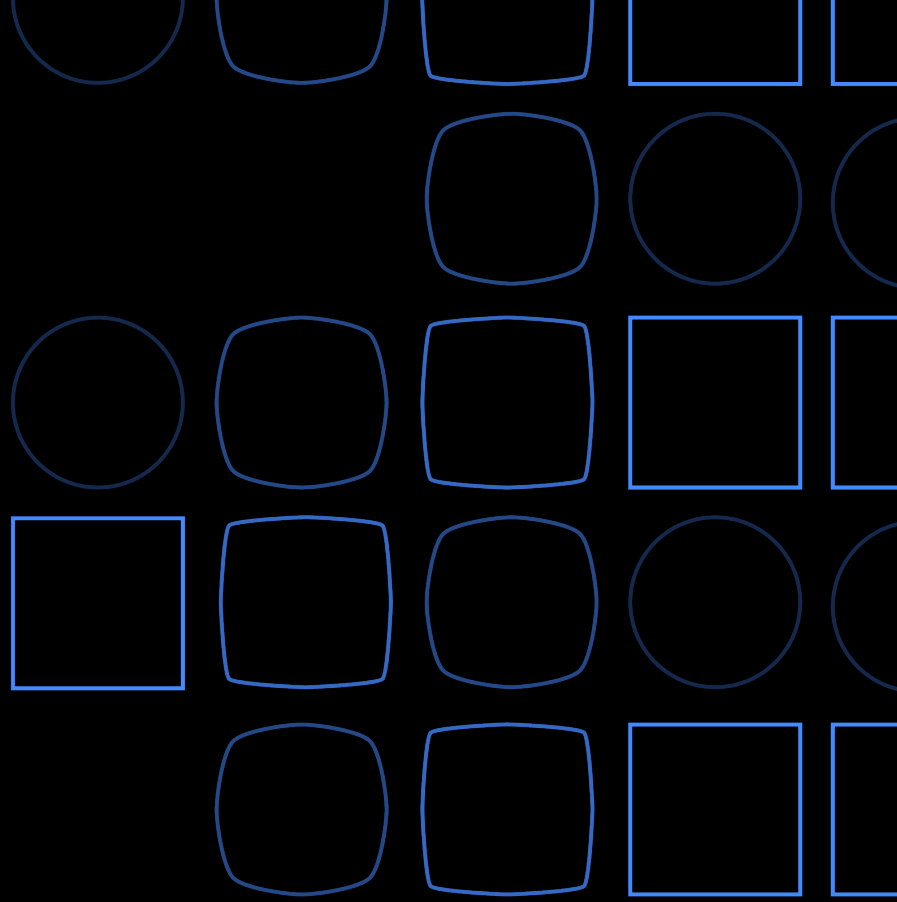
Quantum Optimal Control: Using GRAPE to Generate Optimal Pulses

Ben Rosand
IBM Quantum Intern

with
Thomas Alexander
Zachary Schoenfeld



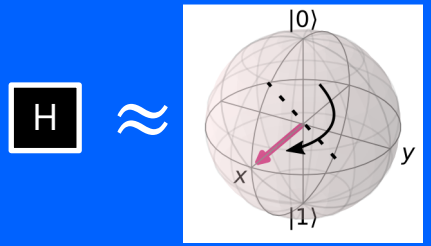
Quantum Optimal Control

Quantum Optimal Control is the process of engineering a unitary evolution as close to a target unitary in a system. Specifically we use pulse controls to evolve the unitary of a transmon.

Problem

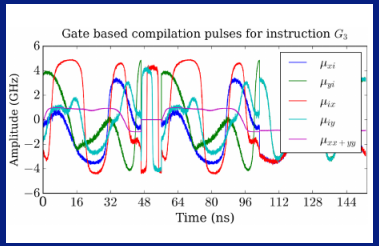
- Pulses are calibrated for a specific set of basis
- Calibrations optimize fidelities of basis gates, not circuit processes.
- Arbitrary unitaries must be compiled to basis gates

