VQA as System Identification

Variational Quantum Algorithms · Module 1 — Hybrid Quantum Classical Systems

Node 1.6

Position in Knowledge Graph

System: Variational Quantum Algorithms **Structural Domain:** Hybrid Quantum-Classical Systems **Node:** 1.6

This node offers a reframing: **variational quantum algorithms are a special case of system identification.** The classical optimizer is trying to learn the shape of an unknown function through noisy probes, and the structure of that problem is the same whether the function happens to be a physics Hamiltonian expectation, an industrial plant's response curve, or a simulator's reward landscape.

This framing is not just cosmetic. It lets you import a century of control-theory and sysid literature into VQA, and it foreshadows Module 7 (control-theoretic view).

System Identification in One Paragraph

System identification is the classical engineering discipline of building a model of an unknown dynamical system by observing its input-output behavior. You apply known inputs, measure the outputs, and infer a model — either a parametric model (fit the parameters of a chosen structure) or a non-parametric one (estimate the response function directly).

The three defining features of a sysid problem are:

1. You have **query access** but not **introspection access**. You can poke the system and see what it does, but you cannot look inside.
2. Observations are **noisy**. Every measurement carries sensor or sampling error.
3. The **query budget is finite**. You cannot run experiments forever.

Every one of these bullets applies to VQA verbatim if you replace "unknown dynamical system" with "the landscape $C(\boldsymbol{\theta})$ generated by a quantum circuit."

The VQA-Sysid Map

Sysid concept	VQA instance
Unknown system	The cost landscape $C(\boldsymbol{\theta})$ generated by quantum circuit + Hamiltonian
Input	Parameter vector $\boldsymbol{\theta}_k$
Output	Noisy scalar $\widehat{C}(\boldsymbol{\theta}_k)$
Sensor noise	Shot noise + hardware errors
Query budget	Shot budget
Model class	Implicit in the optimizer's update rule
Identification goal	Find $\arg\min_{\boldsymbol{\theta}} C(\boldsymbol{\theta})$

The last row is where VQA differs slightly from classical sysid: in most sysid applications the goal is to build an accurate model for downstream use (prediction, control), whereas in VQA the goal is to find an optimal input without necessarily caring whether you have learned the full landscape.

This makes VQA a special case of **optimization-driven system identification** — a subfield of sysid where you only care about model accuracy near the optimum. Bayesian optimization is the classical archetype.

Why This Reframing Is Useful

It makes the right quantities the right quantities. A sysid practitioner intuitively knows that the relevant cost metric is "queries used to reach a given accuracy," not "iterations of my inner loop." For VQA this maps to "shots used to reach a given energy accuracy." Getting this x-axis right (see node 5.9, metric 4) is a common place where VQA benchmarks go wrong.

It tells you when classical intuition fails. Most sysid literature assumes linear, time-invariant, or at most smoothly nonlinear systems. VQA landscapes are none of those. But the *framework* of sysid — query, observe, update belief, query again — transfers cleanly. You just need nonlinear/non-parametric identification tools.

It connects to control theory. Sysid is the data-acquisition half of the sysid-plus-control pipeline. Module 7 picks up the other half: once you have identified enough of the landscape to act, how do you design update rules that steer the system effectively? The control framing makes regularization, stability analysis, and disturbance rejection (Module 5) look like standard techniques rather than ad-hoc tricks.

It motivates query-efficient algorithms. In sysid, you care about sample complexity — how many queries are needed in the worst case. This is exactly the shot budget question for VQA. Methods from bandit theory (UCB, Thompson sampling), Bayesian optimization (Gaussian process surrogates), and active learning are all candidates — and many of them are being actively studied as VQA optimizers.

What VQA Is Not

It is worth naming what this framing does **not** say.

- VQA is not **identifying the Hamiltonian**. The Hamiltonian is known; it is an input to the problem. VQA is identifying the response surface $C(\boldsymbol{\theta})$ that the known Hamiltonian induces through a known ansatz.
- VQA is not **quantum-state tomography**. Tomography would mean reconstructing the full state $|\psi(\boldsymbol{\theta})\rangle$ at each $\boldsymbol{\theta}$, which is exponentially expensive. VQA collapses the state to a single number at each query.
- VQA is not **learning a quantum channel**. The channel (the circuit) is fixed and known. VQA is just searching over its input parameters.

