Module 7 — Control Theoretic View

Variational Quantum Algorithms

hbar.university

VQA as a Control System

Position in Knowledge Graph

System: Variational Quantum Algorithms **Structural Domain:** Control-Theoretic View **Node:** 7.1

This is the **opening node** of Module 7, the module that re-frames everything Modules 1-6 have built up — ansätze, costs, optimizers — in the language of **classical control theory**.

The motivation is more than aesthetic. Control theory has 70+ years of accumulated theory about steering dynamical systems with inputs, observing them with sensors, and designing feedback laws — *exactly* the same shape of problem a VQA poses. Importing the framework gives:

1. **A vocabulary** for talking about VQAs that connects to a mature engineering literature.
2. **Theorems** (controllability, observability, separation principle) that apply with minor modification.
3. **Design tools** (LQR, MPC, Pontryagin minimum principle) that can be applied to ansatz design and optimizer scheduling.
4. **A new perspective** on the open problems (barren plateaus, noise, regularization) that often illuminates them differently from the optimization-theoretic view.

The point of this module is *not* to replace the optimization view. It's to provide a second axis of understanding — one that is especially powerful for the questions Modules 9 (Hardware Noise) and 10 (VQA as Adaptive Control Systems) take up.

What A Control System Is

A **classical control system** is a dynamical system with:

- A **state** $\mathbf{x}(t) \in \mathcal{X}$ that evolves over time.
- A **control input** $\mathbf{u}(t) \in \mathcal{U}$ that the controller can choose.
- A **dynamics** $\dot{\mathbf{x}} = f(\mathbf{x}, \mathbf{u}, t)$ that says how the state changes given the input.
- An **output** $\mathbf{y}(t) = h(\mathbf{x}, t) \in \mathcal{Y}$ that the controller can observe.
- An **objective** $J = \int L(\mathbf{x}, \mathbf{u})\, dt$ (or terminal cost) that the controller is trying to minimize.

The **control problem** is: choose $\mathbf{u}(t)$ so that J is minimized, possibly subject to constraints on $\mathbf{x}$, $\mathbf{u}$, and the observable measurements $\mathbf{y}$.

Different specializations give:

- **Open-loop control:** $\mathbf{u}(t)$ is chosen as a function of time alone, without observing the actual $\mathbf{x}(t)$.
- **Closed-loop / feedback control:** $\mathbf{u}(t) = K(\mathbf{y}(t), t)$ — the input depends on the measured output.
- **Optimal control:** the input is chosen to extremize a functional, typically using the Pontryagin minimum principle (Section 7.5).
- **Adaptive control:** the controller's parameters adapt over time in response to system behavior (Module 10).

A VQA Is A Control System

The mapping is direct:

Control concept	VQA analogue
State $\mathbf{x}$	Quantum state $\lvert\psi(\boldsymbol{\theta})\rangle$ (or its density matrix)
Control input $\mathbf{u}$	Circuit parameters $\boldsymbol{\theta}$ (or update steps)
Dynamics $\dot{\mathbf{x}} = f(\mathbf{x}, \mathbf{u})$	Schrödinger evolution under parameterized gates
Output $\mathbf{y}$	Measurement outcomes (estimated expectation values)
Objective J	Cost function $C(\boldsymbol{\theta}) = \langle\psi(\boldsymbol{\theta})\lvert H\rvert\psi(\boldsymbol{\theta})\rangle$
Open-loop	Pre-compiled, fixed-circuit VQA
Feedback	Parameter updates based on measurements
Optimal control	Choosing parameter trajectories to minimize cost
Adaptive control	Optimizers that change strategy based on observed behavior

This isn't an analogy — it's a translation. A VQA *is* a control system, with quantum dynamics replacing classical ODEs.

The question is what changes when you make this substitution. Most of control theory was developed for classical, deterministic, low-dimensional dynamical systems. Quantum systems are stochastic (measurement is intrinsically random), high-dimensional (2^n states), and have fundamentally different observability constraints (you collapse the state when you measure). So the import is not literal — it requires care.

Two Time Scales: The Inner And Outer Loops

A VQA has *two* dynamical loops, and the control framing makes them explicit.

Inner loop (state preparation). At fixed parameters $\boldsymbol{\theta}$, the quantum circuit evolves the state from $\lvert 0\rangle$ to $\lvert\psi(\boldsymbol{\theta})\rangle$. The "time" variable here is the position in the circuit. Each gate is a control input that advances the state.

Outer loop (parameter optimization). The parameters $\boldsymbol{\theta}^k$ are updated iteration-by-iteration based on measurements of the cost. The "time" variable is the optimizer iteration. Each update is a control input that adjusts the *next* circuit's behavior.

These two loops have very different character:

- The **inner loop** is *unitary*, deterministic given $\boldsymbol{\theta}$, and runs in microseconds. Quantum control theory (which exists as a field) addresses it.
- The **outer loop** is *classical*, stochastic (because measurements are sampled), and runs in seconds-to-hours. This is where the optimization theory of Module 6 lives.

The control-theoretic view treats them as a *cascaded* control system: the outer loop controls the parameters, which in turn control the inner-loop dynamics, which in turn produce measurements, which feed back to the outer loop. Each level has its own controllability, observability, and stability properties.

What The Control Framing Buys You

A few specific things the control view gives that the pure-optimization view misses:

1. Controllability questions. Given a parameterized circuit, *which states can I steer the system to?* This is exactly the "reachability" question of expressivity (Module 8), but the control version connects it to a 60-year theorem stack about Lie algebras of vector fields, accessibility conditions, and the Chow-Rashevsky theorem.

2. Observability questions. Given measurement access to certain observables, *what can I learn about the underlying state?* This is the question of whether your cost function can resolve the parameter values you care about. Classical observability theory has explicit conditions (the observability matrix rank) that translate cleanly.

3. The separation principle. Classical control theory says: under linearity and Gaussian noise, you can design the state estimator separately from the feedback law and the result is optimal. A quantum analogue exists in some regimes and gives a clean architecture for adaptive VQAs.

4. Robust control. What if the dynamics aren't exactly known? Classical robust control (H_∞, μ-synthesis) gives explicit guarantees on stability margins. The VQA analogue addresses noise and miscalibration, and gives quantitative robustness statements.

5. Model predictive control. Looking ahead a few steps and optimizing over the lookahead window. Naturally fits VQA optimizer design with lookahead — the optimizer "thinks" several iterations ahead.

The control framing is *especially* useful when noise, drift, or hardware time-variation are present — exactly the regimes where the static optimization view starts to break down.

Quantum Control Vs Classical Control

Quantum control has been a field in its own right since the 1980s. It addresses the inner-loop problem: given a quantum system with controllable Hamiltonian $H(\mathbf{u}(t))$, choose $\mathbf{u}(t)$ to steer the state from one configuration to another.

Major results:

- **Controllability theorems** (Schirmer-Solomon, D'Alessandro): a quantum system is fully controllable iff the Lie algebra generated by its drift and control Hamiltonians spans $\mathfrak{su}(2^n)$. Same Lie algebra story as Section 8.1 — the dynamical Lie algebra.
- **Optimal control algorithms** (GRAPE, Krotov, CRAB): iterative methods that find time-dependent control pulses to perform target unitaries. Used for high-fidelity gate calibration.
- **Decoherence-free subspaces and dynamical decoupling**: control protocols that protect quantum states from noise.

These results are *directly relevant* to VQAs. A parameterized circuit can be viewed as a discretization of an underlying continuous control problem. Many of the GRAPE-style insights transfer.

The thesis's view: **VQA optimization is quantum control with discrete inputs, and it inherits both the strengths and the limitations of the broader quantum control field.**

Where The Mapping Breaks Down

The control-theory analogy is powerful but imperfect. A few places it strains:

1. Discrete vs continuous time. Classical control is typically continuous time ($\dot{\mathbf{x}} = f(\mathbf{x}, \mathbf{u})$) or discrete-time with regular step sizes. A VQA is event-driven: each gate is a discrete event with no fixed time step. Translating between continuous and discrete views introduces some friction.

2. Measurement is irreversible. Classical sensors don't disturb the state; quantum measurements collapse it. So "observe and feed back" is not free — it changes the system. Continuous quantum measurement theory (Wiseman-Milburn) addresses this but is more complex than classical observation theory.

3. The cost is itself stochastic. In classical control, you typically know J exactly. In a VQA, you only have a *noisy estimate* of the cost from finitely many shots. This is a stochastic-control problem, not a deterministic one.

4. The state space is exponentially large. $|\psi\rangle \in \mathbb{C}^{2^n}$. Classical control theory works for low-dimensional state spaces. The high dimension is what makes barren plateaus possible (and what makes quantum control hard in general).

5. Two-loop structure. Classical control is usually a single loop. The VQA two-loop structure (inner unitary loop, outer classical loop) requires extending the framework.

These caveats matter, and Module 7 will address them as we go. None is fatal, but each requires the standard control concepts to be modified before they apply cleanly.