Less accurate circuits



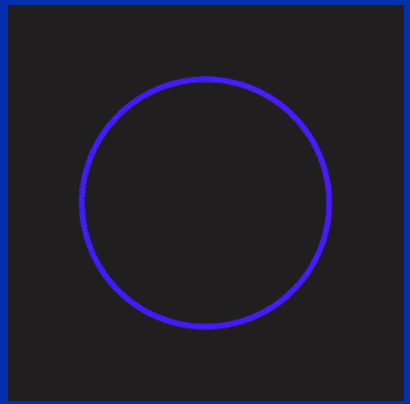
[1]

Inefficient pulse sequences



[3]

Slower and potentially less accurate qc programs



[2]

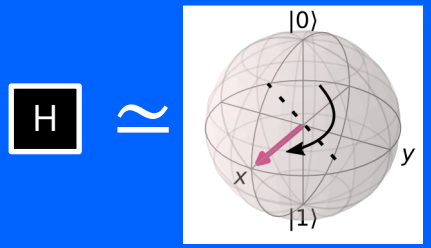
Solution

GRAPE (or other OCT)

Gradient ascent pulse engineering

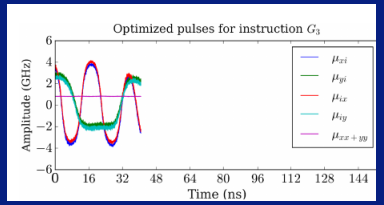
Imagine if we could optimize a drug for a given patient's genome as we inject the drug.

Unitary evolution- optimized pulse sequences



[1]

Gate aggregation



[3]

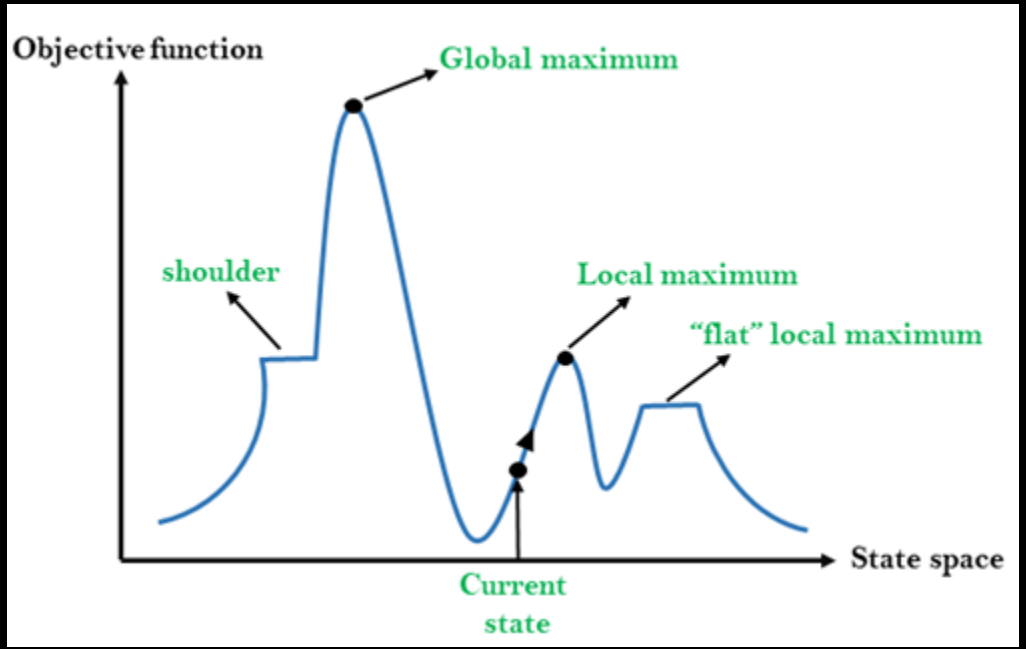
OPTIMIZED Quantum Programs

[4]

Gradient Ascent Pulse Engineering (GRAPE)

1. Initialize pulse guesses
2. Calculate density operator fidelity with given sequence
3. Move in direction of greatest increase in fidelity

