Microarchitecture and Pulse Compilation

Daniel Enright

ESCAL
Extreme Storage & Computer Architecture Lab

UC RIVERSIDE

Outline

- Overarching Questions
- Overview
- Pulse Compilation Flow and Review
- Quantum Controls and Pulse Shaping
- Quantum Optimal Control
- Summary

Overarching Questions

- What kind of microarchitecture can keep up with the speed and bandwidth of quantum processing technology?
- How do we build a reliable interface between classical control/feedback signals and quantum data?
- Can we efficiently translate and synchronize machine pulses from gate instructions?

Overview

- We will explore classical and quantum control of qubits, pulse generation and optimization, and calibration and verification.
- First, we will discuss constructing pulse sequences.
- Then we will discuss the progress and challenges in managing the classical and scalable quantum datapath with timing, energy, and bandwidth constraints.
- Finally, we will discuss translating quantum gates to hardware pulses.

General Pulse Compilation Flow

- As we know from Ch.2, qubits are controlled by analog pulses
- Compilation must translate device-independent high-level quantum programs to device-dependent control pulses
  - Similar to compiling programs written in C for both x86 and ARM platforms.

General Pulse Compilation Flow (cont)

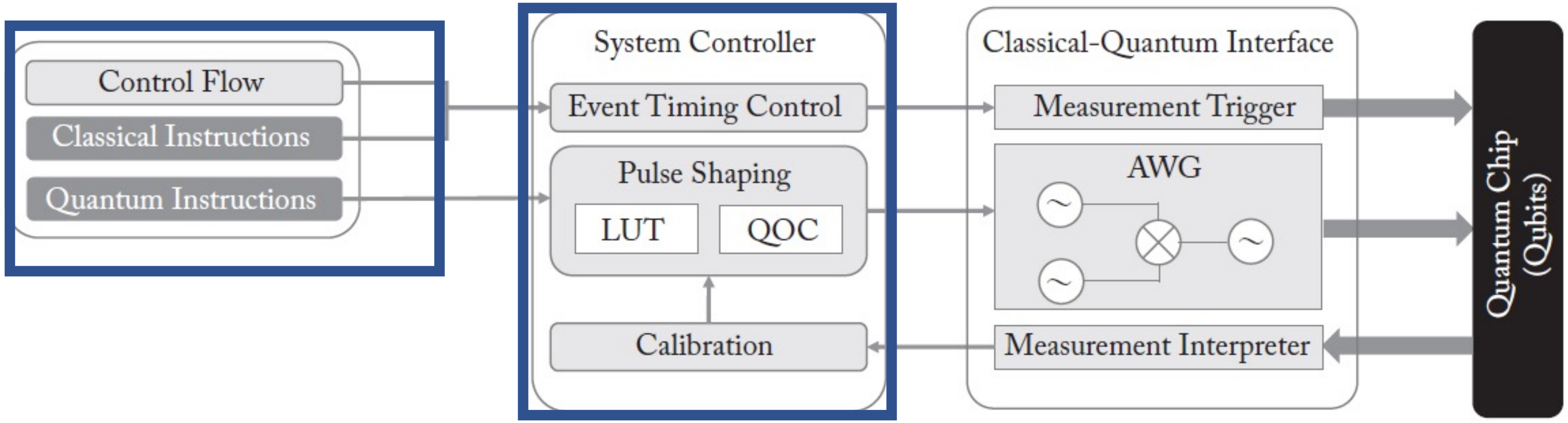


Figure 7.1: General flow for translating quantum gates to analog pulses.