The Module 7 Plan

Module 7 will develop the control-theoretic framework in stages:

- **Node 7.2 (Observability):** what can we learn from measurements? When does the cost function have enough resolution to identify the optimal parameters?
- **Node 7.3 (Controllability):** which states can we reach? The Lie-algebraic conditions for full and partial controllability.
- **Node 7.4 (Optimal Control Theory):** the variational framework for choosing trajectories. Connects to quantum optimal control (GRAPE).
- **Node 7.5 (Pontryagin Principle):** the necessary conditions for optimality. Gives a different angle on why VQA optimization is hard.
- **Node 7.6 (Feedback Control):** closing the loop. Adaptive optimizer design and the separation principle.

Module 7 is the bridge between Module 6 (optimization dynamics) and Module 10 (adaptive control systems). The theoretical tools live here; the deployment patterns live in 10.

Structural Role

This node is the **opening framing** of Module 7. It establishes the control-theoretic vocabulary, maps VQA concepts onto control concepts, identifies the two-loop structure, and previews the rest of the module.

It is also the bridge from the optimization-theoretic view (Modules 5, 6) to the adaptive deployment view (Modules 9, 10), with control theory as the connective tissue.

Relations to Other Concepts

- **Wiener (1948)** — cybernetics, the founding text of control as an interdisciplinary field.
- **Pontryagin et al. (1962)** — the minimum principle, the variational foundation of optimal control.
- **Schirmer-Solomon (2001)** — quantum controllability theorems.
- **D'Alessandro (2007)** — *Introduction to Quantum Control and Dynamics*, the textbook.
- **Module 6 (Optimization Dynamics)** — the optimization view, complementary to this one.

Worked Example — Mapping A QAOA Onto A Control System

Consider QAOA at depth $p = 2$ on a 4-qubit MaxCut problem. The circuit is

$$|\psi(\boldsymbol{\beta}, \boldsymbol{\gamma})\rangle = e^{-i\beta_2 H_M} e^{-i\gamma_2 H_C} e^{-i\beta_1 H_M} e^{-i\gamma_1 H_C} |+\rangle^{\otimes 4},$$

where H_C is the cost Hamiltonian and H_M is the mixer.

Map onto control system.

- **State:** $|\psi(t)\rangle \in \mathcal{H} = \mathbb{C}^{16}$, evolving as the gates are applied.
- **Control input:** $\mathbf{u}(t)$ is piecewise constant, equal to β_1 on the first mixer interval, γ_1 on the first cost interval, etc. There are 4 distinct control intervals.
- **Dynamics:** Schrödinger equation $i|\dot{\psi}\rangle = H(\mathbf{u}(t))|\psi\rangle$, where $H(\mathbf{u})$ is either H_C or H_M depending on the interval.
- **Output:** measurement of H_C at the end, giving $y = \langle\psi(T)|H_C|\psi(T)\rangle$.
- **Objective:** minimize y (or equivalently maximize $-y$).

This is a **bang-bang control** problem — the control switches abruptly between two values H_C and H_M. The controller chooses how long to spend in each mode (the β_k, γ_k values) to minimize the terminal cost.

In control-theoretic language, QAOA is **discrete-time bang-bang optimal control with terminal cost**. The Pontryagin minimum principle (Section 7.5) gives explicit necessary conditions for the optimal β_k, γ_k, and these conditions match the gradient conditions from parameter-shift differentiation.

Why this matters: the mapping shows that QAOA is *not* an exotic quantum heuristic — it is a classical control problem in quantum disguise. The same theorems that govern bang-bang optimal control of classical systems govern QAOA, with appropriate modifications for the quantum state space.

Visualization

Pending. A block diagram of the VQA two-loop control system: inner loop (parameters → quantum circuit → state preparation → measurement → cost estimate), outer loop (cost estimate → optimizer → parameter update → back to inner loop). Caption: "VQA is a cascaded control system with quantum dynamics inside and classical optimization outside."

Exercises

1. For a simple 1-qubit ansatz $R_Y(\theta_2)R_X(\theta_1)|0\rangle$, identify the state, control input, dynamics, output, and objective in control-theoretic language.
2. The two-loop structure has different time scales (inner: microseconds, outer: seconds). Identify *which* control problems are most affected by this disparity. (Hint: think about adaptive control with measurement feedback.)
3. (VQA) For a quantum hardware system with a known calibration drift on the order of hours, sketch how a control-theoretic adaptive scheme would respond. Which loop (inner or outer) handles the drift?

Research Extensions

- **Quantum-classical hybrid control theory** — formalizing the two-loop structure with explicit theorems.
- **Real-time adaptive VQA controllers** — implementing feedback loops at the inner-loop time scale.

- **Robust VQA design via robust control techniques** — applying H_∞ and μ-synthesis to ansatz selection.
- **Receding-horizon (MPC) VQA optimization** — model-predictive control variants of VQA optimizers.

Cross-links to Other Courses

- **Classical Control Theory** — Wiener, Kalman, Pontryagin, the foundational literature.
- **Quantum Control (D'Alessandro)** — the existing field at the inner-loop level.
- **Dynamical Systems** — the broader framework of state-space dynamics.
- **Module 6 (Optimization Dynamics)** — the optimizer-theoretic view, complementary.
- **Module 10 (Adaptive Control Systems)** — the deployment endpoint.

Observability of Quantum States

Position in Knowledge Graph

System: Variational Quantum Algorithms **Structural Domain:** Control-Theoretic View **Node:** 7.2

In control theory, **observability** is the question: given the outputs you can measure, can you reconstruct the underlying state? It is the dual of **controllability** (which input combinations can drive the system to which states) — observability asks what *information* you have about a state, not what states you can reach.

For VQAs, observability translates into a specific operational question: **does the cost function uniquely identify the optimal parameters, or are there directions in parameter space along which the cost is flat (or symmetric)?** If the cost is observability-deficient, then no amount of optimization will distinguish between equivalent parameter values — there is a *gauge symmetry* and the optimizer wanders aimlessly along the gauge orbit.

This matters for VQA design in several distinct ways: it explains certain "stuck" optimizer behaviors, motivates the choice of cost function, and connects to the broader question of *why some VQAs are easier to train than others.*

This node introduces classical observability, the quantum-state observability problem, the cost-function observability question for VQAs, and how to diagnose observability deficiencies in practice.

Classical Observability

In classical control, a linear system is

$$\dot{\mathbf{x}} = A\mathbf{x} + B\mathbf{u}, \quad \mathbf{y} = C\mathbf{x},$$

where $\mathbf{x} \in \mathbb{R}^n$ is the state, $\mathbf{u} \in \mathbb{R}^m$ is the input, and $\mathbf{y} \in \mathbb{R}^p$ is the measured output.

Observability theorem (Kalman): the system is observable if and only if the **observability matrix**

$$\mathcal{O} = \begin{pmatrix} C \\ CA \\ CA^2 \\ \vdots \\ CA^{n-1} \end{pmatrix}$$

has rank n. Equivalently, no two distinct initial states $\mathbf{x}(0)$ and $\mathbf{x}'(0)$ produce identical output trajectories.

If the observability matrix has rank $r < n$, then there is an $(n-r)$-dimensional **unobservable subspace**: states that differ by a vector in this subspace are indistinguishable from output measurements.

This is the canonical setup. The **rank of the observability matrix** is the central quantitative measure.

Observability For Quantum States

The quantum analogue is more delicate. Quantum measurement has two distinct features:

1. Measurement collapses the state. Once you measure, the state changes. So the "output trajectory" of a single state is impossible — you can only measure one shot, then the state is gone.

2. Single-shot measurements are random. Even given a fixed observable, the result of a single measurement is sampled from a distribution.

So "observability" in the quantum sense means: **given the ability to prepare many copies of the same state and measure any of a fixed set of observables, can you reconstruct the state?**

The answer is: **yes, with sufficiently informative observables**. Specifically, a set of observables $\{O_1, O_2, \ldots, O_K\}$ is **informationally complete** if knowing all expectation values $\{\langle\psi|O_k|\psi\rangle\}_k$ uniquely identifies $|\psi\rangle$ (up to global phase).

Standard informationally complete sets:

- **Pauli decomposition:** all 4^n products of single-qubit Paulis $\{I, X, Y, Z\}^{\otimes n}$. Together they span all Hermitian operators on n qubits.
- **Symmetric informationally complete (SIC) POVMs:** d^2 outcomes that uniquely identify any d-dimensional state.
- **Mutually unbiased bases (MUBs):** $d+1$ bases, each measurement gives d outcomes, together informationally complete.

Cost of informational completeness: 4^n Pauli observables (or $d^2 = 4^n$ SIC outcomes) — exponentially many. **Full quantum state reconstruction is exponentially expensive.**

In practice, you don't need full reconstruction. You only need enough observables to identify the *parameters of the ansatz*, which is a much smaller set (at most P numbers).

Cost-Function Observability For VQAs

The VQA-specific observability question:

Definition (cost observability). A cost function $C(\boldsymbol{\theta})$ is **observable** at $\boldsymbol{\theta}^*$ if no other parameter value gives the same cost in a neighborhood: $C(\boldsymbol{\theta}^* + \delta) \neq C(\boldsymbol{\theta}^*)$ for all $\delta \neq 0$ in some neighborhood.

More quantitatively: the cost is **first-order observable** if the gradient $\nabla C(\boldsymbol{\theta}^*)$ has full rank in directions that matter (i.e., is not zero along any direction you care about). It is **second-order observable** if the Hessian has full rank.

