Quantum Elements

QEsim

A Quantum Emulator

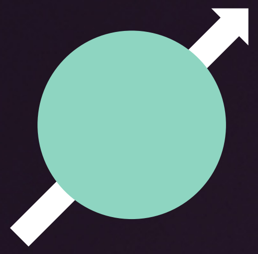
How to build open system models for current quantum devices?

- Quantum Computers are Noisy
- To fully understand and emulate the performance of such systems a noise model needs to be developed and solved
- An Open Quantum System allows one to emulate the real performance of a noisy system
- Our software solves the time evolution of the quantum system in the presence of its environment

QUANTUM ELEMENTS

Building Blocks

Building blocks: qubit



$|0\rangle \rightarrow \begin{pmatrix}1\\0\end{pmatrix}$

$|1\rangle \rightarrow \begin{pmatrix}0\\1\end{pmatrix}$

Computational basis state

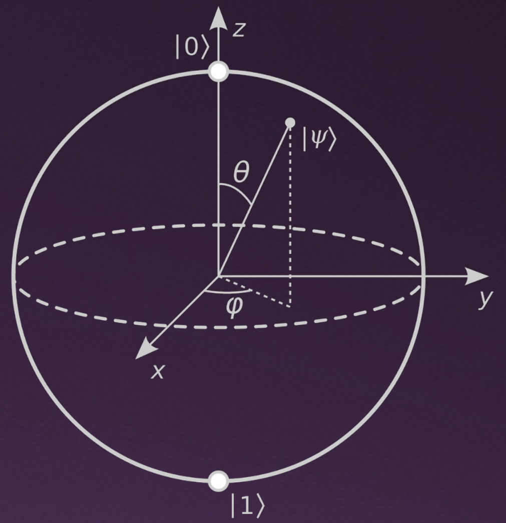
$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$$
$$= \cos(\theta/2)|0\rangle + e^{i\phi}\sin(\theta/2)|1\rangle$$

The dynamics of the qubit is determined by the Schrödinger Equation

$$|\dot{\psi}\rangle = -i\hat{H}|\psi\rangle$$

H is the Hamiltonian

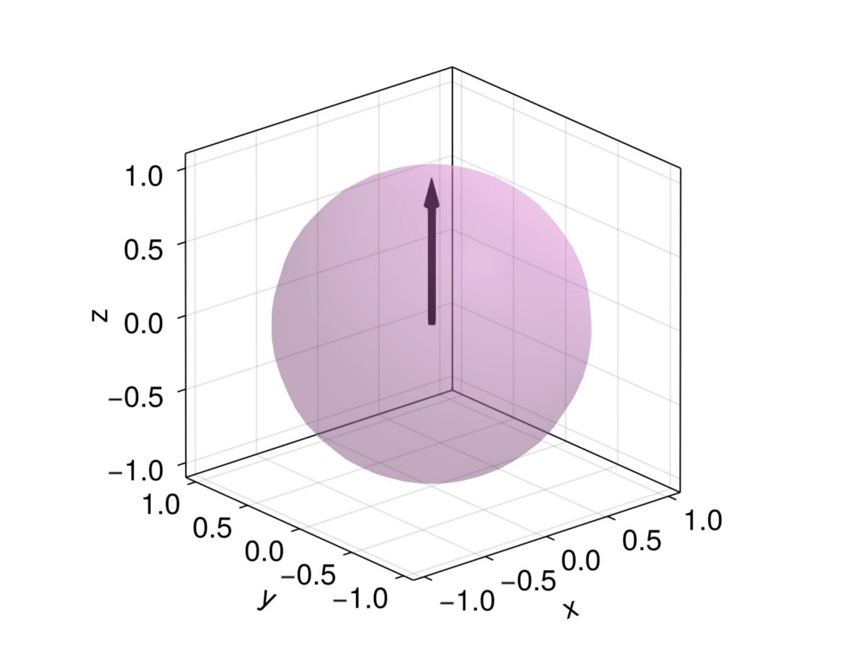
$|\psi\rangle$ can be represented by a point on the Bloch sphere