Motivation for Robust Quantum Control

- Rapid state transfer
- High-fidelity gate operations

Extreme Storage & Computer Architecture Lab

Quantum Controls and Pulse Shaping

- Open and closed loop control

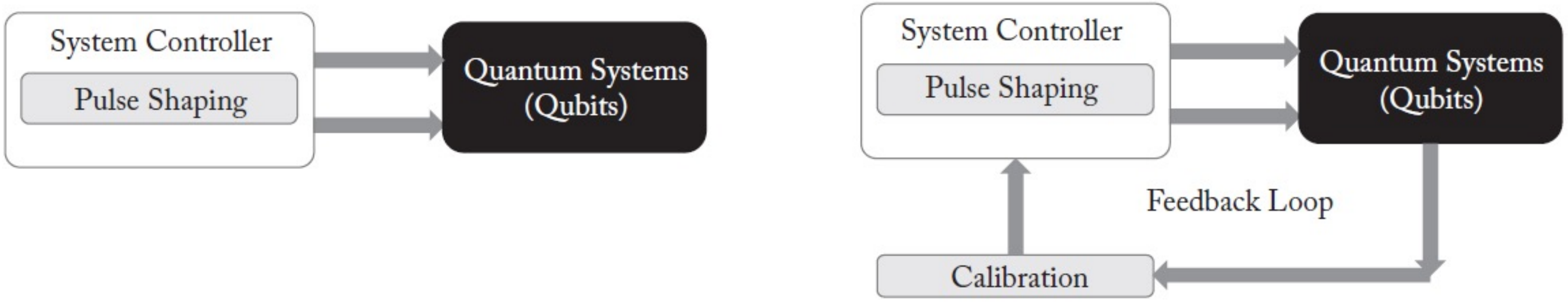
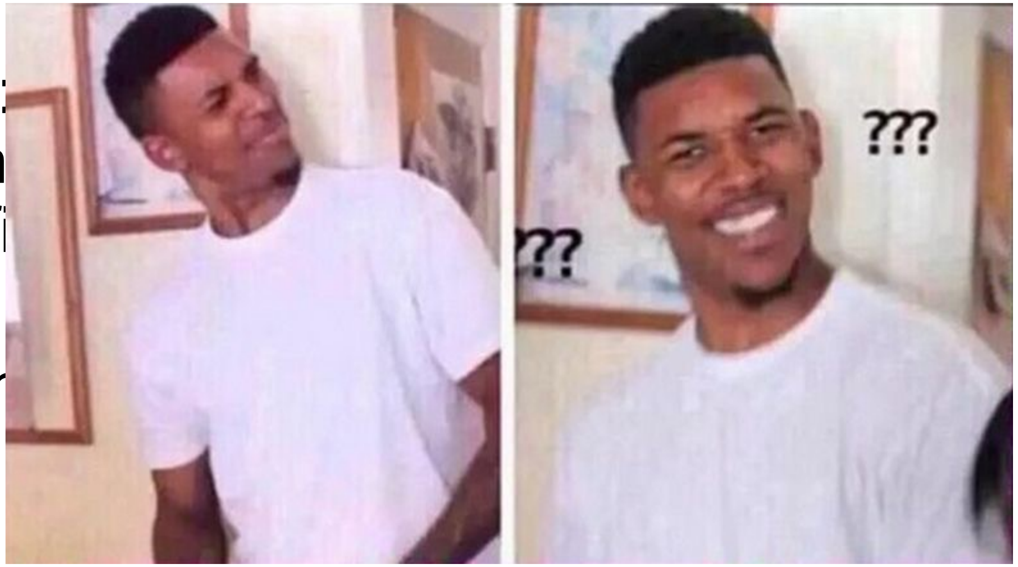


Figure 7.2: Open-loop control (left) vs. closed-loop control for pulse shaping.

Quantum Controls and Pulse Shaping (cont)

- Current work on open-loop quantum system controllers
  - Hamiltonians – optimality and reachability of pulses for different quantum systems. We can express controllability criteria in terms of structures in "Lie groups and Lie algebras[276] or in terms of graph theoretical concepts [277, 278]."
  - Numerical Optimal control theory: ... e best way of achieving given quan... with the most realistic circuit confi...
    - AKA open-loop coherent control
  - Lyapunov-based controller design ... quantum states.
    - (seems like closed loop control?)



Hamiltonian

- The Hamiltonian of a system is the sum of the kinetic energies of all the particles, plus the potential energy of the particles associated with the system.

Lie Algebras/Groups and controllability

- We need to first understand Lie groups and Lie algebras.
  - Basically, we can over-simplify a Lie algebra to a reduction of a vector space over a vector field by one dimension using a commuter. The commuter can be any number of operations that achieve the same result. This operation must also satisfy a few axioms listed in the textbook, but not here. For example:

Perhaps the simplest nontrivial example of a Lie algebra is the set of vectors in three dimensional space, $\mathbb{R}^3$, with the cross product playing the role of the commutator. If we choose a basis $\{\vec{i}, \vec{j}, \vec{k}\}$, then the commutation relations are given by

$$\vec{i} \times \vec{j} = \vec{k}, \qquad \vec{j} \times \vec{k} = \vec{i}, \qquad \vec{k} \times \vec{i} = \vec{j}. \tag{3.5}$$

Lie Groups

- A defined point or vector space where units can be multiplied and their inverses taken.

