Research Plan: Quantum Circuits for Open System Dynamics: Simulating the GKSL Master Equation

6-Week Internship Roadmap

Project Synopsis

This project develops a modular Python/Qiskit library to simulate open quantum-system dynamics (GKSL master equation and related noise channels) on quantum hardware and simulators. Participants will reformulate GKSL Master Equation into quantum-amenable linear systems, implement Hamiltonian- and variational-based simulation routines, and benchmark them against classical methods. Emphasis will be on resource analysis (qubit count, circuit depth) and error scaling.

Week-by-Week Plan

Week 1: Foundations & Theory

- **Literature Reading**:
  - Gorini–Kossakowski–Sudarshan–Lindblad (GKSL) master equation and Lindblad operators.
  - Standard noise channels (amplitude damping, dephasing, depolarizing).
- **Time-Evolution Methods**:
  - Trotter–Suzuki decomposition for discretizing continuous evolution.
  - Overview of amplitude encoding and "schrödingerization."

Week 2: Alternative Formulations

- **Kraus-Operator Formalism**:
  - Derive Kraus maps for common noise channels.
  - Design quantum circuits implementing Kraus operators via ancilla.
- **Stochastic Differential Equations (SDEs)**:
  - Euler–Maruyama update for unraveling GKSL to stochastic Schrödinger form.
  - Practical amplitude-encoded state updates.

Week 3: Prototype Implementation & Verification

- **Model Conversion**:
  - Convert GKSL master equation into SDE form using Euler–Maruyama.
  - Express evolution as a sequence of Kraus-map circuits.
- **Validation**:
  - Verify single-qubit amplitude damping and dephasing against analytical and SciPy ODE solutions.
  - Compute fidelities and trace distances.

Week 4: Quantum Simulation Methods

- **Hamiltonian Simulation Track**:
  - Implement Trotter-based simulation of the effective non-Hermitian generator via dilation to a Hermitian Hamiltonian.
  - Benchmark circuit depth vs. accuracy.
- **Linear-System Track**:
  - Formulate implicit time-stepping (e.g. Crank–Nicolson) as $A\,\psi^{n+1} = B\,\psi^n$.
  - Explore quantum linear-system algorithms (HHL or variational) to solve $A\,x = b$.
- Teams will compare performance and resource costs of both approaches.

Week 5: Optimization & Open-Source Preparation

- **Performance Tuning**:
  - Optimize Trotter step sizes, variational ansätze, and mid-circuit measurements to reduce qubit/anilla overhead.
  - Assess error-mitigation strategies (zero-noise extrapolation).
- **Repository Assembly**:
  - Organize code into a modular library with clear API.
  - Write usage examples and Jupyter notebooks.
  - Draft documentation (README, installation guide, tutorials).

Week 6: Finalization & Future Work

- **Final Tasks**:
  - Complete remaining benchmarks on simulators and (if available) real devices.
  - Polish documentation and examples.
  - Perform a final code review and ensure reproducibility.
- **Dissemination**:
  - Prepare a presentation summarizing methods, results, and lessons learned.
  - Draft an outline for a workshop paper or preprint.
- **Future Directions**:
  - Extend to multi-qubit chains and interacting spin networks.
  - Integrate QMPS (tensor-network) compression for larger systems.
  - Publish the library to `pip` and contribute to Qiskit Dynamics.

Mentorship & Workflow

- Weekly one-hour video calls for technical guidance and progress review.
- Code reviews via GitHub pull requests.
- Shared Kanban board for task tracking.

Expected Deliverables

- A Python/Qiskit library for open-system simulation, with tutorial notebooks.
- Benchmark reports comparing quantum vs. classical approaches.
- Presentation slides and draft manuscript outline.

Mentor: Haris Ansari
Contact: harishaseebat92@gmail.co