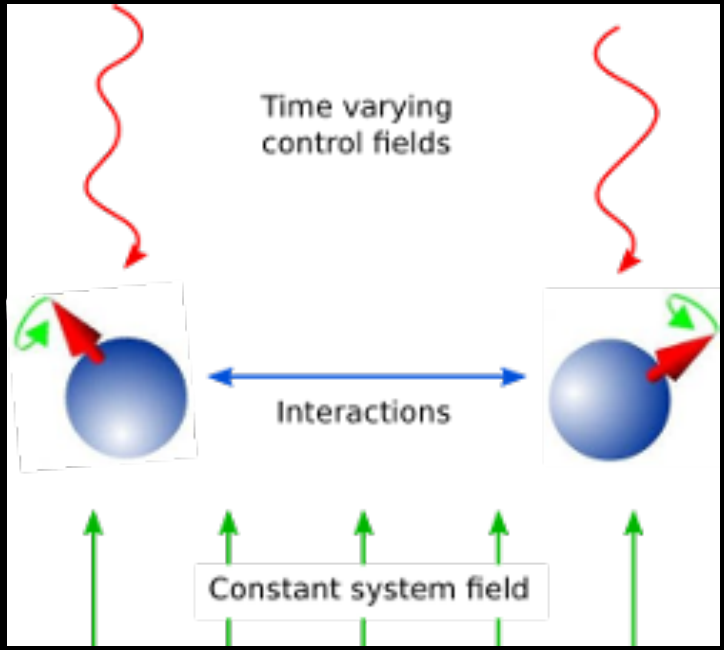
- $H(t) \approx H(t_k) = H_0 + \sum_{j=1}^{N} u_{jk} H_j$
- $f_{PSU} = \frac{1}{d}\left|tr\{X_{targ}^{\dagger} X(T)\}\right|$



[5]

QuTiP Grape

- Models Unitary evolution
- Control field parameters + system field
- Goal: maximize fidelity f_{PSU}
- $f_{PSU} = \frac{1}{d}\left|tr\{X_{targ}^{\dagger} X(T)\}\right|$