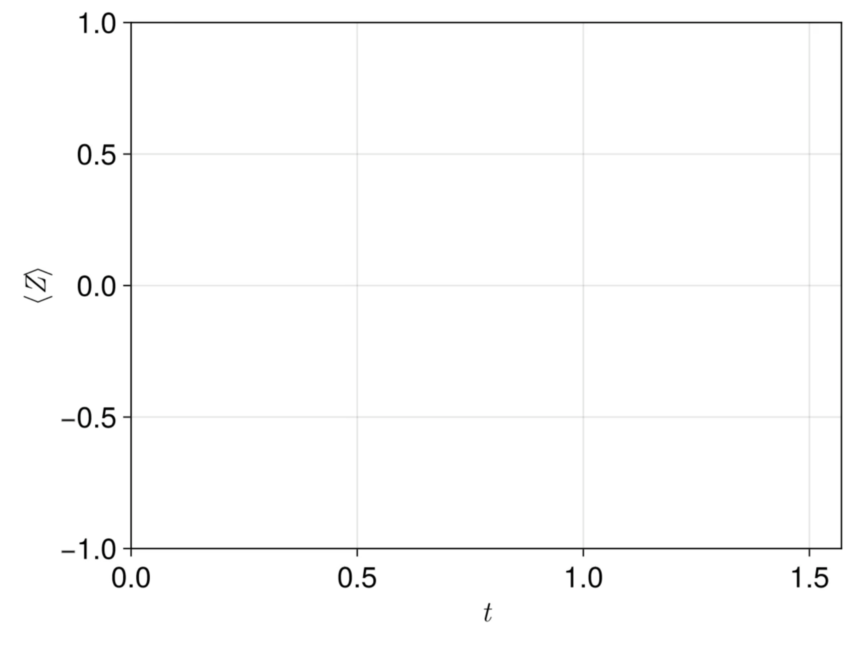


Building blocks: quantum gate

$|0\rangle$ —[X]— $|1\rangle$ $\quad |\dot{\psi}\rangle = -i\Omega\hat{X}|\psi\rangle$ $\quad$ Pauli X matrix: $\hat{X} = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$

$R_X(\pi)$: a 180° rotation around X axis





Building blocks: density matrix

Due to noise in quantum systems, we no longer have perfect knowledge of their states. Instead, we understand that the system could be in a specific quantum state $|\psi_i\rangle$ with a certain probability p_i.

Probability	State
p_1	$\|\psi_1\rangle$
p_2	$\|\psi_2\rangle$

$\vdots$

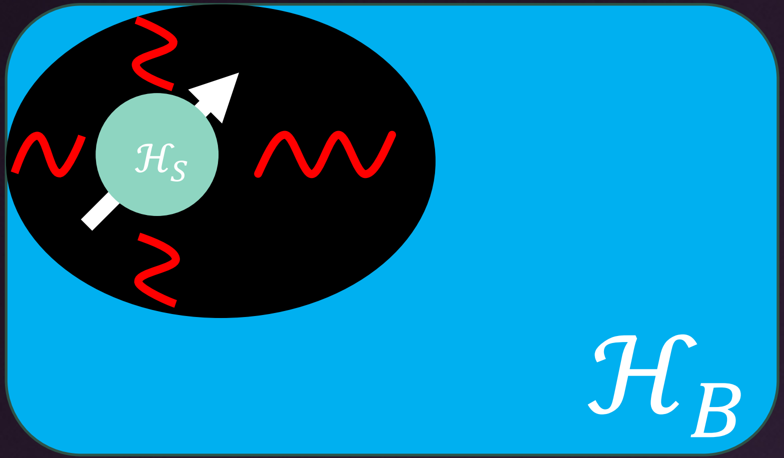
Combine together

Density matrix: $\hat{\rho} = \sum_i p_i |\psi_i\rangle\langle\psi_i|$

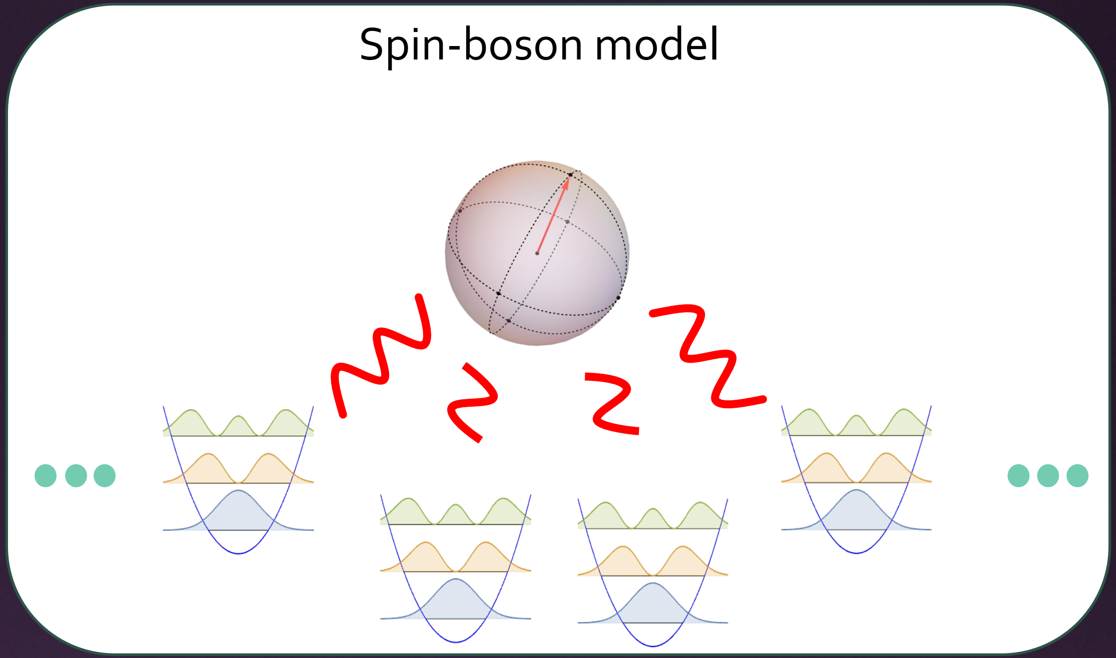
von Neumann equation: $\dot{\hat{\rho}} = -i[\hat{H}, \hat{\rho}]$