Dynamical Lie Algebra

- This is the controllability criterion. The book describes it:

Theorem 3.2.1 *The set of reachable states for system (3.1) is the connected Lie group associated with the Lie algebra* $\mathcal{L}$ *generated by* $\text{span}_{u \in \mathcal{U}}\{-iH(u)\}$. *In short*

$$\mathcal{R} = e^{\mathcal{L}}.$$

The Lie algebra $\mathcal{L}$ is called **the dynamical Lie algebra** associated with the system. This is always a subalgebra of $u(n)$. In the case $\dim(\mathcal{L}) = n^2 = \dim(u(n))$, which is equivalent to $\mathcal{L} = u(n)$ and $e^{\mathcal{L}} = U(n)$, the system is said to be **controllable**. In this case $\mathcal{R} = U(n)$, which means

Quantum Controls and Pulse Shaping (cont)

- Current work on open-loop quantum system controllers
  - Hamiltonians – optimality and reachability of pulses for different quantu, systems. We can expressed controllability criteria in terms of structures in the “Lie groups and Lie algebras[276] or in terms of graph theoretical concepts [277, 278].”
  - Numerical optimal control theory: numerical methods to search for the best way of achieving given quantum objectives in the shortest time with the most realistic circuit configuration.
    - AKA open-loop coherent control
  - Lyapunov-based controller design: control input is determined by the quantum states.
    - (seems like closed loop control?)

Coherent Control

- The goal is to control quantum interference by shaping the behavior of pulses. (light or other forms of radiation)
- Very basically, coherent control is a method to transform a quantum system from an initial state to a target state.
  - The foundation for quantum gates.

Quantum Controls and Pulse Shaping (cont)

- Current work on open-loop quantum system controllers
  - Hamiltonians – optimality and reachability of pulses for different quantum systems. We can express controllability criteria in terms of structures in the “Lie groups and Lie algebras[276] or in terms of graph theoretical concepts [277, 278].”
  - Numerical Optimal control theory: numerical methods to search for the best way of achieving given quantum objectives in the shortest time with the most realistic circuit configuration.
    - AKA open-loop coherent control
  - Lyapunov-based controller design: control input is determined by the quantum states.
    - (seems like closed loop control?)

Control-Lyapunov Functions

- Lyapunov functions are used to determine whether a dynamical system is asymptotically *stable.*
- In the quantum case and in the textbook's citations, Lyapunov functions can be used to determine whether a system with control inputs is asymptotically *controllable.*
  - Controllable -> stability with continuous feedback (not open-loop as our textbook suggests)

Any Questions on this part?

- My answer:

Closed-Loop Control

- Closed loop control for quantum systems has been approached in two ways:
  - Learning-based control
    - Optimization problems
    - Learning algorithms to find optimal parameters for desired performance
      - Gradient Algorithms (e.g. gradient descent)
        - Non-convex/concave optimization with many local maxima/minima (optima)
        - Numerical methods take too long
      - Stochastic Algorithms
        - Genetic Algorithm
        - Differential Evolution
  - Feedback-based control
    - Quantum-applied feedback controllers

Quantum Optimal Control

- What are we optimizing?
  - Control Pulses
- How are we optimizing?
  - Textbook explains one numerical method:
    - Gradient Ascent Pulse Engineering (GRAPE)
- What are the inputs and outputs of our optimizer?
  - Quantum instructions
  - Optimized control pulses

Quantum Optimal Control (cont)

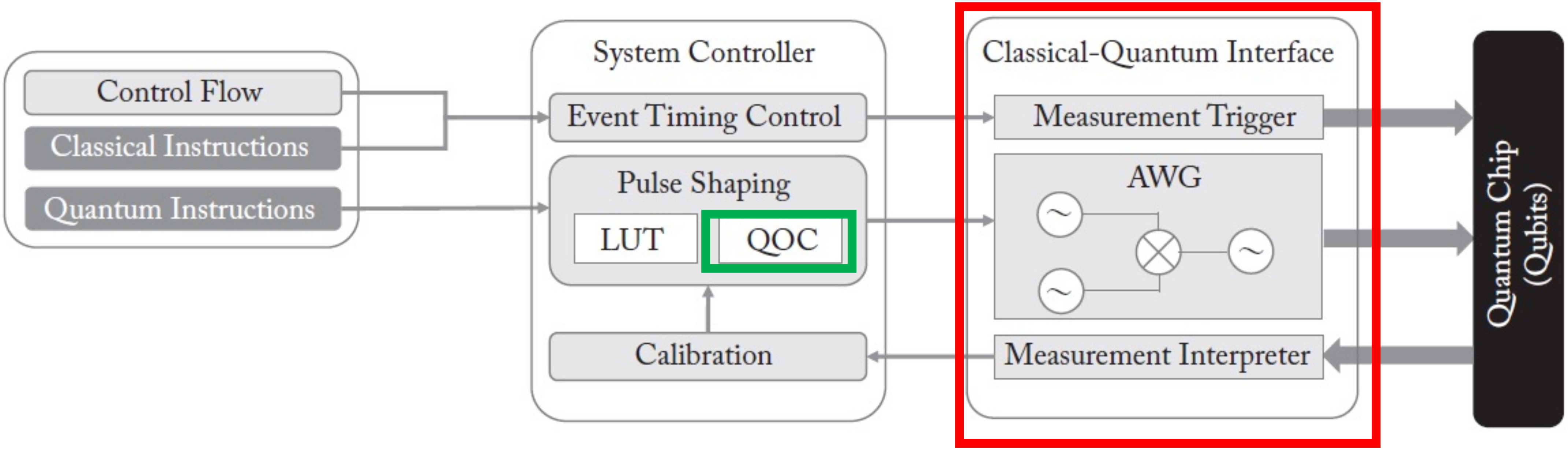


Figure 7.1: General flow for translating quantum gates to analog pulses.