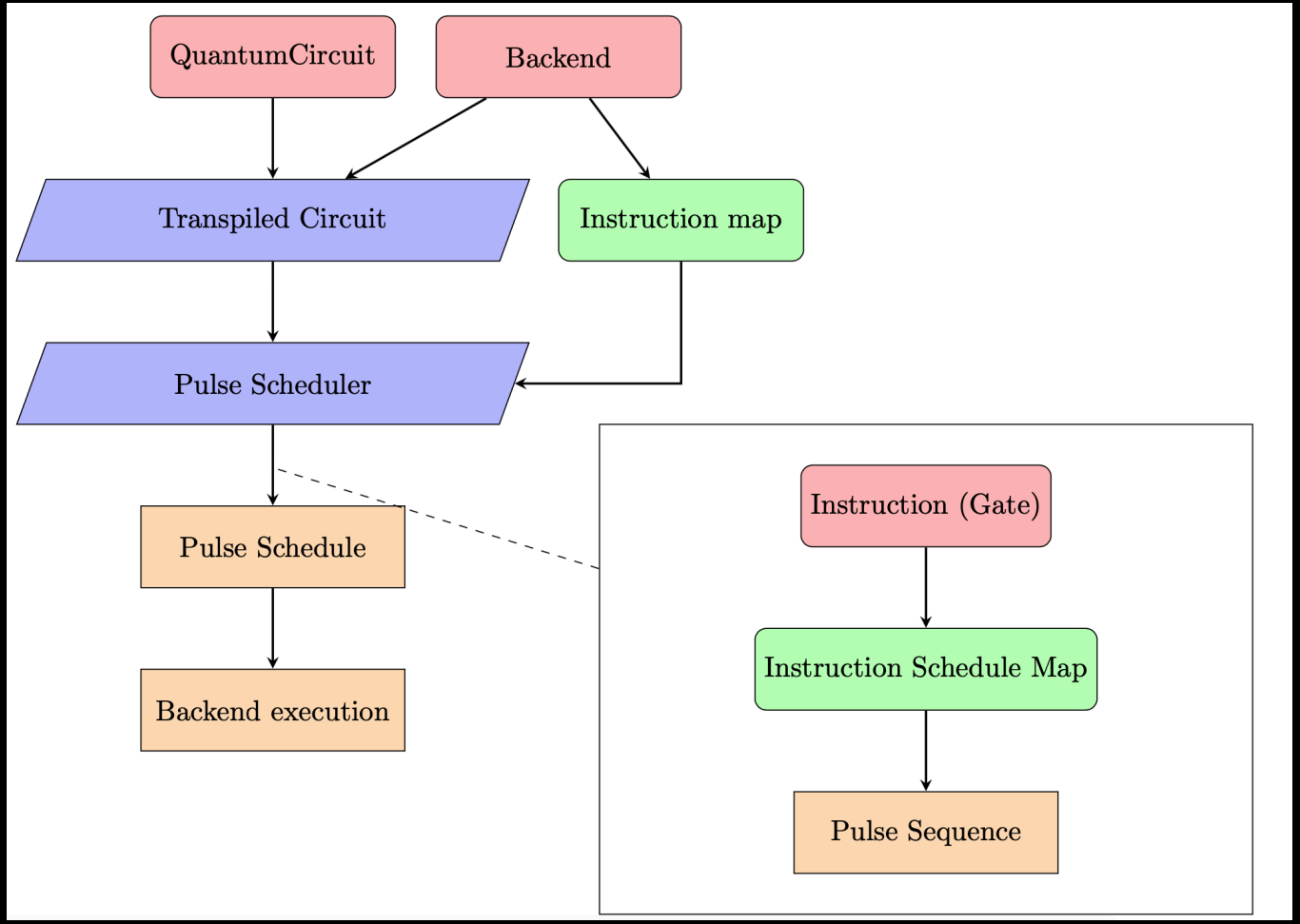


[7]