Theory of open quantum systems

The theory of open quantum systems studies the physical properties of a small quantum system in contact with a huge environment.



Example:



$$H_S \otimes I_B + \sum_{\alpha} g_\alpha A_\alpha \otimes B_\alpha + I_S \otimes H_B$$

[1] H.-P. Breuer and F. Petruccione, *The Theory of Open Quantum Systems* (Oxford University Press, 2002).
[2] U. Weiss, *Quantum Dissipative Systems*, Vol. 13 (World scientific, 2012).
[3] Lidar, Daniel A. "Lecture Notes on the Theory of Open Quantum Systems." arXiv.1902.00967, https://doi.org/10.48550/arXiv.1902.00967.

Building blocks: master equations

QUANTUM ELEMENTS

Incoherent quantum noise

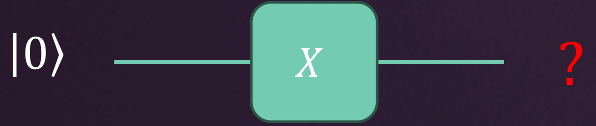
$$\dot{\hat{\rho}} = -i[\hat{H}, \hat{\rho}] + \hat{\mathcal{L}}[\hat{\rho}]$$

One of the most commonly used mater equation: Lindblad equation

$$\hat{\mathcal{L}}[\hat{\rho}] = \sum_i \gamma_i \left(\hat{L}_i \hat{\rho} \hat{L}_i^\dagger - \frac{1}{2} \{ \hat{L}_i^\dagger \hat{L}_i, \hat{\rho} \} \right)$$

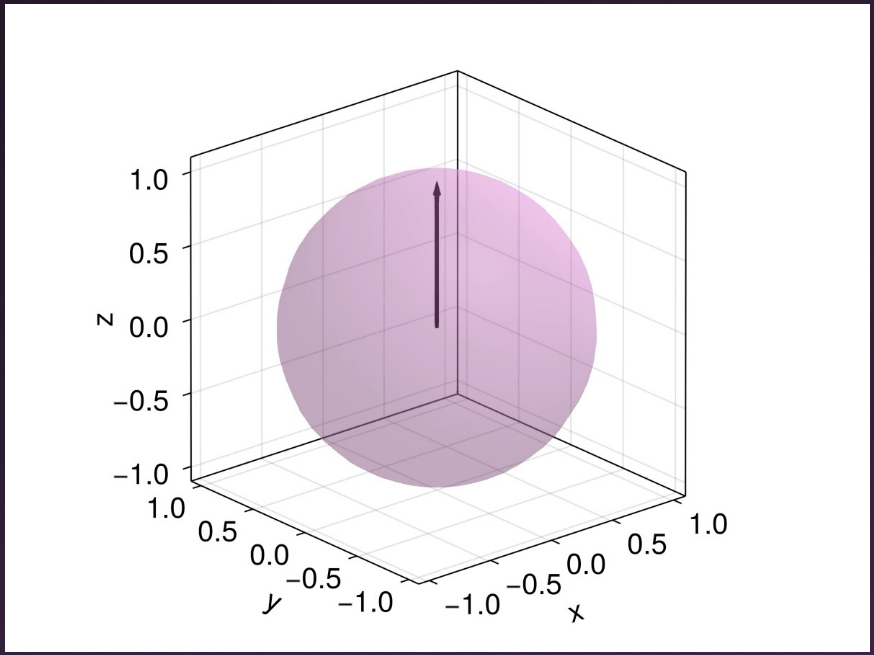
$\hat{L}_i$ are a set of jump operators describing the "noisy" part of the dynamics

