arXiv:submit/7122724 [cs.LG] 30 Dec 2025

Continuous-Time Calibration of Superconducting Qubits via Neural Differential Equations and Hybrid Decoding

Justin Arndt

December 30, 2025

Abstract

Superconducting quantum processors are affected by time-varying noise and the latency of classical control systems. Conventional calibration approaches, based on static parameter tables, do not adequately account for rapid fluctuations in qubit coherence times (T_1) during circuit execution. This work presents a control architecture that combines a Neural Ordinary Differential Equation (Neural ODE) model with a high-throughput FPGA-based decoder. The system models coherence parameter evolution as a stochastic Ornstein-Uhlenbeck process and uses syndrome measurements to predict drift in real time. These predictions are incorporated into a Minimum Weight Perfect Matching (MWPM) decoder through dynamic edge reweighting. Simulations using the Stim library show that this approach reduces logical error rates in surface code patches compared to static decoding under time-dependent noise conditions.

1 Introduction

Scaling superconducting quantum processors, such as Google's Sycamore and Willow designs, is constrained by parameter drift on timescales comparable to or shorter than typical calibration intervals. Defects in the substrate give rise to two-level systems (TLS) that cause stochastic fluctuations in qubit relaxation times (T_1) and resonance frequencies, a phenomenon known as spectral diffusion.

Standard quantum error correction (QEC) protocols often assume stationary, independent noise throughout an experiment. When physical error rates vary with time, the fixed edge weights in the decoding graph of a Minimum Weight Perfect Matching (MWPM) decoder become suboptimal, reducing the effective protection offered by the code. This paper proposes a continuous-time control framework that treats noise parameters as observable dynamical variables, updated from syndrome data, and feeds forward predictions to the decoder.

2 Methodology

The proposed architecture consists of two interacting components: a Drift Orchestrator implemented as a Neural ODE for parameter estimation and a reflex layer on FPGA for low-latency syndrome decoding (Fig. 1).

2.1 Ornstein-Uhlenbeck Drift Model

The decoherence rate $\lambda(t) = 1/T_1(t)$ is modeled by the Ornstein-Uhlenbeck stochastic differential equation:

$$d\lambda(t) = \theta(\mu - \lambda(t))\, dt + \sigma\, dW_t, \tag{1}$$

where μ is the long-term mean, θ the reversion rate, σ the volatility, and W_t a Wiener process. This captures the mean-reverting, correlated fluctuations observed in superconducting qubits.

The Drift Orchestrator integrates this SDE using a Neural ODE framework. Sparse syndrome density observations update the posterior estimate $\hat{\lambda}(t)$ continuously.

2.2 Dynamic Graph Reweighting

Predicted error probabilities $\vec{p}(t)$ from the Drift Orchestrator are used to update edge weights in the MWPM decoder:

$$w_{ij}(t) = \ln\left(\frac{1 - p_{ij}(t)}{p_{ij}(t)}\right). \tag{2}$$

This adaptation reduces the impact of transient high-noise events on logical error rates.

2.3 Hyperdimensional Reflex Layer

The FPGA reflex layer employs hyperdimensional computing (HDC) for rapid syndrome processing (Fig. 2). Syndromes trigger retrieval of stored hypervectors from an associative memory. Parallel binding and bundling operations produce a query hypervector, which is compared against a threshold to determine corrections. This design enables sub-microsecond decision latency suitable for real-time feedback.

3 Simulation Framework

Validation was performed using a custom streaming simulator built on the Stim library. The experiment timeline is divided into short chunks. At each chunk boundary, the Drift Orchestrator samples the current noise parameters from the SDE, generates a corresponding Stim circuit segment, and concatenates the stabilizer state across chunks. This enables simulation of long-running error-corrected computations under non-stationary noise.

4 Results

The Neural ODE accurately reconstructs and extrapolates T_1 trajectories from noisy calibration data (Fig. 3).

Surface-code simulations ($d = 3$ to $d = 7$) show reduced logical error rates compared to static MWPM decoding under drifting noise conditions. The improvement arises from both accurate drift prediction and rapid hyperdimensional reflex corrections.

