. Variational quantum control (VQC) is a substantial subfield with its own literature (Khaneja-Brockett-Glaser, D'Alessandro, Sugny-Bonnard). Recent work has merged it with the VQA literature.

This node covers the main applications:

1. **Variational gate synthesis** — using VQA machinery to design optimal pulse sequences for a target gate.
2. **State preparation via VQA** — preparing a target quantum state by variational pulse design.
3. **Variational quantum simulation of control problems** — when the controlled system is itself a quantum device.
4. **Feedback-aware variational control** — integrating real-time feedback into VQA-based control.
5. **Hybrid control-VQA architectures** — using VQAs inside a larger control system.

Variational Gate Synthesis

Problem. Given a target unitary U_{target} and a parameterized pulse family $H(\boldsymbol{\theta}, t)$, find the pulse parameters that implement U_{target} as accurately as possible.

VQA formulation.

- **Ansatz:** a time-discretized circuit where the gate at time step k is $\exp(-iH(\theta_k)\Delta t)$. The parameters $\boldsymbol{\theta}$ are the pulse amplitudes.
- **Cost:** negative average gate fidelity $C = -|\text{Tr}(U_{\text{target}}^\dagger U(\boldsymbol{\theta}))|^2/d^2$.
- **Optimizer:** gradient descent (often GRAPE-style, which *is* a VQA optimizer in disguise).

Connection to VQA. This is literally a VQA: parameterized quantum circuit, classical optimizer, cost function. The difference from chemistry VQAs is the cost function — instead of an energy expectation, it's a gate fidelity. Everything else (optimization, noise handling, convergence) is the same.

Advantage. The VQA framework lets you use standard optimization toolboxes (Adam, HOPSO, natural gradient) for gate design. Early quantum control relied on GRAPE or Krotov, which are specialized; modern work uses VQA-style optimizers.

Example. An 8-qubit toffoli gate design on superconducting hardware: parameterize a 40-slot time grid of 2-qubit Pauli terms, optimize via natural gradient, reach 99.5% fidelity in ~300 iterations. This is competitive with hand-designed Toffoli decompositions.

State Preparation Via VQA

Problem. Prepare a specific quantum state $|\psi_{\text{target}}\rangle$ from a known initial state $|\psi_0\rangle$ by variational pulse or gate design.

VQA formulation.

- **Ansatz:** a parameterized circuit (could be HEA, UCC, or pulse-level).
- **Cost:** $C = 1 - |\langle\psi_{\text{target}}|\psi(\boldsymbol{\theta})\rangle|^2$, the infidelity.
- **Optimizer:** standard VQA optimizer.

Connection to VQA. State preparation is identical to the "ground state preparation" task at the heart of VQE — except that the target is known exactly, not implicitly through a Hamiltonian.

Advantage over non-variational methods. Variational state preparation can be *shorter* than explicit unitary decomposition (which requires $O(2^n)$ gates in the worst case). A variational ansatz with $O(\text{poly}(n))$ parameters can approximate many states to high accuracy.

Caveat. Which states are approximable depends on the ansatz's expressibility (Module 8). Not all states can be efficiently prepared by a polynomial-depth VQA.

Example. Preparing a 10-qubit W state (superposition over single-excitation states) via variational circuit. A depth-5 HEA reaches 99% fidelity. A direct construction needs 20+ gates.

Variational Quantum Simulation Of Control Problems

Problem. Simulate the dynamics of a quantum system under a control Hamiltonian, for use in designing the control.

Why VQAs help. Control design often requires simulating the controlled system's dynamics. If the system is a many-body quantum system, classical simulation is expensive. A *variational quantum simulator* can simulate the dynamics using a smaller quantum device with variational ansatz.

Architecture.

1. Parameterize an ansatz $|\phi(\boldsymbol{\theta})\rangle$ that represents the current state of the controlled system.
2. Update $\boldsymbol{\theta}(t)$ using McLachlan's variational principle: $d\boldsymbol{\theta}/dt = A^{-1}b$ where A, b depend on the ansatz and the Hamiltonian.
3. At each time step, evaluate the current observable (e.g., fidelity to target).
4. Use the observable as the cost for classical optimization of the control.

Reference. Li-Benjamin (2017) introduced variational imaginary time evolution. McArdle et al. (2019) extended to real time.

Use case. Designing control pulses for systems larger than classical simulators can handle (30+ qubits). The quantum device simulates the dynamics while a classical optimizer tunes the control.

Feedback-Aware Variational Control

Problem. The standard VQA for quantum control produces an *open-loop* pulse sequence — a pre-computed schedule that's applied regardless of feedback. Real quantum control often benefits from *closed-loop* feedback — observe the current state, adjust the control.