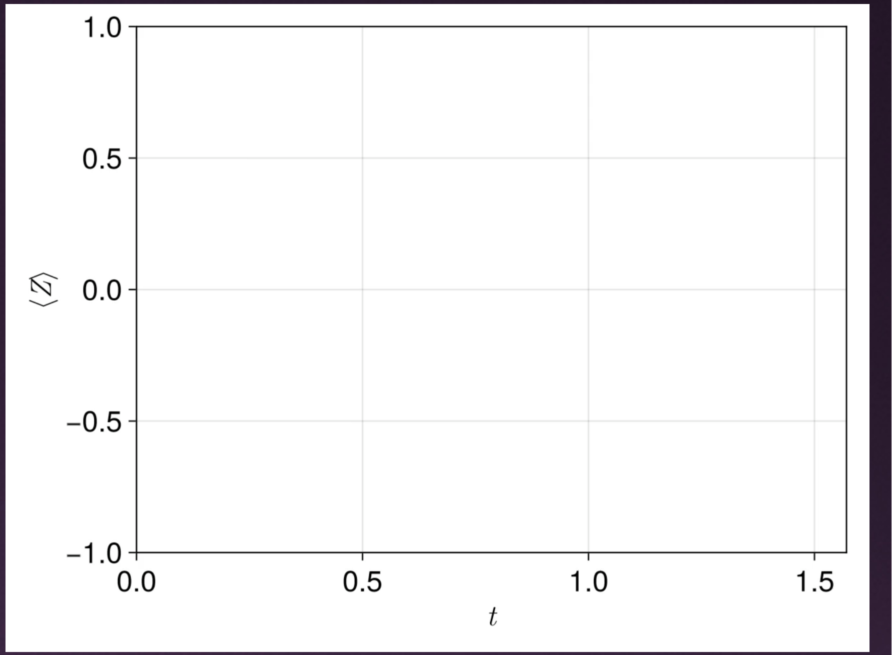
Noisy quantum gate



Consequence of noise:

- We are no longer able to directly flip the quantum state from |0⟩ to |1⟩
- The quantum state evolves inside the Bloch sphere





Building blocks: Redfield equation

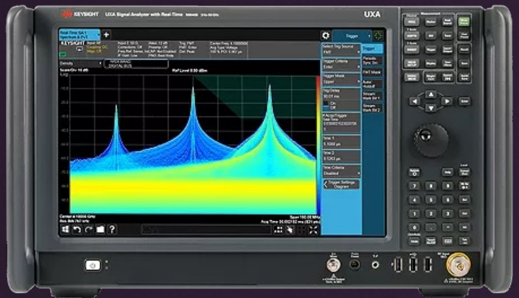
$$\dot{\hat{\rho}} = -i[\hat{H}, \hat{\rho}] - \sum_{\alpha} [\hat{A}_{\alpha}, \hat{\Lambda}_{\alpha}(t)\hat{\rho}] + h.c.$$

$$\hat{\Lambda}_{\alpha}(t) = \sum_{\beta} \int_0^t \boxed{C_{\alpha\beta}(t-\tau)} \hat{U}(t,\tau) \hat{A}_{\beta} \hat{U}^{\dagger}(t,\tau) d\tau$$

Bath (noise) correlation function

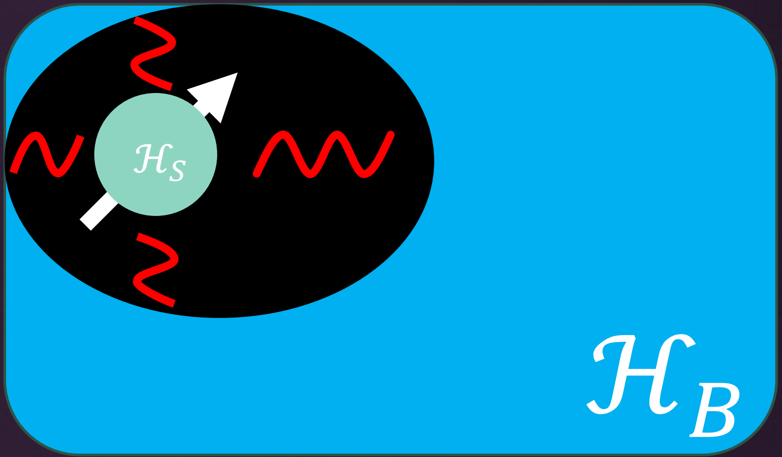
$$S_{\alpha\beta}(\omega) = \int C_{\alpha\beta}(\tau) e^{i\omega\tau} d\tau$$

Bath (noise) spectrum



picture from KEYSIGHT (https://www.keysight.com)

Analog of the power spectral density (PSD) of classical signals



$$H_S \otimes I_B + \sum_{\alpha} \boxed{g_{\alpha}} A_{\alpha} \otimes \boxed{B_{\alpha}} + \boxed{I_S \otimes H_B}$$