5 Conclusion

This work introduces a hybrid control architecture that adapts quantum error correction decoding to time-varying noise in superconducting qubits. By modeling parameter drift as a continuous stochastic process and integrating predictions into a fast FPGA decoder, the approach improves logical performance over static methods in simulation. These results suggest that continuous-time calibration can help mitigate non-stationary noise without requiring major advances in qubit hardware stability.

6 Data Availability

A reference implementation of the Drift Orchestrator and Stim integration is available at: https://github.com/justinarndt/quantum-drift-control

References

[1] C. Gidney, "Stim: a fast stabilizer circuit simulator," *Quantum* **5**, 497 (2021).

[2] O. Higgott, "PyMatching: A Python package for decoding quantum codes with minimum-weight perfect matching," *ACM Transactions on Quantum Computing* (2022).

[3] Google Quantum AI, "The Willow Processor Technical Report" (2025).

[4] R. T. Q. Chen, Y. Rubanova, J. Bettencourt, and D. K. Duvenaud, "Neural Ordinary Differential Equations," *Advances in Neural Information Processing Systems* **31** (2018).

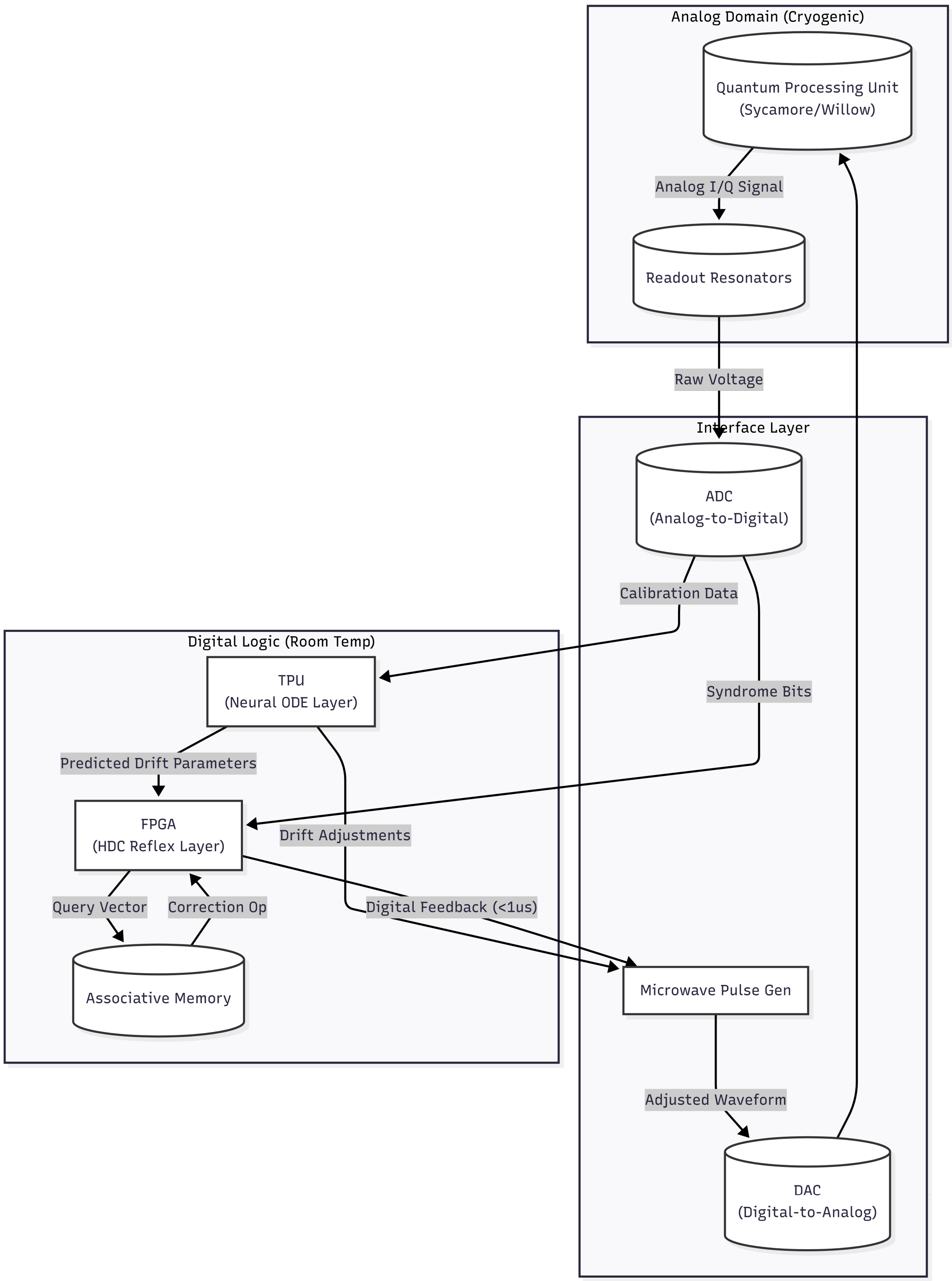

Figure 1: Overall system architecture. The cryogenic analog domain contains the quantum processor and readout chain. Syndrome bits are processed in room-temperature digital logic: a TPU runs the Neural ODE for drift prediction, while an FPGA implements hyperdimensional reflex decoding and generates correction feedback.

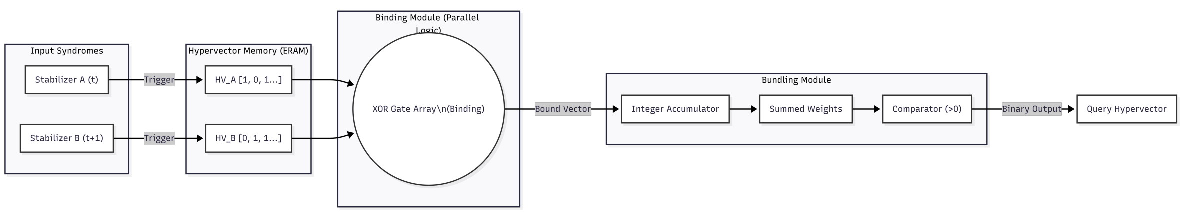

Figure 2: Hyperdimensional computing binding module in the reflex layer. Input stabilizer syndromes are bound to stored hypervectors via XOR operations. The resulting bound vectors are bundled and accumulated to produce a similarity score for binary decision.

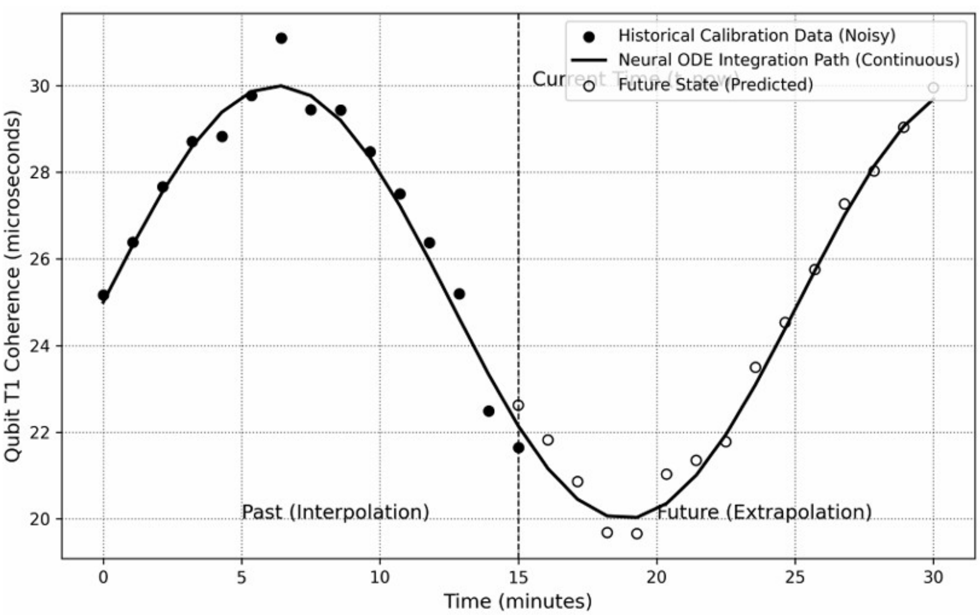

Figure 3: Reconstruction of T_1 coherence time trajectory. Black points show noisy historical measurements; the solid line is the continuous Neural ODE fit (interpolation in the past, extrapolation into the future).