GRAPE

- System Hamiltonian (overall system energy as a function of time)

$$\mathcal{H}(t) = \mathcal{H}_r + \sum_{i=1}^{m} \mathcal{H}_i(t),$$

- Time dependent Hamiltonian operators $\mathcal{H}_i(t) = u_i(t)\mathcal{H}_i).$

GRAPE (cont)

- H(t) approximates the target Unitary .
- So what is the purpose of Quantum Optimal Control?
  - To find the control fields (inputs) so that we can accurately approximate H(t) !
- How does the system Hamiltonian help approximate the Unitary?
  - First, discretize the system.
  - Then, perform ‘piecewise-constant approximation to get the time-independent Hamiltonian

GRAPE (cont)

- What does this look like?
  - Evolve a Quantum system from time 0 to time j… $t = t_0 + j\delta t.$
  - Then, for each timestep *j* we can determine a set of *constant* control fields where m is the number of control fields.
  - Then we can approximate the time-independent Hamiltonian with those control inputs:

$$H_j = \mathcal{H}_0 + \sum_{i=1}^{m} u_{i,j}\mathcal{H}_m.$$

  - And the Unitary operation at each timestep *j* can be realized by:

$$U_j = e^{-iH_j\delta t}.$$

GRAPE (cont)

- Now that we have the Unitary operation at each time step on the interval [t0, j], we can apply the ‘piecewise-constant approximation’ method to get the target Unitary matrix:

- So where is the GR in GRAPE?
  - GRAPE performs gradient descent search over the space of possible control field parameters that approximate the targeted Unitary within specified error constraints.

Mathematical Wizardry of GRAPE

- If we know the input state of the computation to be performed…. GRAPE can optimize a control pulse that works for the particular input state. The book claims that we can approximate a U that does not resemble the true unitary matrix. However, we still achieve, with a considerable degree of fidelity:

$$|\psi_{out}\rangle = \hat{U}\,|\psi_{in}\rangle \approx U\,|\psi_{in}\rangle\,.$$

ESCAL
Extreme Storage & Computer Architecture Lab

UC RIVERSIDE

Constraints of GRAPE Parameter Estimation

- The amplitude of each control field (u) must be minimized or the set of control fields must be normalized so that we can use it in practice.
  - We can't make a light beam with infinite intensity…
- Each control parameter needs to form a smooth pulse over time
  - i.e. we can minimize the difference between one control parameter at time j and time j + 1 $\sum_{i,j} |u_{i,j} - u_{i,j+1}|$
- Pulse time needs to be limited
  - Long pulses = long programs = much higher risk of quantum decoherence.

Was this Quantum Optimal Control?

- Sort of… This was a specific method of quantum optimal control using a learning-based method.
- The example is really limited, and does not work with more than 4 qubits because of the amount of time it takes to calculate the those optimal control parameters.
- There are several other methods, some more robust and 'better' than this one.
    - Daoyi Dong – Differential Evolution (msMS-DE) proposed in "Learning Based Quantum Robust Control: Algorithm, Applications, and Experiments" published in 2020.
- Scalability of quantum optimization is terrible these days, we can only do optimized pulse shaping for a small number of qubits – future work, perhaps?

Compilation For Variational Algorithms

- How are large quantum programs compiled?
  - Take a small set of quantum gates, synthesize a quantum program (generate quantum instructions) using those gates, apply quantum optimal control to each gate to generate the *shaped* pulses, and then concatenate all pulses to accomplish the computation.
    - SUPER LONG COMPILE TIMES
  - Use a lookup table for simple quantum gates! → short and constant compile times
    - NOT OPTIMAL (long pulses)
- What should be done?
  - Partial compilation.

Partial Compilation

- Use a hybrid approach:
  - Use lookup tables AND quantum optimal control methods
  - I guess the best compromises are when neither party gets what they want.
    - Slow compile times AND non-optimal pulses ☹
    - Or *faster* compile times and *more* optimal pulses
- Works for ‘variational’ quantum algorithms – each iteration of the algorithm only changes slightly. So we use QOC to optimize/recompile the changes to the gates. It’s less expensive than recompiling the entire circuit

Summary

- Generating pulses from quantum programs
- Math Magic
- Various optimizations for pulse shaping
- GRAPE
- Pulse Compilation

Any Questions?

ESCAL
Extreme Storage & Computer Architecture Lab

UC RIVERSIDE