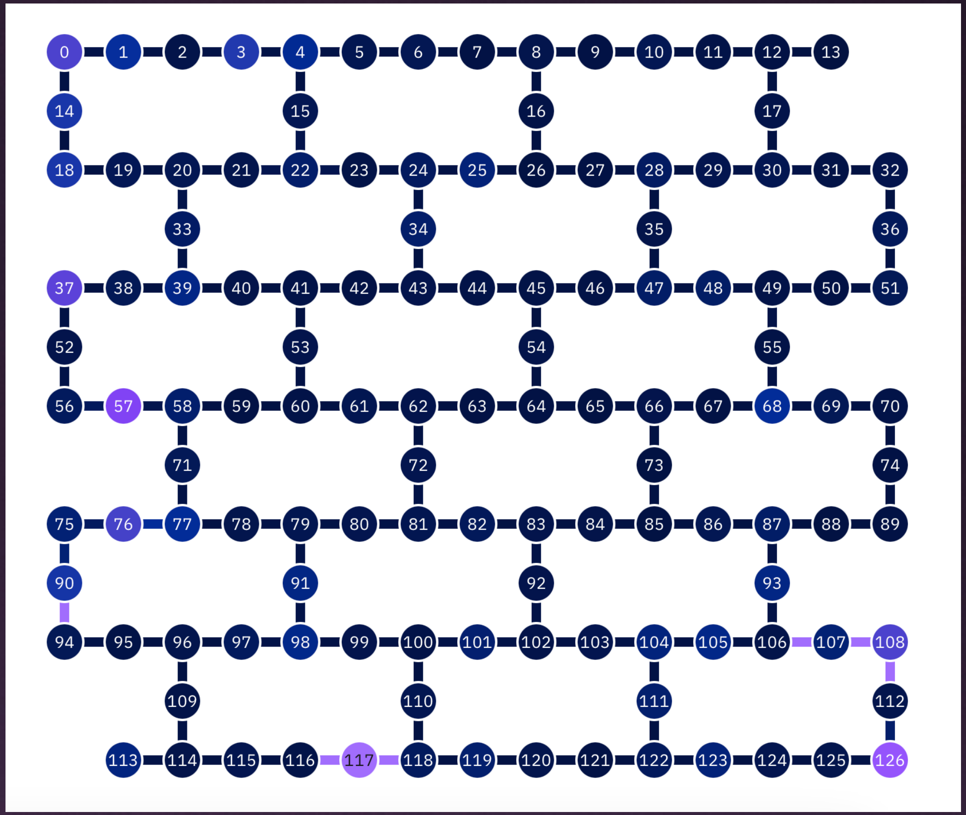
All the information about the environment are contained in the noise spectrum, which can be experimental measured.

Modeling IBM quantum computers

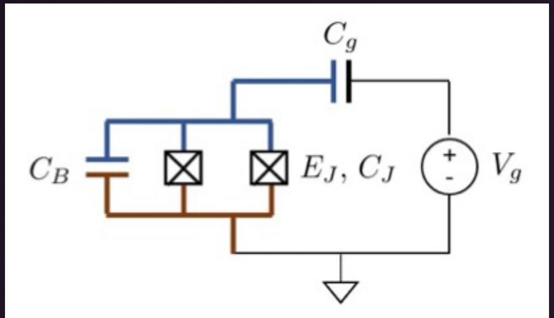
An introduction to the quantum gates used in the IBM quantum computers

QUANTUM ELEMENTS

127 Qubits	Status:	Online	Median ECR error:	9.034e-3
	Total pending jobs:	431 jobs	Median SX error:	2.675e-4
3.6% EPLG	Processor type ⓘ:	Eagle r3	Median readout error:	1.600e-2
	Version:	1.2.38	Median T1:	223.52 us
5K CLOPS	Basis gates:	ECR, ID, RZ, SX, X	Median T2:	105.32 us
	Your instance usage:	0 jobs		