Observability deficiencies in VQAs:

1. **Gauge symmetries.** If the ansatz has parameter combinations that produce the same state (e.g., $R_Z(\theta_1)R_Z(\theta_2)$ only depends on $\theta_1 + \theta_2$), then the cost depends only on $\theta_1 + \theta_2$, and the perpendicular direction is unobservable. This is the *gauge orbit* problem.

2. **Symmetries of the cost.** If the cost is invariant under some parameter transformation (e.g., $\boldsymbol{\theta} \to -\boldsymbol{\theta}$ for some symmetric ansätze), then there are multiple equivalent minima and the cost can't distinguish them.

3. **Local degeneracies.** Even without explicit symmetries, the cost can have flat directions at specific points (e.g., saddles, ridges) where the gradient vanishes along some direction.

4. **Barren plateaus as an extreme case.** A barren plateau is a region where the gradient is *near zero in all directions* — the cost is unobservable in *every* direction simultaneously. This is the worst-case observability deficiency.

The Observability Matrix For VQAs

The classical observability matrix can be adapted to VQAs, with parameters playing the role of state.

For a cost function $C(\boldsymbol{\theta})$ viewed as a function on parameter space, the local observability is captured by the **Hessian matrix** $H_{ij}(\boldsymbol{\theta}) = \partial^2 C / \partial\theta_i \partial\theta_j$. The rank of H at a critical point measures the local observability:

- **Full rank P:** the critical point is non-degenerate. Locally, the cost uniquely identifies the parameters in all directions.
- **Rank $r < P$:** the critical point has $P - r$ flat directions. Locally, the cost is unobservable in those directions.
- **Rank 0:** the cost is locally flat — barren plateau.

The **observability gap** is the difference between the largest and smallest non-zero Hessian eigenvalues. A small gap means the optimizer struggles to distinguish important from unimportant directions, and the convergence rate degrades accordingly.

Observability And The Fubini-Study Metric

A more geometric view connects observability to the **Fubini-Study metric** on the projective Hilbert space.

For a parameterized state $|\psi(\boldsymbol{\theta})\rangle$, the FS metric on parameter space has matrix elements

$$g_{ij}(\boldsymbol{\theta}) = \mathrm{Re}\left[\langle \partial_i \psi | \partial_j \psi \rangle - \langle \partial_i \psi | \psi \rangle \langle \psi | \partial_j \psi \rangle\right].$$

This is the **Quantum Fisher Information Matrix** (QFIM, Module 6.2) and it captures how much the state changes per unit parameter change.

Connection to observability:

- If $g(\boldsymbol{\theta})$ is full rank, every parameter direction produces a distinguishable state — full state-level observability.
- If $g(\boldsymbol{\theta})$ is rank-deficient, there are parameter directions that don't change the state — *gauge directions*. These are unobservable in any cost function.
- The gauge dimension is $P - \mathrm{rank}(g)$.

This is a *state-level* observability that is independent of the cost. The cost-level observability (Hessian rank of C) is generally tighter (a flat direction in g is automatically flat in C, but a flat direction in C need not be flat in g).

The QFIM gives the **theoretical upper bound** on cost observability for any cost function built from $|\psi(\boldsymbol{\theta})\rangle$.

Practical Diagnostics

How do you *check* observability for a real VQA? A few standard diagnostics:

1. Hessian rank at a candidate optimum. Compute the Hessian (via second-order parameter shift), check the eigenvalue distribution. Rank $< P$ signals an observability deficiency.

2. QFIM rank. Compute the QFIM (Section 6.2), check rank. The gap to P gives the gauge-direction count.

3. Trajectory variance. Run the optimizer from multiple random starting points. If the final parameters cluster, the cost is observable. If they spread out (different parameters reaching the same cost), there are equivalent solutions.

4. Cost contour visualization. For low-dimensional cases, plot the cost contours. Flat regions = observability deficiency.

5. Gradient norm distribution. During training, track the gradient norm. If it becomes very small without the cost decreasing, you're in a flat region.

These diagnostics are simple in principle but require care in interpretation — many VQAs have *some* observability deficiency without it being catastrophic.

Fixing Observability Deficiencies

Once diagnosed, what can you do?

1. Remove gauge redundancies. If the ansatz has explicit gauge freedoms, parameterize the gauge slice instead. E.g., fix one of the redundant parameters to zero.

2. Choose a more informative cost. Some costs are observability-deficient by design (e.g., a single global expectation value). Adding multiple observables (Section 7.6 sketches the multi-observable case) can break the degeneracy.

3. Add a regularization term. A regularization term that depends on the parameters directly (rather than on the cost) can break flat directions and make the optimization landscape better-conditioned. This is one of the thesis's main themes.

4. Restart from a different initialization. Sometimes the observability deficiency is local — restarting from a different point lands you in a region with better observability.

5. Use an observability-aware optimizer. Natural gradient (Module 6.2) explicitly accounts for the QFIM and avoids moving in unobservable directions. This is a partial fix.

The Separation Of State And Parameter Observability

A subtle but important distinction:

State observability: can the cost function identify the state $|\psi(\boldsymbol{\theta})\rangle$ — i.e., does the cost uniquely depend on the state?

Parameter observability: can the cost function identify the parameters $\boldsymbol{\theta}$ — i.e., does the cost uniquely depend on the parameters?

These are *not* the same. A cost can be state-observable (different states give different costs) but parameter-deficient (different parameters give the same state, hence the same cost). This is exactly the gauge-orbit case.

Conversely, a cost can be parameter-observable but state-deficient (different parameters give different costs even though they produce different but cost-equivalent states). This is rare but possible with badly designed costs.

For practical VQA design, **state observability is what you want at the application level** (the optimizer should find the right state), but **parameter observability is what the optimizer sees** (it can only adjust parameters, not states directly). The mismatch between them is a recurring source of subtle bugs and slow convergence.

Structural Role

This node is the **measurement-side foundation** of Module 7. It introduces the observability concept, walks through the classical and quantum versions, applies it to the VQA cost-observability question, and provides diagnostics for observability deficiencies.

It directly enables node 7.4 (optimal control theory, which uses observability to design state estimators) and node 7.6 (feedback control, which requires observable outputs to close the loop).

Relations to Other Concepts

- **Kalman (1960)** — the original classical observability theorem.
- **D'Alessandro (2007)** — quantum observability via informationally complete observables.
- **Quantum tomography** — the operational version of state observability.
- **Module 6.2 (Natural Gradient)** — the QFIM, which captures state observability.
- **Module 8 (Expressivity)** — the dual question (controllability rather than observability).

Worked Example — Observability Of A 1-Qubit Ansatz

Consider $|\psi(\theta_1, \theta_2)\rangle = R_Y(\theta_1) R_Y(\theta_2)|0\rangle$ — two R_Y rotations on 1 qubit.

State. $R_Y(\theta_1)R_Y(\theta_2) = R_Y(\theta_1 + \theta_2)$. So the state depends only on $\theta_1 + \theta_2$.

Cost. Suppose $C = \langle\psi|Z|\psi\rangle = \cos(\theta_1 + \theta_2)$. The cost depends only on the sum.

Observability analysis. - State observability: the cost uniquely identifies the state (since cost = cos of the rotation angle, which is monotone in a half-period). **- Parameter observability:** the cost is *constant* along the line $\theta_1 + \theta_2 = \text{const}$. This is a 1-dimensional unobservable direction in 2-dimensional parameter space. **- Gauge dimension:** 1 (the $\theta_1 - \theta_2$ direction is unobservable).

QFIM. Let $\phi = \theta_1 + \theta_2$. Then $|\psi\rangle = R_Y(\phi)|0\rangle$ and $|\partial_1\psi\rangle = |\partial_2\psi\rangle = (\partial/\partial\phi)|\psi\rangle$. The QFIM is

$$g = \begin{pmatrix} 1/4 & 1/4 \\ 1/4 & 1/4 \end{pmatrix},$$

which has rank 1, eigenvalues $\{1/2, 0\}$. The zero eigenvalue confirms the unobservable direction.

Hessian. $H_{ij} = \partial^2 C/\partial\theta_i\partial\theta_j = -\cos(\theta_1 + \theta_2)$ for all i, j. Same matrix structure (rank 1).

Practical implication. A gradient-descent optimizer started at random will move in the gradient direction (which is along $(1,1)^T$), reaching a cost-decreasing trajectory but never resolving the perpendicular direction. The final (θ_1, θ_2) will not be unique — it will lie on the line $\theta_1 + \theta_2 = \theta^*$ for some θ^*.

Fix. Eliminate θ_2 (set to zero) and use only θ_1. Now you have a 1-parameter ansatz with full observability. Same expressive power, half the parameters, no gauge problem.

This worked example is small but the structural lesson generalizes: **redundant parameters create observability holes**, and pruning them makes the optimization easier.

Visualization

Pending. A 2D heatmap of $C(\theta_1, \theta_2) = \cos(\theta_1 + \theta_2)$ showing the diagonal stripes of constant cost. Overlaid: a gradient-descent trajectory that follows the gradient direction but ends at an arbitrary point on a stripe. Caption: "The flat direction is the unobservable subspace. The optimizer doesn't know where on the stripe to stop."