VQA formulation with feedback.

- **Ansatz:** a parameterized circuit that *includes* mid-circuit measurements (as in Module 10.2).
- **Cost:** expectation over all measurement trajectories.
- **Optimizer:** standard, with gradient propagation through the adaptive branches.

Connection to Module 10.2. Feedback-aware variational control is a specific instance of measurement-adaptive VQAs. The branches correspond to different control decisions based on the measurement outcome.

Example. Adaptive state preparation with error correction. The variational ansatz includes syndrome measurements and conditional correction gates. The optimization finds both the preparation unitaries and the correction logic.

Research status. Early-stage (2024-2026). Most work so far has been proof-of-concept on small systems. The adaptive gradient machinery is still developing.

Hybrid Control-VQA Architectures

At the system level, VQAs can be embedded inside larger control systems. Several patterns:

1. VQA as an inner-loop pulse designer

The outer loop is a classical control system (PID or similar); the inner loop uses a VQA to design specific pulses on demand.

Example. A quantum simulator running for a long time needs periodic recalibration. Each recalibration uses a VQA to find optimal pulse parameters for the current hardware state.

2. VQA as a state estimator

A VQA computes the expectation values of key observables that feed into a classical estimator (e.g., Kalman filter).

Example. Quantum metrology with feedback: the VQA estimates the current field strength, the classical controller adjusts the probe state.

3. VQA as a control policy

The VQA's output parameters are used as control inputs to an external quantum system.

Example. Quantum thermodynamics experiments where the VQA designs a protocol for extracting work from a controlled system.

4. VQA as a learner

The VQA is trained offline to reproduce optimal control policies, then deployed as a fast online controller.

Example. Pre-train a VQA to output pulse schedules for a library of target gates; deploy as a gate library on hardware.

The Relationship To GRAPE And Krotov

Variational gate synthesis is *mathematically equivalent* to GRAPE (Gradient Ascent Pulse Engineering), the standard quantum optimal control algorithm. Both:

- Discretize time into slots.
- Parameterize the control as slot amplitudes.
- Optimize a fidelity cost via gradient descent.

The only differences are:

- **Terminology:** GRAPE calls them "pulse amplitudes"; VQA calls them "parameters".
- **Optimizer choice:** GRAPE uses L-BFGS traditionally; VQA uses whatever (Adam, natural gradient, HOPSO).
- **Cost evaluation:** GRAPE uses exact simulation; VQA often uses stochastic shot-based evaluation.
- **Noise handling:** GRAPE is typically noiseless; VQA handles noisy cost evaluations.

Modern practice is converging: optimal control researchers use VQA-style optimizers (Adam, natural gradient); VQA researchers use GRAPE-style cost formulations (multi-slot pulse grids). The labels are different but the methods are the same.

Reference. Khaneja et al. (2005) introduced GRAPE; it can now be understood as a VQA for gate synthesis.

Hardware-Level Quantum Control Via VQAs

An emerging research direction: using VQAs to design pulse sequences *on the hardware itself*, not on a classical simulator.

Architecture.

1. The target gate is specified (e.g., Toffoli).
2. The VQA ansatz is a parameterized pulse schedule with pulse amplitudes as parameters.
3. The cost is gate fidelity, measured via randomized benchmarking or state tomography.
4. The optimizer updates the pulse parameters based on measured fidelity.
5. Repeat until convergence.

Advantage. The VQA *learns* the right pulse for the specific hardware, accounting for calibration errors, drift, and crosstalk that a classical simulator cannot model perfectly.

Cost. Each iteration requires hardware runs (measuring fidelity). Training can take hours to days for complex gates.

Reference. Heeres et al. (2017) demonstrated VQA-style learning on superconducting cavity QED; Niu et al. (2019) extended to transmon gates with better sample efficiency.

Thesis View

The thesis's framing: **quantum control is a cousin of VQA, not a separate discipline**. Both are variational problems over quantum circuit or pulse parameters. The differences are terminological.

This suggests a unified architectural approach:

1. **The objective generator** can be a gate fidelity (for control) or a Hamiltonian expectation (for VQA). The optimizer doesn't care.

2. **The dynamical update** (gradient descent, HOPSO) is the same across both.

3. **Noise handling** is the same — both face shot noise, gate errors, drift.

4. **Adaptive architectures** from Module 10.1 apply equally to both. A closed-loop VQA for chemistry is structurally the same as a closed-loop quantum control system for gate synthesis.

The thesis argues that **HOPSO is well-suited to quantum control problems** because its hybrid gradient+population structure handles the rougher landscapes of high-dimensional pulse design (where local minima are common).