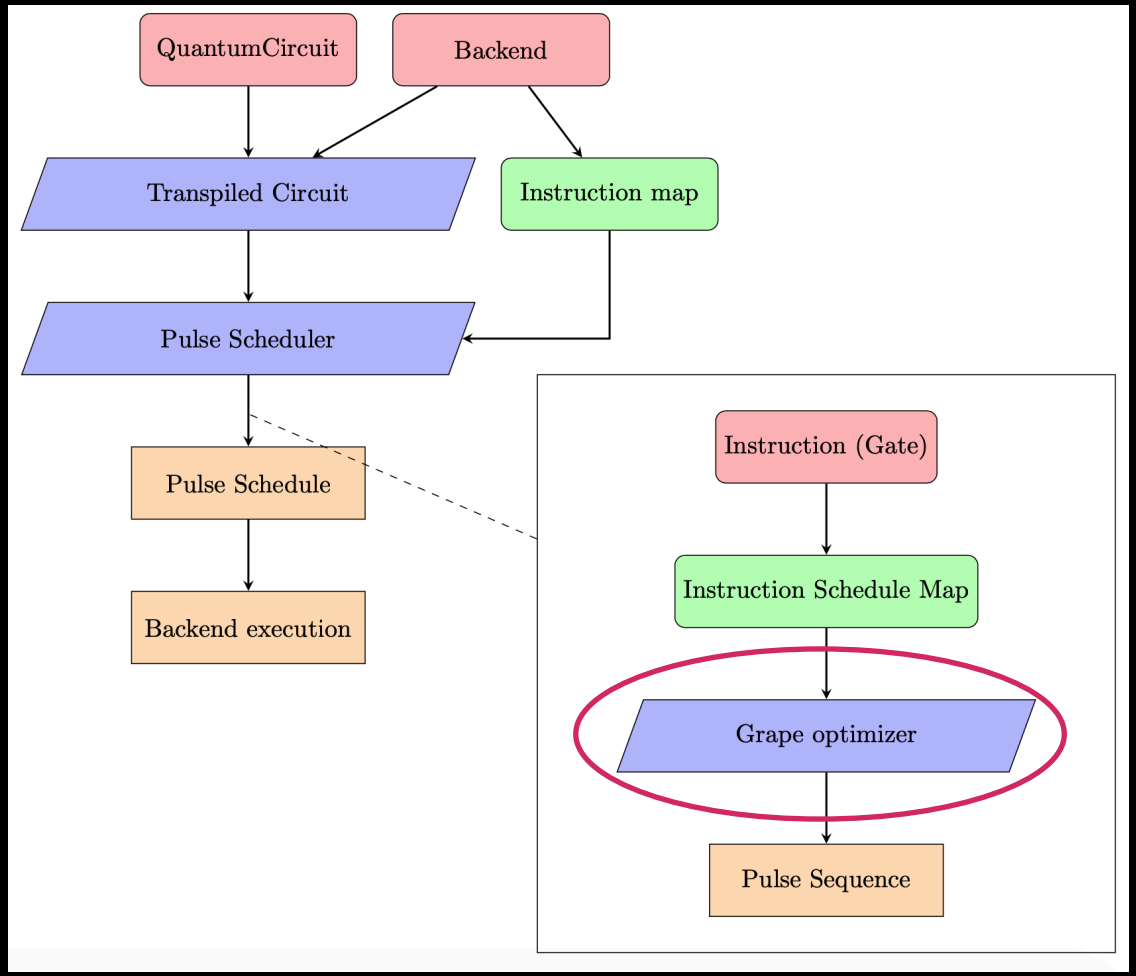
Demo

http://localhost:8892/notebooks/systems_demo_aug3.ipynb

Current Circuit Pipeline



QOC Circuit Pipeline



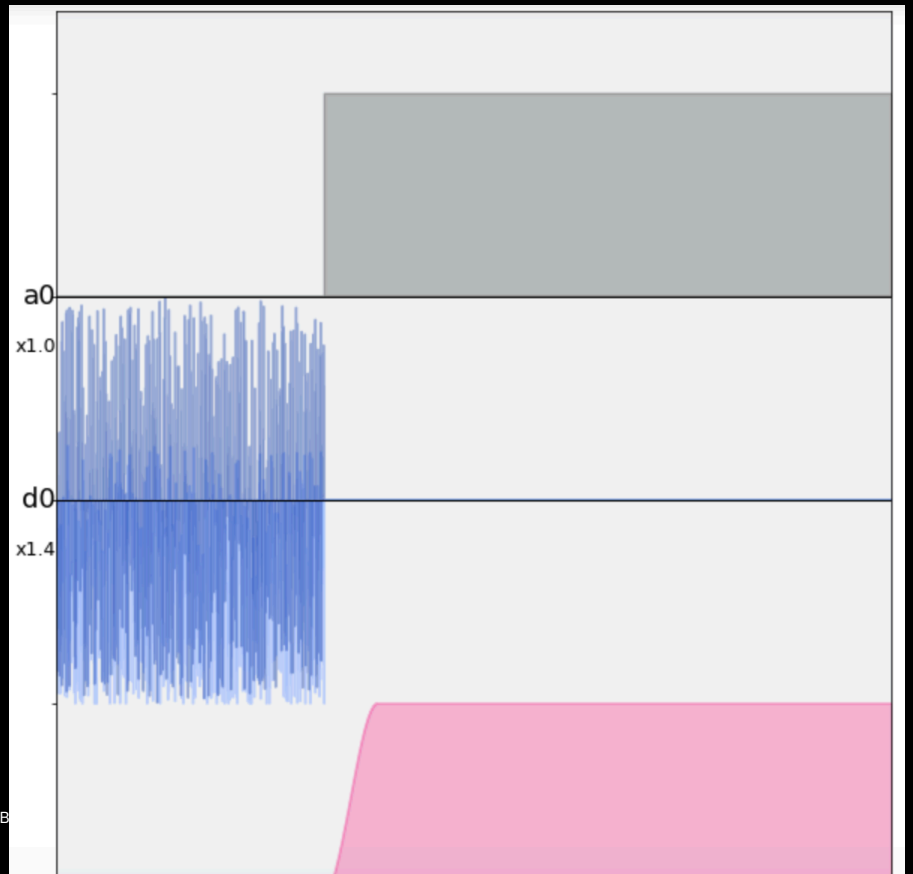
Prerun demo

http://localhost:8892/notebooks/prerun_demo_aug3.ipynb

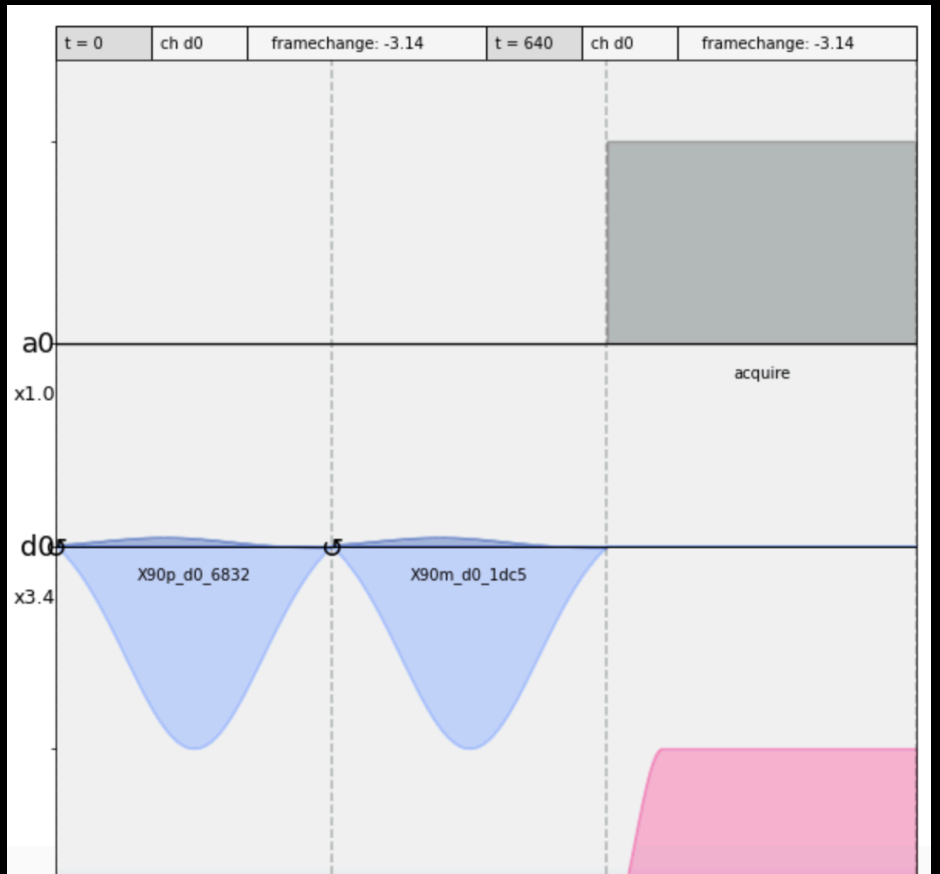
Running demo

http://localhost:8892/notebooks/systems_demo_aug3.ipynb

Grape pulse – 71ns

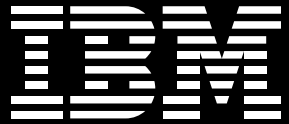


Default U3 pulse – 284ns



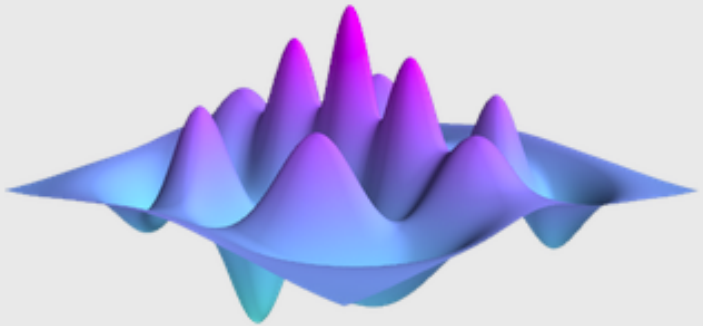
Next Steps

- Package and submit PR for single qubit QOC
- Add in gate aggregation
- Extend to n-qubit systems
- Extend to 2q gates
- Automate gate time selection
- Thoroughly study accuracy and gate time