Exercises

1. For a 2-qubit ansatz $R_Y(\theta_1) \otimes R_Y(\theta_2)|00\rangle$ with cost $C = \langle Z_0 + Z_1\rangle$, is the cost observable in both parameters?
2. The QFIM has rank r at a parameter point. How many gauge directions does the parameter space have at that point?

3. (VQA) For a Hamiltonian Variational Ansatz on 6 qubits with 12 parameters, suppose you compute the Hessian rank at a candidate optimum and find it is 9. What does this imply about the optimizer behavior?

Research Extensions

- **Dynamical observability for time-varying systems** — extending classical observability theorems to drift-prone hardware.
- **Approximate observability under shot noise** — how many shots are needed to identify a parameter to a given accuracy?
- **Cost-design for observability** — algorithmic methods to choose costs that maximize parameter observability.
- **Observability and adaptive optimization** — using observability diagnostics to drive optimizer choice.

Cross-links to Other Courses

- **Linear Systems Theory** — Kalman observability, the classical foundation.
- **Quantum State Tomography** — the operational version of state-level observability.
- **Module 6.2 (Natural Gradient)** — the QFIM-based view of observability.
- **Module 7.3 (Controllability)** — the dual notion.
- **Module 7.6 (Feedback Control)** — where observability becomes operationally critical.

Controllability of Parameterized Circuits

Position in Knowledge Graph

System: Variational Quantum Algorithms **Structural Domain:** Control-Theoretic View **Node:** 7.3

This is the **dual** of node 7.2. Where observability asks "which states can we *learn about*?", controllability asks "which states can we *steer to*?"

For a VQA, controllability is the question of which states are *reachable* by varying the circuit parameters. This is the same question Module 8 asked under the name "expressivity," but the control-theoretic view gives it a different mathematical form (Lie algebras of vector fields, accessibility conditions, Chow-Rashevsky theorem) and a more *dynamical* flavor.

The two perspectives — expressivity (Module 8) and controllability (this node) — are equivalent in their conclusions but they are formulated differently and they each give different intuitions and proof techniques. This node walks through the controllability formulation, makes the link to expressivity explicit, and applies the controllability theorems to the practical question of ansatz design.

Classical Controllability

For a linear classical system

$$\dot{\mathbf{x}} = A\mathbf{x} + B\mathbf{u},$$

the **controllability matrix** is

$$\mathcal{C} = \begin{pmatrix} B & AB & A^2B & \cdots & A^{n-1}B \end{pmatrix}.$$

Kalman controllability theorem: the system is controllable (i.e., from any initial state $\mathbf{x}_0$ you can reach any target state $\mathbf{x}_f$ in finite time using suitable $\mathbf{u}(t)$) if and only if $\mathcal{C}$ has full rank n.

If the rank is $r < n$, then there is an $(n-r)$-dimensional **uncontrollable subspace**: states that differ by a vector in this subspace cannot be reached from each other.

This is the linear version. For nonlinear systems, the controllability question is more delicate and uses the language of Lie algebras of vector fields.

Nonlinear Controllability And The Lie Bracket

For a nonlinear control-affine system

$$\dot{\mathbf{x}} = f_0(\mathbf{x}) + \sum_{j=1}^{m} u_j f_j(\mathbf{x}),$$

where f_0 is the **drift vector field** (autonomous dynamics with no control) and $f_1, \ldots, f_m$ are the **control vector fields**, the relevant object is the **Lie algebra generated by the vector fields**.

Lie bracket of two vector fields:

$$[f_i, f_j](\mathbf{x}) = \frac{\partial f_j}{\partial \mathbf{x}}(\mathbf{x}) f_i(\mathbf{x}) - \frac{\partial f_i}{\partial \mathbf{x}}(\mathbf{x}) f_j(\mathbf{x}).$$

The Lie bracket measures the "non-commutation" of two vector fields. Operationally, if you flow along f_i for a small time, then f_j for a small time, then $-f_i$, then $-f_j$, you don't return to the start — you arrive at a point displaced from the start by approximately $[f_i, f_j]$ times the square of the time step.

Chow-Rashevsky theorem: the system is controllable on a connected manifold if the Lie algebra generated by $\{f_0, f_1, \ldots, f_m\}$ under repeated commutation spans the tangent space at every point.

This is the generalization of Kalman's controllability theorem to nonlinear systems. The "rank" condition is replaced by a Lie-algebra spanning condition.

Quantum Controllability

The Schrödinger equation with control Hamiltonian $H(\mathbf{u}(t)) = H_0 + \sum_j u_j(t) H_j$ is

$$i|\dot{\psi}\rangle = \left(H_0 + \sum_j u_j(t) H_j \right) |\psi\rangle.$$

This is a control-affine system on the projective Hilbert space, with drift H_0 and control vector fields generated by H_j.

Theorem (Schirmer-Solomon, D'Alessandro): the system is **fully controllable on the unitary group** $U(2^n)$ (i.e., any unitary can be implemented with suitable $\mathbf{u}(t)$) if and only if the Lie algebra generated by $\{iH_0, iH_1, \ldots, iH_m\}$ under commutation is the full algebra $\mathfrak{u}(2^n)$.

This is the dynamical Lie algebra (DLA) introduced in node 8.1. The controllability theorem and the expressivity theorem are *the same theorem* in different language.

Variants:

- **State controllability (rather than unitary controllability):** the system is state-controllable if any state can be reached from any other. Weaker than full unitary controllability — requires the DLA to act *transitively* on the state space, not generate the full unitary group.
- **Operator controllability:** any unitary in some subgroup (e.g., particle-number-conserving) can be implemented. Requires the DLA to be exactly that subgroup's algebra.
- **Density-matrix controllability:** any density matrix (mixed states) can be reached. Requires *more* than full unitary controllability — you also need access to non-unitary operations (measurement, dissipation).

Controllability And Expressivity Are The Same Thing

The link is exact. For a parameterized circuit $U(\boldsymbol{\theta}) = \prod_j e^{-i\theta_j H_j}$:

- **Expressivity** = dimension of the reachable manifold = dimension of the orbit of $|0\rangle$ under the group generated by $\{e^{-i\theta H_j}\}$.
- **Controllability** = dimension of the orbit reachable under the control system with control Hamiltonians $\{H_j\}$.

These are the same dimension. The DLA $\mathfrak{g} = \mathrm{Lie}(\{iH_j\})$ governs both:

- $\dim \mathcal{R}(U) = \dim(\mathfrak{g} \cdot |0\rangle)$ (the orbit of $|0\rangle$).
- $\dim(\text{controllable set}) = \dim(\mathfrak{g} \cdot |0\rangle)$ (same).

So **the controllability theorem of quantum control theory and the expressivity theorem of VQA theory are the same theorem in different language**, both reducing to the dimension of the DLA.

This is more than a notational point — it means **70 years of quantum control theory directly applies to VQA expressivity questions**, including:

- Algorithmic methods for computing the DLA (graph-theoretic Lie closure).
- Conditions for *small* DLAs (when symmetries reduce the algebra).
- Approximate controllability theorems (when you can't reach the whole orbit but get arbitrarily close).
- Time-optimal control (the minimum time to reach a target — analogous to minimum circuit depth).

The thesis's view: **VQA practitioners should be reading the quantum control literature**, because that's where the relevant theorems live.

Reachable Set Vs Reachability In Finite Time

A subtle but important distinction:

Reachable set: the set of states you can *eventually* reach, possibly with arbitrarily long control trajectories.

Time-bounded reachable set: the set of states you can reach within a fixed time bound T.

For a fully controllable system, the reachable set is the whole projective Hilbert space. But the *time-bounded* reachable set is much smaller — it grows with T and only equals the full space in the limit $T \to \infty$.

For VQAs, "time" is replaced by **circuit depth**. The depth-bounded reachable set is the set of states reachable by a circuit of at most L layers. As L increases, this grows monotonically toward the full reachable set.

Quantitative version: for an HEA with random initialization, the depth-bounded reachable set covers an ϵ-fraction of the Hilbert space at depth $L \sim n \log(1/\epsilon)$. Below this depth, the reachable set is a sparse subset; above it, the ansatz is approximately fully expressive.

This is the bridge between the *algebraic* expressivity (DLA dimension, asymptotic) and the *practical* expressivity (what depth do I actually need?).

Approximate And Conditional Controllability

For real ansätze, exact controllability is often unattainable, but approximate or conditional versions hold:

1. Approximate controllability. The reachable set is dense in the target manifold but not equal to it. Practically, you can get *arbitrarily close* to any target state but not exactly reach it.

2. Conditional controllability. The reachable set is the full target manifold *given* certain initial conditions (e.g., starting from a specific state, or with parameters in a specific range).

3. Bounded-depth controllability. The reachable set at depth L is some subset of the full target manifold, growing with L.

4. Symmetry-restricted controllability. The reachable set is the orbit of a symmetry group. If your problem has the symmetry, this is enough.

For VQAs, **bounded-depth controllability with symmetry restriction** is the typical practical regime. You don't need to be able to reach every state in Hilbert space — you need to be able to reach every state in the symmetry-allowed sector that contains your problem's solution.

Practical Diagnostics For Controllability

How do you check if your ansatz is controllable for the problem at hand?

1. Compute the DLA. Symbolically close $\{iH_j\}$ under commutators and count the dimension. If the DLA dimension matches the expected target manifold dimension, you have full controllability.