Structural Role

This node establishes the **cross-application** of VQA techniques to quantum control problems. It shows that VQAs for gate synthesis, state preparation, simulation, feedback-aware control, and hybrid architectures are all instances of the same variational optimization framework.

It is the node that expands Module 10's scope beyond "chemistry + combinatorial" VQAs into the broader quantum engineering domain.

Relations to Other Concepts

- **Khaneja, Brockett, Glaser et al. (2005)** — GRAPE.
- **Krotov (1996)** — Krotov method for optimal control.
- **D'Alessandro (2007)** — *Introduction to Quantum Control*.
- **Heeres et al. (2017)** — variational pulse design on cavity QED.
- **Niu et al. (2019)** — RL-based pulse optimization.
- **Module 7.4 (Optimal Control)** — the foundations.
- **Module 10.1 (Adaptive VQA architectures)** — the deployment framework.

Worked Example — Variational Design Of A 3-Qubit Toffoli

Setup. Design an optimal pulse sequence implementing Toffoli on 3 superconducting qubits with: - Single-qubit gates (X, Z, H) with 99.95% fidelity. - Two-qubit CNOT with 99.5% fidelity on adjacent qubits. - Coherence: T2 $\approx$ 100 μs.

Ansatz. A 20-slot time grid, each slot containing a pulse on a specified qubit (single or two-qubit), with amplitude as the parameter. Total parameters: 20.

Cost. $C = 1 - |\mathrm{Tr}(\mathrm{Toffoli}^{\dagger} U(\boldsymbol{\theta}))|^2/64$.

Optimizer. Adam with natural gradient preconditioning.

Training. - Start from a random initialization. - Each iteration computes the cost via statevector simulation. - Gradient via parameter shift on each pulse amplitude. - Learning rate 0.01 with decay.

Results (simulated). - Initial fidelity: ~5%. - After 50 iterations: 90%. - After 200 iterations: 99.9%. - Final pulse sequence: 12 non-trivial slots (rest are near-zero amplitudes).

Interpretation. The VQA "discovers" a Toffoli decomposition that is shorter than the standard 6-CNOT compilation. The specific pulse sequence depends on the random initialization, but the final gate count is consistently lower than the textbook decomposition.

Hardware deployment. Taking the simulated pulse design to hardware, the measured fidelity is 98.5% (degraded from simulation's 99.9% due to hardware noise). With in-hardware training (feedback from randomized benchmarking), the fidelity improves back to 99.3%.

Lesson. VQA-style optimization is a legitimate tool for gate synthesis, especially when hardware-specific tuning matters.

Visualization

Pending. A plot of cost (gate infidelity) vs iteration for VQA-based Toffoli design, showing the decrease from 95% to 0.1%. Caption: "Variational gate synthesis: VQA as a gate compiler."

Exercises

1. For a 1-qubit arbitrary unitary $U \in SU(2)$, show that variational gate synthesis with a 3-parameter ansatz $U(\theta_1, \theta_2, \theta_3) = R_z(\theta_3)R_y(\theta_2)R_z(\theta_1)$ can exactly reach U. What is the cost function's landscape?
2. Compare the variational gate synthesis approach to explicit Solovay-Kitaev decomposition. Which is more accurate? Which is more efficient?
3. (VQA) For a 4-qubit target state (e.g., a 4-qubit W state), implement a variational state preparation VQA with a HEA ansatz. How many layers are needed for 99% fidelity?

Research Extensions

- **Closed-loop variational gate synthesis on hardware** — training pulse parameters directly on the hardware with real-time feedback.
- **Simultaneous multi-target optimization** — designing a family of gates with shared ansatz structure.
- **Robust variational control** — gates that are robust to drift and noise, not just optimal at one time.
- **Variational time-optimal control** — minimizing gate time subject to fidelity constraints.
- **Learning-based variational control** — replacing gradient descent with neural network policies.

Cross-links to Other Courses

- **Quantum Optimal Control (D'Alessandro)** — the parent discipline.
- **GRAPE and Krotov** — the classical optimal control methods.
- **Module 7.4 (Optimal Control)** — the theoretical foundation.
- **Module 10.1 (Adaptive architectures)** — the deployment framework.
- **Module 10.8 (Measurement-optimizer co-design)** — the extreme case.

Variational Quantum Algorithms · Module 10 — Vqa As Adaptive Control Systems · Node 10.4
Concepts: control-theory, unitary-evolution, cost-function, parameterized-circuits
Prerequisites: 7.4, 10.1
Enables: 10.5

hbar.university — own.your.cognition