Following slides are extras

Qiskit – QuTiP – Qiskit

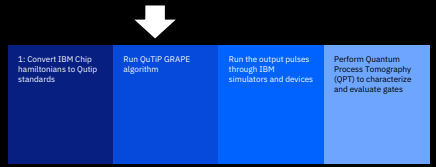


QuTiP

Quantum Toolbox in Python

[6]

- IBM Hamiltonian format differs from Qutip
- Conversion and unit conversion necessary
- Extensive Qutip API investigation
- Ultimately one of the biggest unexpected challenges of this project



Real device testing

GRAPE π pulse: **92.7% -- latest test (same time)**

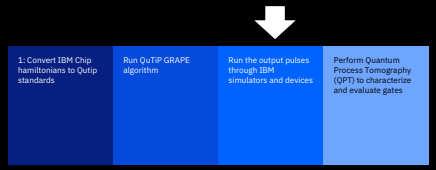
Default π pulse: **92.6%**

4096 shots

Is this bad – No

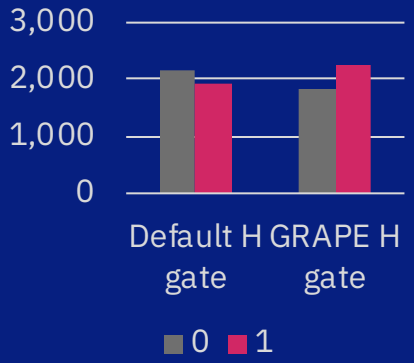
GRAPE Hamiltonian:

$$H = \frac{\omega(1-\sigma_z)}{2} + \Omega_{d0}\sigma_x$$



X gates

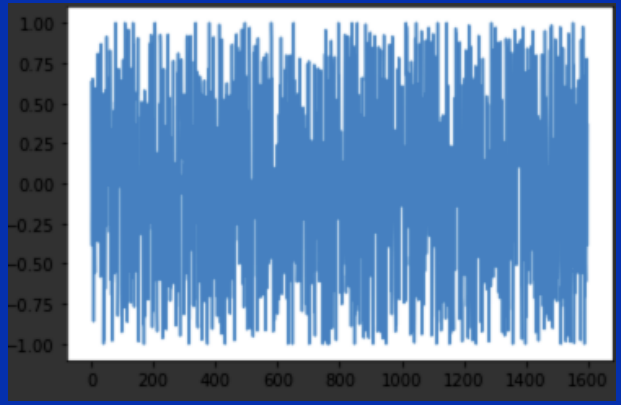
H gates



GRAPE

π Pulse

Schedule



Quantum Process Tomography

- Test response on every permutation of start basis and measurement basis

process_tomography_circuits

process_tomography_circuits(*circuit, measured_qubits, prepared_qubits=None, meas_labels='Pauli', meas_basis='Pauli', prep_labels='Pauli', prep_basis='Pauli'*) [source]

Return a list of quantum process tomography circuits.

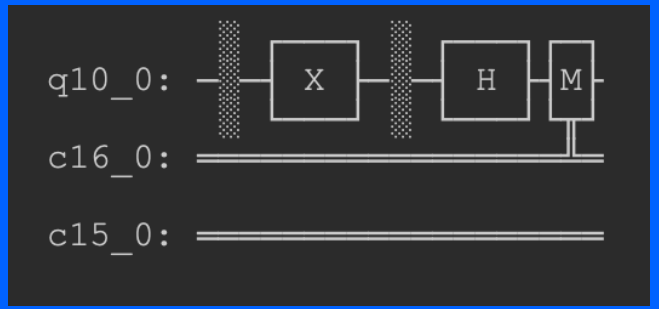
This performs preparation in the minimial Pauli-basis eigenstates

- `"Z_p"` : $|0\rangle$
- `"Z_m"` : $|1\rangle$
- `"X_p"` : $|+\rangle$
- `"Y_m"` : $|+i\rangle$

on each qubit, and measurement in the Pauli-basis X, Y, Z resulting in $4^n 3^n$ circuits for an n-qubit process tomography experiment.