2. Check symmetry conservation. Identify the conserved quantities (operators that commute with all H_j). The reachable set is restricted to the joint eigenspaces of these conserved quantities. Make sure your target state is in the correct eigenspace.

3. Empirical reachability test. Sample many target states from the desired distribution and check whether your ansatz can approximate each one (via local optimization). If most can be approximated, you're effectively controllable.

4. Random parameter coverage. Sample many random parameter values and check the *variety* of states produced. Low variety = limited reachable set.

These diagnostics are operational versions of the formal Lie-algebraic conditions, suitable for ansätze where exact computation of the DLA is not feasible.

Controllability And Trainability — Revisited

A key insight from Module 8 (and the Ragone et al. theorem):

Trainability is inversely proportional to controllability.

- Fully controllable ansatz (DLA = $\mathfrak{u}(2^n)$): unique answer reachable, but barren plateaus everywhere.
- Restricted-controllability ansatz (DLA = small Lie subalgebra): reachable set is smaller, but trainability is better.

The control-theoretic framing makes this very clean: **controllability and trainability are conflicting design objectives, with the DLA dimension as the single parameter that controls both**.

This is the same tradeoff that Module 8.5 stated in expressivity language. The control language gives it a slightly different flavor:

- **Expressivity language:** "How big is the reachable manifold? Bigger = harder to optimize."
- **Control language:** "Which states can the control system reach? More reachable = noisier feedback signal."

Both are correct. The choice is rhetorical.

Time-Optimal Control And Minimum Circuit Depth

A particularly nice application of control theory to VQAs: **time-optimal quantum control**.

Problem. Given a target unitary U_T and a set of control Hamiltonians $\{H_j\}$, find the **minimum time** T^* to implement U_T (within tolerance) using piecewise-constant controls.

Theorem (Khaneja-Brockett-Glaser 2001). For two-qubit systems with a specific control Hamiltonian, the minimum time has an explicit closed form in terms of the singular values of U_T. This is one of the few cases where time-optimal control of quantum systems is solved exactly.

Implication for VQAs. The minimum circuit depth to implement a target state is the discrete-time analogue. Time-optimal control gives **lower bounds** on circuit depth: no ansatz can implement the target faster than the time-optimal control trajectory.

For VQA design, this is useful: it gives you a *target depth* below which you should not bother trying. If your ansatz is at this depth or above, the question is whether the *parameter optimization* can find the right control trajectory.

Structural Role

This node is the **reachability foundation** of Module 7. It introduces classical and quantum controllability, identifies the DLA-based conditions, and establishes the equivalence between control-theoretic controllability and VQA expressivity.

It is the dual of node 7.2 (observability) and a prerequisite for node 7.4 (optimal control), which uses controllability to formulate the cost-minimization problem.

Relations to Other Concepts

- **Chow-Rashevsky theorem** — nonlinear controllability via Lie algebras.
- **Schirmer-Solomon (2001)** — quantum controllability theorem.
- **D'Alessandro (2007)** — *Introduction to Quantum Control and Dynamics.*
- **Khaneja-Brockett-Glaser (2001)** — time-optimal two-qubit control.
- **Module 8.1 (Expressivity)** — the same theorem in different language.

Worked Example — Controllability Of A 2-Qubit XX Ansatz

Consider $U(\theta_1, \theta_2, \theta_3) = e^{-i\theta_3 X_0 X_1} R_Y(\theta_2)_1 R_Y(\theta_1)_0$ — single-qubit R_Y on each qubit followed by an XX coupling.

Generators: - $H_1 = Y_0$ (generator of R_Y on qubit 0). - $H_2 = Y_1$ (generator of R_Y on qubit 1). - $H_3 = X_0 X_1$ (generator of the XX coupling).

Compute the DLA. Take Lie brackets: - $[H_1, H_3] = [Y_0, X_0 X_1] = -2iZ_0 X_1$ (new operator). - $[H_2, H_3] = [Y_1, X_0 X_1] = -2iX_0 Z_1$ (new operator). - $[H_1, [H_1, H_3]] = ...$ more new operators.

Continuing, the DLA closes after several steps and contains: $\{Y_0, Y_1, X_0X_1, Z_0X_1, X_0Z_1, Z_0Y_1, Y_0Z_1, Z_0Z_1, X_0, X_1, ...\}$.

For 2 qubits, the full Lie algebra $\mathfrak{u}(4)$ has dimension 16. The DLA of this ansatz, after closure, is exactly all of $\mathfrak{u}(4)$ — so it is **fully controllable**.

Implication. This 3-parameter ansatz (after enough repetitions) can reach any 2-qubit state. State-universal at sufficient depth.

Trainability. Because the DLA is full $\mathfrak{u}(4)$, the Ragone bound predicts gradient variance $\sim 1/16$ — small but not exponentially small. For 2 qubits, this is fine. For n qubits with similar structure, the bound becomes $\sim 1/4^n$ — barren plateau.

This is a clean instance where controllability theory gives both a positive result (universality) and a negative result (untrainability at scale) from the same calculation.

Visualization

Pending. A diagram showing two ansätze: (a) a controllable one (DLA = full algebra) with reachable set = full Hilbert space; (b) an uncontrollable one with reachable set = a low-dimensional submanifold. Caption: "Controllability is about which states you can steer to."

Exercises

1. For a 1-qubit system with control Hamiltonians $\{X, Z\}$, is it fully controllable? Compute the DLA.
2. For a 2-qubit system with control Hamiltonians $\{X_0, X_1, Z_0Z_1\}$, is it fully controllable? (Hint: this is the standard QAOA-style set.)
3. (VQA) For a 4-qubit system with only $\{Y_0, Y_1, Y_2, Y_3, Z_0Z_1, Z_1Z_2, Z_2Z_3\}$ (linear chain XX-equivalent), compute the DLA dimension. Is the ansatz fully controllable?

Research Extensions

- **Efficient DLA computation** — algorithmic methods for large ansätze.
- **Approximate controllability with explicit error bounds** — quantitative versions of "almost controllable."
- **Time-optimal VQA depth** — combining time-optimal quantum control with VQA design.
- **Controllability under noise** — how does decoherence affect the reachable set?

Cross-links to Other Courses

- **Lie Theory** — Lie algebras, Lie groups, Lie brackets.
- **Differential Geometry** — vector fields, integral curves, manifolds.
- **Classical Control Theory** — Chow-Rashevsky, Kalman controllability.
- **Module 8.1 (Expressivity Theory)** — the same content in expressivity language.
- **Module 7.4 (Optimal Control)** — uses controllability as a precondition.

Optimal Control Theory

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 $|\text{Tr}(U_T^\dagger U_{\text{total}})|^2/d^2$ 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

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

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.

Pontryagin Minimum Principle

Position in Knowledge Graph

System: Variational Quantum Algorithms **Structural Domain:** Control-Theoretic View **Node:** 7.5

The **Pontryagin minimum principle** (PMP) is the central theorem of classical optimal control theory. It gives **necessary conditions for a control trajectory to be optimal** — a generalization of the calculus-of-variations Euler-Lagrange equations to problems with control inputs.

For VQAs, the PMP has both theoretical and practical importance:

1. **Theoretically**, it tells you what an optimum *must* look like — every optimal VQA parameter setting satisfies a specific structural condition derived from the PMP.
2. **Practically**, it gives a *different* way to derive gradient conditions and update rules. The PMP-derived gradient is equivalent to the parameter-shift gradient, but the derivation is cleaner and connects to the broader optimal control literature.
3. **Conceptually**, the PMP makes the "two-point boundary value" structure of VQA optimization explicit, which clarifies why some VQAs are easy to optimize and others are not.

This node walks through the PMP, applies it to VQAs, and discusses the bang-bang structure it predicts for QAOA-like ansätze.

The Statement Of The Pontryagin Minimum Principle

Setup. Consider the optimal control problem

$$\min_{\mathbf{u}} J = \phi(\mathbf{x}(T)) + \int_0^T L(\mathbf{x}, \mathbf{u}, t)\, dt, \quad \dot{\mathbf{x}} = f(\mathbf{x}, \mathbf{u}, t), \quad \mathbf{x}(0) = \mathbf{x}_0.$$

Define the control Hamiltonian:

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

Here $\boldsymbol{\lambda}(t) \in \mathbb{R}^n$ is the **costate** (or "adjoint") variable — a Lagrange multiplier for the dynamics constraint.

Theorem (Pontryagin et al. 1962). If $\mathbf{u}^*(t)$ is optimal and $\mathbf{x}^*(t)$ is the corresponding state trajectory, there exists a costate $\boldsymbol{\lambda}^*(t)$ such that:

1. **State equation:** $\dot{\mathbf{x}}^* = \partial H / \partial \boldsymbol{\lambda}$.
2. **Costate equation:** $\dot{\boldsymbol{\lambda}}^* = -\partial H / \partial \mathbf{x}$.
3. **Minimum condition:** $\mathbf{u}^*(t) = \arg\min_{\mathbf{u}} H(\mathbf{x}^*(t), \mathbf{u}, \boldsymbol{\lambda}^*(t), t)$.
4. **Boundary conditions:** $\mathbf{x}^*(0) = \mathbf{x}_0$ (initial), $\boldsymbol{\lambda}^*(T) = \partial \phi / \partial \mathbf{x}|_{\mathbf{x}(T)}$ (terminal).