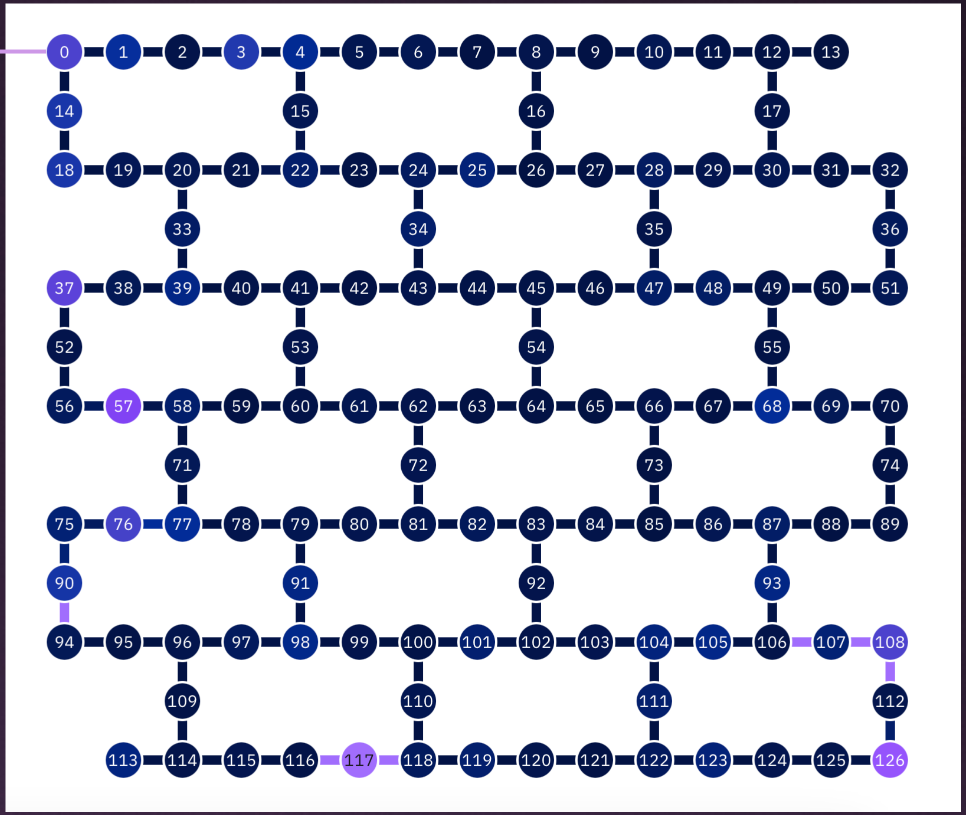


The Transmon



transmon qubit

IBM_kyoto connectivity graph



qubit anharmonicity

qubit frequency

number operator

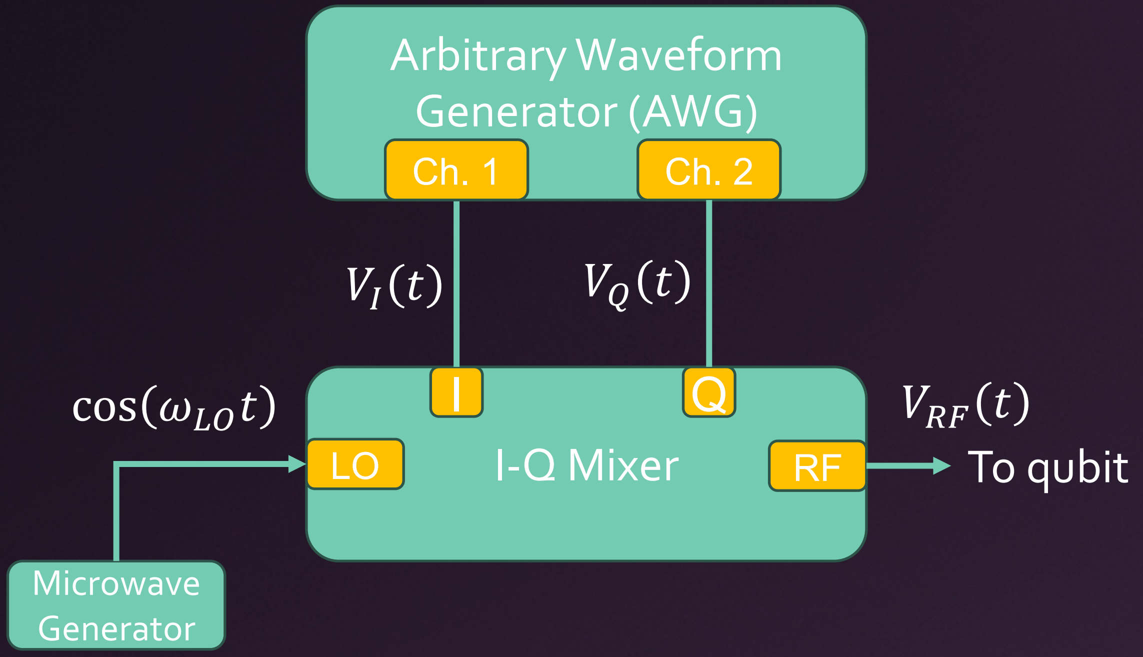
$$\hat{H} = \boxed{\Omega(t)\cos(\omega_D t - \gamma)\left(\hat{a} + \hat{a}^\dagger\right)} + \omega_{01}\hat{n} + \frac{\alpha}{2}(\hat{n} - 1)\hat{n}$$

single qubit gate pulse shape

single qubit driving frequency

This part is what we control.

Single qubit gate pulse



- LO: local oscillator
- RF: radio frequency

Similar to single-sideband modulation used in AM radio.

$$V_I(t) = \Omega(t)\cos(\omega_{SSB} t - \gamma)$$

$$V_Q(t) = -\Omega(t)\sin(\omega_{SSB} t - \gamma)$$

ω_{SSB} is an intermediate frequency used to generate high frequency modulation

$\Omega(t)$ and γ are pulse parameters we can control using AWG.

$$\begin{aligned} V_{RF}(t) &= V_I(t)\cos(\omega_{LO} t) + V_Q(t)\sin(\omega_{LO} t) \\ &= \Omega(t)\cos(\omega_D t - \gamma) \end{aligned}$$