[8]

Start $|0>$ apply X gate and measure in X basis

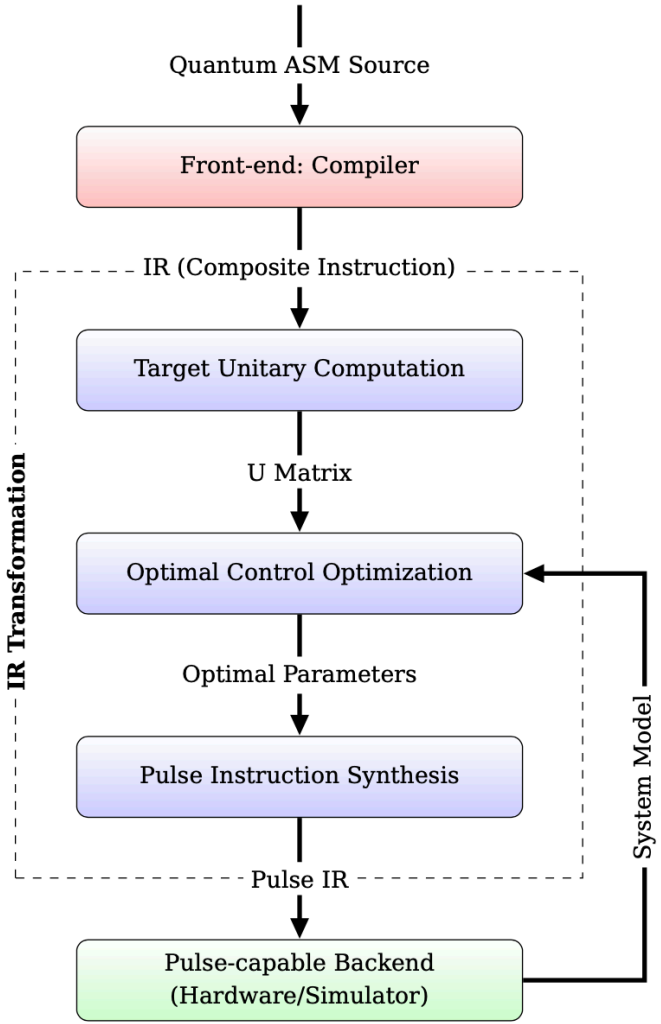


GRAPE X gate fidelity (missing process see git)

```
Least-Sq Fitter
my fit fidelity (state): 0.9548905798483875
default fit fidelity (state): 0.9674396321373899
ideal fit fidelity (state): 0.9925334990120357
my fit time: 0.019559383392333984
```

XACC QOC pipeline [9]

- XACC example pipeline



Qiskit Pulse Pipeline

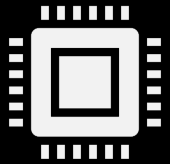
Quantum Logic Gates

Gates transpiled to basis gates

Basis gates compiled to pulses

Pulses run on hardware

Qiskit Pulse Pipeline



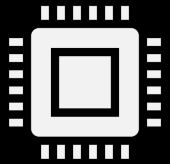
Quantum Logic Gates

Gates transpiled to basis gates

Basis gates compiled to pulses

Pulses run on hardware

Potential Qiskit Pulse Pipeline



Quantum Logic Gates

Gates transpiled to basis gates

Basis gates compiled to pulses, with some gates compiled to GRAPE pulses

Pulses run on hardware

Citations

- [1] Loading gif: https://gifer.com/en/LE57
- [2] Bloch Sphere: https://qiskit.org/textbook/ch-states/single-qubit-gates.html
- [3] Pulse graph: https://arxiv.org/pdf/1902.01474.pdf
- [4] Fast speedtest: https://www.youtube.com/watch?v=xt0EOY0GcuI
- [5] Hill climbing: https://www.geeksforgeeks.org/introduction-hill-climbing-artificial-intelligence/
- [6] Qutip banner: qutip.org
- [7] Grape image: http://qutip.org/docs/latest/guide/guide-control.html
- [8] Qiskit docs: https://qiskit.org/documentation/stubs/qiskit.ignis.verification.process_tomography_circuits.html
- [9] XACC pipeline: https://arxiv.org/pdf/2006.02837.pdf

Shi gate aggregation speedup [3]