Interpretation: the PMP says optimal trajectories are characterized by a *two-point boundary value problem* in the state and costate, and at every instant the control minimizes the Hamiltonian. The costate plays the role of "marginal value of an extra unit of state" — it's the sensitivity of the cost-to-go with respect to the current state.

Why The PMP Is Useful

A few reasons:

1. It is necessary for optimality. Any optimal trajectory satisfies the PMP. So if you find a trajectory that does *not*, it's not optimal.

2. It reduces an infinite-dimensional problem to ODEs. The original problem is over function spaces (the control $\mathbf{u}(t)$), but the PMP gives a finite-dimensional system (state, costate, and a pointwise minimization).

3. It gives structural information. Even without solving, you can read off properties: when the Hamiltonian is linear in the control, the optimum is **bang-bang** (control jumps between extreme values); when it's quadratic, the optimum is **smooth** (linear feedback).

4. It generalizes to constraints, inequality constraints, free terminal time, and stochastic versions.

For VQAs, the PMP gives a non-obvious viewpoint on parameter optimization that is sometimes more natural than gradient descent.

The PMP Applied To VQAs

For a VQA, the optimal control problem is:

- **State:** $|\psi(t)\rangle$, evolving via discretized Schrödinger equation.
- **Control:** parameters $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_P)$, one per gate (or one per gate-time interval).
- **Dynamics:** unitary evolution under the gate Hamiltonians.
- **Cost:** terminal expectation $C = \langle\psi(T)|H|\psi(T)\rangle$, no running cost.

Discretized PMP for VQA:

The "time" is the position in the circuit. The state $|\psi_k\rangle$ at step k is the result of applying the first k gates. The costate $|\lambda_k\rangle$ is propagated *backward* from $|\lambda_T\rangle = H|\psi_T\rangle$ using the adjoint dynamics.

The **gradient with respect to** θ_k is then

$$\frac{\partial C}{\partial \theta_k} = -i\langle\lambda_k|\frac{\partial U_k}{\partial \theta_k}U_k^\dagger|\psi_k\rangle + \text{c.c.},$$

which is exactly the form one gets from the parameter-shift rule. **The PMP-derived gradient and the parameter-shift gradient are the same expression in different notation.**

Why this matters: the costate-based view shows that the gradient computation is doing **forward propagation of the state** plus **backward propagation of the costate**, with a contraction at each step. This is structurally identical to backpropagation in classical neural networks. The "VQA backprop" picture is the PMP applied to circuits.

This is not new — Schuld et al. and others have made the connection explicit — but the PMP formulation is the *most general* statement of the underlying structure.

Bang-Bang Control And QAOA

A particularly important consequence of the PMP: when the control Hamiltonian is **linear in the control variable**, the optimum is **bang-bang** — the control switches between its extreme values.

For QAOA: the ansatz is

$$|\psi(\boldsymbol{\beta}, \boldsymbol{\gamma})\rangle = \prod_k e^{-i\beta_k H_M} e^{-i\gamma_k H_C} |+\rangle^{\otimes n},$$

where H_M is the mixer and H_C is the cost. This is a *piecewise-constant control*: at each step, the control selects either H_M or H_C, and the only choice is the duration.

PMP analysis. The control Hamiltonian for the optimal control problem is

$$H_{\text{ctrl}}(\boldsymbol{\theta}) = i\boldsymbol{\lambda}^\dagger \cdot (H_M \beta + H_C \gamma)|\psi\rangle + \text{c.c.}$$

This is *linear* in the durations β_k, γ_k. So the optimum durations are at the *boundary* of their allowed range — which is the bang-bang result.

Implication: the optimal QAOA (in the unconstrained-duration limit) uses only "extreme" values for β_k, γ_k — it switches abruptly between the cost and mixer Hamiltonians. Real QAOA has constrained durations (typically $[0, \pi]$), which softens the bang-bang prediction but the underlying structure is preserved.

This was first observed by Brandao et al. (2018) and subsequently studied as the **bang-bang QAOA** literature. The connection to Pontryagin gives the theoretical foundation.

Singular Arcs And QAOA Difficulty

A subtlety: when the control Hamiltonian is *exactly* zero in some region (the coefficient of the control vanishes identically), the bang-bang result fails — you have a **singular arc** where the optimal control is determined by higher-order conditions.

For VQAs: singular arcs correspond to **flat regions of the cost landscape** — the gradient (which is the linear coefficient of the control in the Hamiltonian) is identically zero. **Barren plateaus are exactly singular arcs in the optimal control formulation.**

This gives a *third* characterization of barren plateaus, complementary to: - Module 4's measure-theoretic view (Haar 2-design integration). - Module 8's Lie-algebraic view (small gradient variance from large DLA). - Now: **the plateau is a region where the PMP minimum condition is degenerate** — the control variable doesn't appear linearly in the Hamiltonian, so the standard gradient information is gone.

Implication for optimizer design: in a singular arc, you cannot use the PMP-derived gradient. You need either second-order conditions (Hessian) or non-gradient methods (population optimizers, Module 6.7-6.8). This is the optimal-control justification for Module 6's hybrid methods.

The Dual Variable Interpretation

A useful interpretation of the costate $\boldsymbol{\lambda}(t)$: it is the **dual variable** for the dynamics constraint. In Lagrangian terms, $\boldsymbol{\lambda}^T f(\mathbf{x}, \mathbf{u})$ is the penalty for violating $\dot{\mathbf{x}} = f$.

For VQAs, this means the costate $|\lambda_k\rangle$ is the "sensitivity of the final cost to perturbations of the state at step k." Knowing the costate at every step tells you exactly how much each intermediate quantum state matters for the final outcome.

Practical use: the costate identifies the *bottleneck* gates in a circuit — the gates whose perturbations have the largest effect on the final cost. These are the gates that should be parameterized most carefully and updated most aggressively. Gates with small costate norm are not contributing much and can be left alone or removed (this is the basis of **adaptive ansatz** methods like ADAPT-VQE).

Numerical Methods For PMP

Solving the PMP numerically is hard because it's a two-point boundary value problem (state at $t = 0$, costate at $t = T$).

Standard methods:

1. **Shooting.** Guess $\boldsymbol{\lambda}(0)$, integrate forward, check if the terminal condition is satisfied, adjust the guess.
2. **Multiple shooting.** Break the trajectory into segments, with a guess for the state and costate at each segment boundary.
3. **Collocation.** Discretize state and costate as polynomials over time intervals; solve the resulting algebraic system.
4. **Indirect methods generally.** Solve the PMP system directly as a system of nonlinear equations.

Compared to gradient descent. Gradient descent on the parameters is the *direct method* — discretize, then optimize — and is much simpler. The PMP-based methods are theoretically cleaner but numerically harder.

For most VQAs, gradient descent dominates. The PMP is used for *theoretical analysis* (proving things about optimal trajectories), not for the actual optimization.

When The PMP Gives New Information

A practical question: does the PMP ever give VQA practitioners *new* information they couldn't get from parameter-shift gradient descent?

Yes, in three specific situations:

1. Bang-bang structure prediction. Knowing that the optimum is bang-bang lets you initialize at the extreme values, which often gives better convergence than uniform initialization.

2. Singular arc detection. If you compute the costate norm and it's small, you're in a singular region (barren plateau). This is a fast diagnostic for plateaus.

3. Adaptive ansatz design. The costate identifies which gates contribute most. Adaptive ansatz methods (ADAPT-VQE) use this to grow the ansatz one gate at a time.

These are not earth-shattering, but they are concrete benefits. They justify the theoretical investment in understanding the PMP for VQA practitioners.

Stochastic And Quantum Pontryagin

A few extensions of the PMP relevant to VQAs:

1. Stochastic PMP. Generalizes to systems with noise, replacing the deterministic costate with a stochastic process. Used in stochastic optimal control. For VQAs with shot noise, this is the right framework, but the analysis is more complex.

2. Quantum PMP. Bonnard, Sugny, and others have developed PMP variants for quantum systems with explicit unitary constraints. These give time-optimal results for specific quantum gates (e.g., Khaneja-Brockett-Glaser).

3. Chattering (Fuller's phenomenon). In some optimal control problems, the optimal trajectory has *infinitely many* bang-bang switches in finite time. This is a pathological case but it appears in some quantum optimal control problems and is worth being aware of.

Structural Role

This node is the **theoretical capstone** of Module 7's optimal control thread. It introduces the Pontryagin minimum principle, applies it to VQAs, identifies the costate-gradient connection, and explains the bang-bang and singular-arc structures that appear in QAOA-like ansätze.

It directly connects to node 7.6 (feedback control), which uses the PMP framework to design closed-loop optimizer architectures.

Relations to Other Concepts

- **Pontryagin et al. (1962)** — *The Mathematical Theory of Optimal Processes*, the original.
- **Brandão et al. (2018)** — bang-bang QAOA.
- **Bonnard-Sugny** — quantum optimal control via PMP.
- **Module 6.1 (Gradient Descent)** — the practical use of PMP-derived gradients.
- **Module 4 (Barren Plateaus)** — singular arcs as plateaus.