$V_{RF}(t)$ is the control single going into the qubit Hamiltonian

$\omega_D = \omega_{LO} + \omega_{SSB}$

Single qubit gate Hamiltonian

An approximated Hamiltonian for a single transmon qubit under driving

$$\hat{H} = \Omega(t)\cos(\omega_D t - \gamma)\left(\hat{a} + \hat{a}^\dagger\right) + \omega_{01}\hat{n} + \frac{\alpha}{2}(\hat{n} - 1)\hat{n}$$

truncate to 2 levels

$$\hat{H} = \Omega(t)\cos(\omega_D t - \gamma)\,\hat{X} - \frac{\omega_{01}}{2}\hat{Z}$$

rotating frame of $\hat{U}_D = e^{-i\frac{\omega_D t}{2}\hat{Z}}$

$$\hat{\hat{H}} \approx -\frac{\Delta}{2}\hat{Z} + \frac{\Omega(t)}{2}\left[\cos(\gamma)\,\hat{X} + \sin(\gamma)\,\hat{Y}\right]$$

- $\hat{X}, \hat{Y}, \hat{Z}$ are single-qubit Pauli operators
- Δ is the detuning of qubit frequency and driving frequency $\Delta = \omega_{01} - \omega_D$

Assume $\Delta = 0$, we can implement two gates:

$\gamma = 0$ $\quad R_X(\theta) = e^{-i\frac{\theta}{2}\hat{X}}$

$\gamma = \frac{\pi}{2}$ $\quad R_Y(\theta) = e^{-i\frac{\theta}{2}\hat{Y}}$

$\theta = \int \Omega(\tau)\tau$

Tripathi, Vinay, Huo Chen, Eli Levenson-Falk, and Daniel A. Lidar. "Modeling Low- and High-Frequency Noise in Transmon Qubits with Resource-Efficient Measurement." *PRX Quantum* 5, no. 1 (February 7, 2024): 010320. https://doi.org/10.1103/PRXQuantum.5.010320.

127
Qubits

3.6%
EPLG

5K
CLOPS

Status:	Online
Total pending jobs:	431 jobs
Processor type ⓘ:	Eagle r3
Version:	1.2.38
Basis gates:	ECR, ID, RZ, SX, X
Your instance usage:	0 jobs

Median ECR error:	9.034e-3
Median SX error:	2.675e-4
Median readout error:	1.600e-2
Median T1:	223.52 us
Median T2:	105.32 us

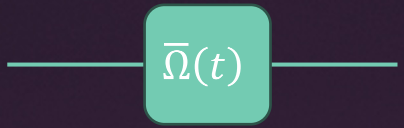
$R_X\left(\frac{\pi}{2}\right)$ $\quad$ $R_X(\pi)$

Complex pulse amplitude

A common practice is to rewrite $\Omega(t)$ as a complex function $\bar{\Omega}(t) = \Omega(t)e^{-i\gamma}$ according to

$$\frac{\Omega(t)}{2}\left[\cos(\gamma)\,\hat{X} + \sin(\gamma)\,\hat{Y}\right] \rightarrow \frac{\bar{\Omega}(t)}{2}|0\rangle\langle 1| + h.c.$$

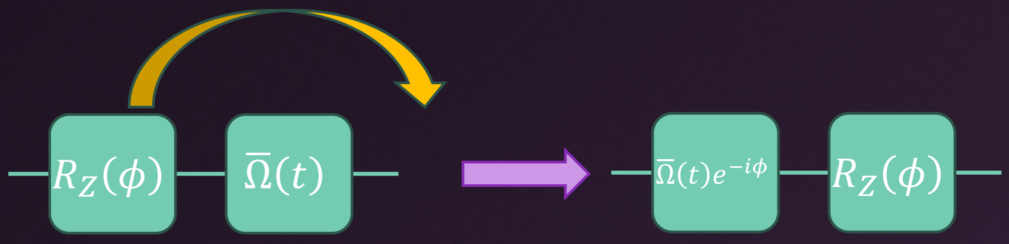
$\bar{\Omega}(t)$ contains all the pulse level information of a single qubit gate