The "unknown" in VQA's sysid interpretation is only the **function** from parameters to expectation values — its local minima, its curvature, where to step next. Nothing quantum remains unknown from the optimizer's perspective. This is the same observation that node 1.7 makes in different language.

Structural Role

This node is the **conceptual bridge** from Module 1 to Module 7 (Control-Theoretic View). It introduces the sysid framing lightly, without the full machinery. Module 7 will formalize it: observability, controllability, the Pontryagin principle applied to VQA trajectories.

It also sets up node 1.9 (separation of objective and dynamics), which is the thesis-flavor version of the same decomposition: the "system" (quantum substrate) and the "identifier" (classical optimizer) are distinct, and you can study them separately before putting them back together.

Relations to Other Concepts

- **Bayesian optimization** — builds an explicit surrogate (Gaussian process or similar) of the unknown function and uses it to decide where to query next.
- **Active learning** — choosing queries to maximize information gain; relevant to shot allocation.
- **Multi-armed bandits** — each parameter location is an arm with unknown reward distribution.
- **Reinforcement learning** — policy-gradient methods are structurally similar; see Module 10.
- **Gaussian process regression** — a common sysid surrogate model; used in Bayesian optimization for VQA.

Worked Example — A Surrogate-Model Optimizer

Consider this simple VQA-sysid algorithm:

1. Sample K random parameter points $\boldsymbol{\theta}^{(1)}, \ldots, \boldsymbol{\theta}^{(K)}$.
2. Evaluate $\widehat{C}(\boldsymbol{\theta}^{(i)})$ for each.
3. Fit a Gaussian process surrogate $\widetilde{C}(\boldsymbol{\theta})$ with estimated mean and variance.
4. Choose the next query as $\boldsymbol{\theta}^{(K+1)} = \arg\min_\theta[\widetilde{C}(\boldsymbol{\theta}) - \kappa\sigma(\boldsymbol{\theta})]$ — the lower confidence bound.
5. Evaluate, add to the training set, refit.
6. Repeat.

This is Bayesian optimization applied to VQA. It is pure sysid: the GP is an explicit model of the unknown landscape, and queries are chosen to trade off exploration (high σ) against exploitation (low $\widetilde{C}$). It works surprisingly well on small VQE instances and is a live research direction.

Visualization

Pending. Diagram: a black box labeled "quantum device" with an input arrow $\boldsymbol{\theta}$ and an output arrow $\widehat{C}(\boldsymbol{\theta})$. Around it, a classical layer that maintains a "belief" about the shape of $C(\boldsymbol{\theta})$ (shown as a dotted contour map) and updates the belief with each query. Arrows feeding back to choose the next query.

Exercises

1. Given a fixed shot budget B, is it better to use all shots for one very accurate evaluation or split them across many noisier evaluations? Argue both sides for an exploration phase and an exploitation phase.
2. Describe a VQA optimizer that would be natural from a pure bandit perspective. What is the reward, what is the regret metric, and what assumption lets you bound regret?
3. (VQA) Why is Bayesian optimization's $O(K^3)$ classical cost per iteration a concern for large-parameter VQAs? What is the tradeoff between quantum shot cost and classical surrogate cost?

Research Extensions

- **Quantum kernel methods** — using quantum circuits as feature maps inside classical sysid pipelines.

- **Information-theoretic shot allocation** — querying where the surrogate's uncertainty is highest, weighted by expected reduction in posterior variance.
- **Meta-learning over problem families** — learning a good warm-start distribution across many VQE instances.

Cross-links to Other Courses

- **Control-Theoretic View (Module 7)** — the natural next step.
- **Statistics Module 6** — Bayesian inference and Gaussian process regression.
- **Game Theory / Bandits** — exploration-exploitation tradeoffs.
- **Systems Thinking** — sysid as a general discipline, not just a VQA trick.

Variational Quantum Algorithms · Module 1 — Hybrid Quantum Classical Systems · Node 1.6
Concepts: dynamical-systems, state-space, observability, cost-function
Prerequisites: 1.2, 1.5
Enables: 1.9

hbar.university — own.your.cognition