Worked Example — PMP For A 1-Qubit State Preparation

Problem. Steer a 1-qubit state from $|0\rangle$ to $|1\rangle$ using the control Hamiltonian $H(t) = u(t)X$, with $|u(t)| \leq 1$.

Setup. - State: $|\psi(t)\rangle = c_0(t)|0\rangle + c_1(t)|1\rangle$. - Cost: terminal fidelity to $|1\rangle$, $J = -|c_1(T)|^2$. - Dynamics: $i|\dot{\psi}\rangle = u(t)X|\psi\rangle$.

PMP analysis. The control Hamiltonian is linear in u, so the optimum is bang-bang: $u^*(t) = \pm 1$.

Solving. With $u = +1$, the dynamics rotates the state at unit angular velocity around the X axis. To go from $|0\rangle$ to $|1\rangle$, you need to rotate by π. The minimum time is $T^* = \pi$.

Costate. The costate is $|\lambda(t)\rangle = \pm 2\mathrm{Im}[c_1^*]|1\rangle - 2\mathrm{Re}[c_1^*]|0\rangle$ or similar — geometrically, it's the gradient of $-|c_1|^2$ with respect to the state.

Connection to a single-parameter VQA. The minimum time π corresponds to the parameter $\theta = \pi$ in $R_X(\theta)$. The "bang-bang" optimum corresponds to "use the maximum allowed parameter value." This is trivial in this 1-parameter case but generalizes.

For multi-parameter ansätze, the bang-bang structure says: in each gate's parameter, the optimum tends to be at the extremes of the allowed range. This is *not* true in general for VQAs (because the cost is non-convex), but it is the right intuition in many cases.

Visualization

Pending. A diagram of a bang-bang control trajectory: the control $u(t)$ jumps between $+1$ and -1, the state evolves smoothly, the costate is computed by backward propagation. Caption: "Optimum, costate, control: PMP in three lines."

Exercises

1. For an LQR problem, derive the optimal feedback law from the PMP. How does it match the Riccati result?
2. The PMP says the costate satisfies $\dot{\boldsymbol{\lambda}} = -\partial H_{\mathrm{ctrl}}/\partial \mathbf{x}$. For a VQA with terminal cost only, derive the costate dynamics.

3. (VQA) For a QAOA at depth $p = 2$ with a 4-vertex MaxCut, write the discretized PMP system. How many costate variables are there?

Research Extensions

- **PMP-aware VQA optimizers** — using costate information to drive parameter updates beyond simple gradient descent.
- **Singular arc detection in practice** — fast tests for whether you're in a barren plateau using costate norms.
- **Bang-bang VQA ansatz design** — explicitly constructing ansätze that have bang-bang optima for tractable problems.
- **Stochastic PMP for shot-noise VQAs** — formal extension to noisy gradient estimates.

Cross-links to Other Courses

- **Calculus of Variations** — Euler-Lagrange equations, Hamilton's principle.
- **Optimal Control (Lewis-Vrabie-Syrmos)** — the textbook treatment.
- **Quantum Optimal Control (D'Alessandro)** — the quantum-specific variant.
- **Module 7.4 (Optimal Control Theory)** — the parent framework.
- **Module 6 (Optimization Dynamics)** — the practical optimizer side.

Feedback Control in Quantum Systems

Position in Knowledge Graph

System: Variational Quantum Algorithms **Structural Domain:** Control-Theoretic View **Node:** 7.6

This is the **closing node** of Module 7. It is the synthesis: having developed observability (7.2), controllability (7.3), optimal control (7.4), and Pontryagin (7.5) as building blocks, this node combines them into the central object of modern control theory — the **closed-loop feedback controller**.

A feedback controller is a system that **observes** the state, **decides** what control input to apply based on the observation, and **acts**. The defining feature is that the controller's decisions depend on the observed state — closing the loop. This is what makes control systems robust to noise, drift, and disturbances; it is the technological basis of every modern engineered system from cruise control to power grids.

For VQAs, feedback control is the right framework for the *adaptive* extension. A non-adaptive VQA has a fixed optimizer schedule and ignores the system's response. An adaptive VQA observes the system (gradient norms, cost trajectory, hardware drift) and modifies its strategy accordingly. The control-theoretic view says: this is a feedback controller, and you should design it as one — using the same principles that have worked for 70 years in classical engineering.

This node walks through feedback control concepts, the quantum measurement complications, and the implications for adaptive VQA design. It is the natural bridge to Module 10 (VQA as Adaptive Control Systems).

What Feedback Control Is

A **feedback control system** has:

- A **plant** (the system being controlled).
- A **sensor** (measures the plant's output).
- A **controller** (decides the input based on the measurement).
- An **actuator** (applies the input to the plant).

The signal flow is: plant $\rightarrow$ sensor $\rightarrow$ controller $\rightarrow$ actuator $\rightarrow$ plant. The loop is *closed* because the controller's output depends on the plant's measured output.

Compare to open-loop control: in open loop, the controller just runs a pre-computed input sequence and doesn't observe the plant. Open loop is simple but fragile — any disturbance or model error throws it off.

Closed loop is robust because errors get observed and corrected. This is why every real engineered system uses feedback.

The Standard Feedback Block Diagram

The canonical block diagram:

```
+---------+      u      +-------+      x      +--------+   y
|         |------------>|       |------------>|        |---+
| Control |             | Plant |             | Sensor |   |
|         |<------------|       |<------------|        |   |
+---------+    error    +-------+    noise    +--------+   |
     ^                                                     |
     |                  setpoint r                         |
     +-----------------------------------------------------+
```

The controller compares the measured output y against a desired setpoint r, computes an error, and produces a control input u. The plant state x evolves under u, the sensor measures $y = h(x) + \text{noise}$, and the loop continues.

The **controller design problem** is: given the plant's dynamics and the sensor characteristics, design the controller logic so that the closed-loop system has desired properties (stability, fast convergence, disturbance rejection, etc.).

Classical Feedback Designs

A few standard patterns:

1. Proportional-Integral-Derivative (PID).

$$u(t) = K_p \cdot e(t) + K_i \int_0^t e(\tau) d\tau + K_d \cdot \dot{e}(t),$$

where $e(t) = r - y(t)$ is the error. Three tunable gains. Simple, robust, dominates classical industrial control. Most servos, thermostats, and chemical reactors use PID.

2. State feedback. $u = -K\mathbf{x}$ for a constant gain matrix K. Requires full state observation. Optimal under LQR assumptions.

3. Output feedback. $u = -K\mathbf{y}$ — directly from measurements without estimating the state. Simpler but less powerful than state feedback.

4. Observer-based feedback (separation principle). First estimate the state $\hat{\mathbf{x}}$ from measurements (Kalman filter), then apply state feedback to the estimate: $u = -K\hat{\mathbf{x}}$. Decouples state estimation from control design.

5. Model Predictive Control (MPC). At each time step, solve a finite-horizon optimal control problem online, apply only the first action, then repeat. Handles constraints naturally. Used in process control and (more recently) robotics.

6. Adaptive control. The controller's parameters update online as the plant is observed. Useful when the plant model is uncertain or slowly varying. Module 10 takes this up.

Why Quantum Feedback Is Different

Translating these to the quantum setting runs into the basic constraint of quantum measurement: **measurement collapses the state**.

This means:

1. You cannot continuously observe the state. Each measurement destroys the state (or at least disturbs it). To get a continuous trace of the state, you need many copies of the system or a *weak* measurement scheme.

2. The measurement is intrinsically noisy. Even an ideal projective measurement gives a sample from a probability distribution, not the underlying state directly.

3. The feedback delay matters. If the time between measurement and feedback action is non-negligible, the state has already evolved during that delay, and the action is based on stale information.

These constraints have spawned a whole sub-field: **quantum feedback control**. The two main paradigms:

(a) Discrete feedback. Measure projectively, then apply a unitary or reset based on the measurement outcome. Repeat. Used in error correction, state preparation, and discrete quantum computation.

(b) Continuous feedback. Use weak measurements that perturb the state minimally, integrate the measurement record to estimate the state in real time, and apply Hamiltonian feedback proportional to the estimate. The framework is **stochastic master equations** plus **conditional dynamics** — Wiseman-Milburn formalism.

For VQAs, the relevant paradigm is **discrete feedback at the optimizer level**: measure the cost (or gradient), update the parameters, then repeat. The "measurement" is the cost estimation, the "controller" is the optimizer, the "actuator" is the parameter update mechanism.

VQA As A Discrete Feedback Loop

The VQA outer loop is *exactly* a discrete feedback control loop:

Feedback element	VQA element
Plant	Quantum circuit + cost evaluation
Sensor	Cost estimator (with shot noise)
Controller	Optimizer (Adam, GD, HOPSO, etc.)
Actuator	Parameter update
Setpoint	Target cost (often $-\infty$ or known optimum)
Loop frequency	Iterations per second
Loop delay	Time per iteration (state prep + measurement + classical compute)

Implications of this framing:

1. **The optimizer is a controller.** It can be designed using control-theoretic principles, not just machine-learning intuitions.
2. **Loop delay matters.** If iterations are slow, the controller's actions are based on stale information. This affects optimizer choice — fast-converging optimizers (Adam) are better when iterations are slow.
3. **Sensor noise matters.** Shot noise in cost estimation is exactly sensor noise in the feedback loop. It can be filtered (Kalman-style), averaged (running mean), or compensated (Robbins-Monro).
4. **Stability is a concern.** In feedback loops, large gains can cause instability (oscillations, divergence). In VQAs, large learning rates have the same effect. The control framing tells you to think about gain scheduling and stability margins, not just convergence rates.
5. **Disturbance rejection is a concern.** Hardware drift acts as a disturbance on the plant. A good optimizer should reject this disturbance — adapt as the hardware changes. This is the adaptive control problem.

The Separation Principle For VQAs

Classical control theory has a beautiful result: under linearity and Gaussian noise, the optimal controller can be **separated** into:

1. A **state estimator** (Kalman filter) that produces a best-estimate state from the measurement record.
2. A **feedback law** (LQR-style) that acts on the estimate.

The two can be designed *independently* and the result is jointly optimal. This is the **separation principle**.

For VQAs, the natural analogue:

1. **State estimator:** an estimator that maintains a posterior over the parameter values (or the cost landscape) given the noisy cost evaluations.
2. **Feedback law:** an optimizer that acts on the posterior estimate.

A naive VQA optimizer skips the estimator and feeds raw measurements into the controller — equivalent to a "no-filter" controller. Adding even a simple moving-average filter (or a Kalman-like estimator) significantly improves performance under noise.

Bayesian VQA optimizers (Bayesian optimization, Gaussian process bandits) implement this separation explicitly: they maintain a posterior over the cost landscape and update parameters based on the posterior. They are the cleanest example of separation-principle thinking in VQAs.

Robust Control For VQAs

A more advanced control-theoretic concept: **robust control** handles uncertainty in the plant model. Classical methods include H_∞ control, μ-synthesis, and Lyapunov-based robust design.

For VQAs, the "uncertainty" comes from:

1. **Hardware drift** (gates calibrate slightly differently over time).
2. **Noise model uncertainty** (you don't know the exact noise rates).
3. **Cost evaluation noise** (shot noise, plus systematic errors).
4. **Optimizer hyperparameter uncertainty** (you don't know the right learning rate).

A robust VQA controller would be designed to **achieve good performance for all plants in some uncertainty set**. This is conservative but reliable — and it is the right framework when you can't trust your plant model perfectly.

Practical robust techniques for VQAs:

- **Conservative learning rates:** stay below the stability threshold for all plausible plants.
- **Trust regions:** limit the parameter step size so a single bad measurement doesn't move you far.
- **Worst-case cost evaluation:** average over multiple shots or multiple circuits to reduce noise sensitivity.
- **Adaptive robust controllers:** learn the plant uncertainty online and tighten the bounds as more data arrives.

These are not standard in the VQA literature but are routine in classical control. Importing them is one of the underexploited bridges between fields.

Adaptive Optimizers As Adaptive Controllers

The thesis's central claim: **adaptive VQA optimizers (HOPSO, Adam with gain scheduling, regularized SGD) are adaptive controllers**, and they should be designed as such.

The argument:

1. The standard fixed-hyperparameter optimizer is the equivalent of an open-loop controller — runs a pre-computed schedule, ignores system response.

2. An optimizer that observes its own performance (gradient norms, cost trajectory, gradient variance) and modifies its hyperparameters online is a closed-loop adaptive controller.

3. The mature literature on adaptive control gives explicit design principles, stability theorems (Lyapunov-based), and convergence guarantees that apply with appropriate translation.

4. Most VQA optimizers are designed by hand-tuning rather than by adaptive control principles. There is significant room for principled improvement.

This is the bridge to **Module 10**, which takes the adaptive control framing and develops it into specific architectures for VQA deployment.

Quantum Feedback Control: A Brief Survey

For completeness, the dominant approaches in *quantum* feedback control (the inner-loop, gate-level version):

1. Wiseman-Milburn continuous quantum measurement. Weak measurements with Wiener-process measurement records. Stochastic master equations describe the conditional dynamics. The optimal controller is a function of the measurement record.

2. Quantum error correction. Repeatedly measure stabilizers, decode the syndrome, apply correction. The discrete version of feedback. Surface codes, color codes, Bacon-Shor codes all use this paradigm.

3. Measurement-based quantum computation. The whole computation is measurement and feed-forward. Each measurement determines the next, and the final state encodes the result. Inverts the usual "prepare then measure" structure.

4. Quantum Kalman filtering. Uses Wiseman-Milburn formalism to maintain a real-time state estimate, then applies feedback to drive the estimate toward a target. Used in cooling experiments and quantum metrology.

These are not VQA techniques per se, but they show that quantum feedback control is a well-developed field with established methods. Importing them into VQA design (especially for hardware-aware optimizers) is an open research direction.

Closing Module 7

Module 7 has now traversed the control-theoretic view of VQAs:

- **7.1:** The framing — VQA as a control system, two-loop structure.
- **7.2:** Observability — what we can learn from measurements, gauge symmetries.
- **7.3:** Controllability — what states we can reach, the DLA equivalence with expressivity.
- **7.4:** Optimal control theory — the algorithmic core, GRAPE, Krotov, CRAB.
- **7.5:** Pontryagin minimum principle — the structural conditions for optimality.
- **7.6 (this node):** Feedback control — closing the loop, adaptive optimizers.

The arc: from passive structural analysis (observability, controllability) to active design (optimal control, PMP) to closed-loop adaptation (feedback). Each step adds a layer that the previous one lacked.

The thesis's central claim from this module: **VQAs are control systems, and 70 years of control theory has techniques the VQA community has not fully imported**. Modules 8 (theory) and 10 (deployment) build on this premise.

Structural Role

This node is the **closing synthesis** of Module 7. It combines observability, controllability, and optimal control into the closed-loop feedback framework, identifies VQA optimization as a discrete feedback loop, and previews the adaptive control deployment of Module 10.

It is also the strongest argument for the control-theoretic view: closed-loop adaptive controllers are the technology that makes modern engineering work, and importing them into VQA design is one of the best-leveraged research directions for making VQAs robust to real hardware.

Relations to Other Concepts

- **Wiener (1948)** — *Cybernetics*, founding text.
- **Kalman (1960)** — Kalman filter, separation principle.
- **Wiseman-Milburn (2010)** — *Quantum Measurement and Control*.
- **Module 6 (Optimization Dynamics)** — the optimizer-as-controller view.
- **Module 10 (Adaptive Control Systems)** — the deployment endpoint.

Worked Example — A Feedback VQA Optimizer

Consider a VQA optimizer that:

1. Estimates the cost $\hat{C}_k$ at iteration k.
2. Computes a moving-average filtered cost $\bar{C}_k = \alpha\bar{C}_{k-1} + (1-\alpha)\hat{C}_k$ (Kalman-like filter).
3. Uses $\bar{C}_k$ (not $\hat{C}_k$) to compute the gradient via parameter shift.
4. Adapts the learning rate based on the variance of recent gradients: low variance $\rightarrow$ larger step, high variance $\rightarrow$ smaller step.
5. Detects "stuck" behavior (cost not decreasing) and applies a random kick to the parameters.

As a feedback controller: - **Plant:** the quantum circuit + Hamiltonian. - **Sensor:** raw cost estimator (noisy). - **Filter:** the moving average — denoises the sensor. - **Controller:** the gradient-based parameter update with adaptive gain. - **Actuator:** parameter modification. - **Disturbance rejection:** the random kick when stuck.

This is *exactly* how a competent classical control engineer would design a controller for a noisy plant. None of the techniques are novel in control theory; all are well-established. But applied to VQAs, this kind of design is uncommon — most optimizers are hand-tuned with a single learning rate and no filtering.

The control framing motivates the use of these techniques and gives the principled justification.

Visualization

Pending. A block diagram of a closed-loop adaptive VQA optimizer, with all the control components labeled (plant, sensor, filter, controller, actuator, adaptation, disturbance rejection). Caption: "VQA optimization, redrawn as classical control."

Exercises

1. For a VQA optimizer with cost noise std σ, what is the equivalent "sensor noise" in a classical feedback loop? How would a Kalman filter improve it?
2. The separation principle says estimator and controller can be designed independently under LQG assumptions. For a VQA, what would the "estimator" be?
3. (VQA) Design a feedback controller for a VQA optimizer that handles slow hardware drift (gate fidelities changing on the hour timescale). Specify the sensor, filter, controller logic.

Research Extensions

- **Kalman-filtered VQA optimizers** — explicit Kalman filtering of the cost trajectory.
- **Robust control for hardware-noise VQAs** — applying H_∞ techniques to optimizer design.
- **MPC-style VQA optimizers** — finite-horizon lookahead with constraint handling.

- **Quantum feedback at the gate level for VQAs** — combining inner-loop quantum feedback with outer-loop optimization.

Cross-links to Other Courses

- **Classical Control (Ogata)** — PID, state feedback, observer-based feedback.
- **Quantum Measurement and Control (Wiseman-Milburn)** — quantum-specific feedback theory.
- **Adaptive Control (Åström-Wittenmark)** — the parent literature for Module 10.
- **Module 6 (Optimization Dynamics)** — the optimizer-only view.
- **Module 10 (Adaptive Control Systems)** — the deployment chapter.