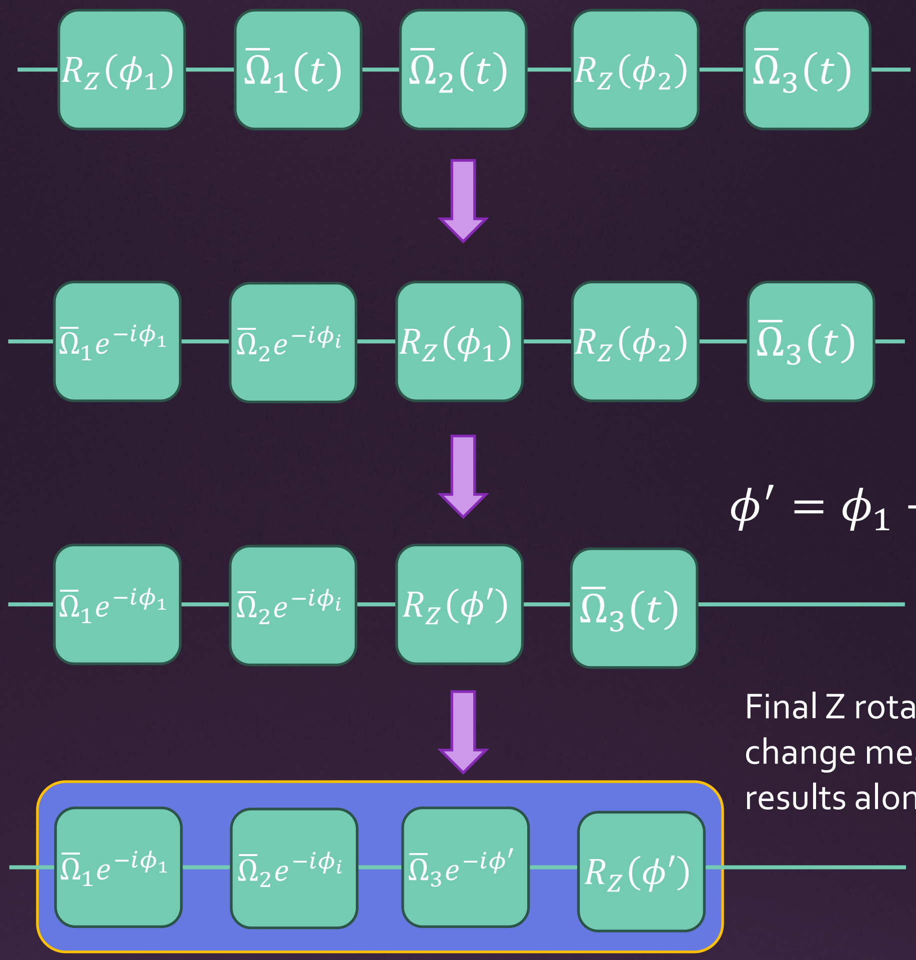


However, there is no controllable Z rotation in the above expression. How to implement a Z gate?

Virtual Z gate

Use the following commutation relation to move Z gate to the end of the circuit





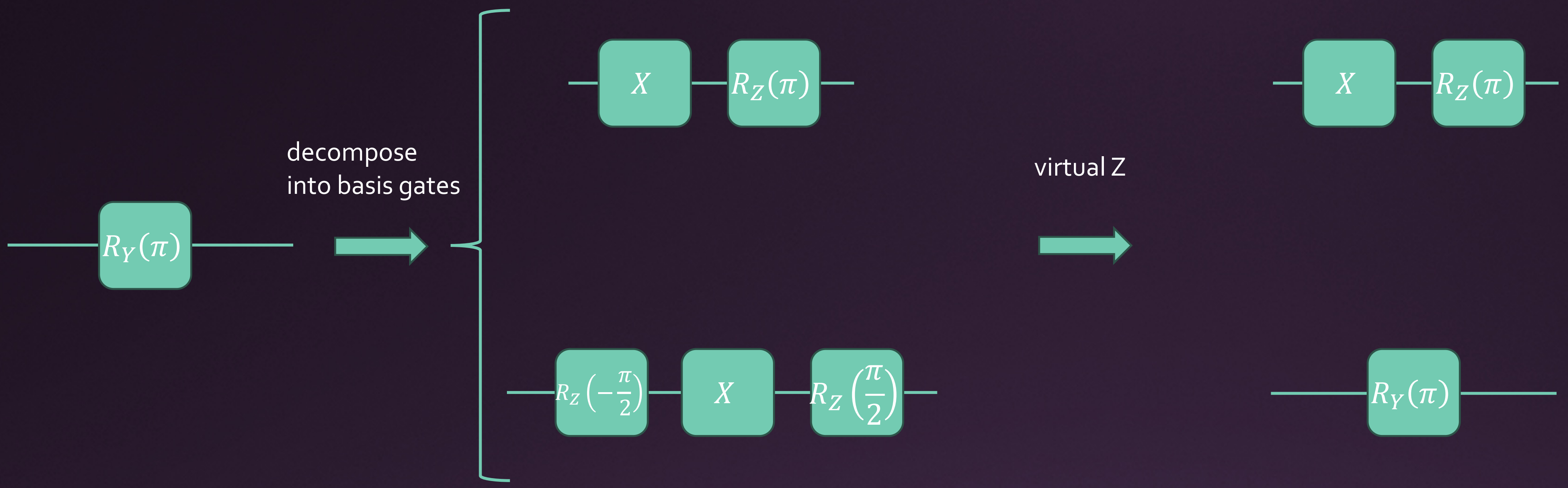
$$\phi' = \phi_1 + \phi_2$$

Final Z rotation does not change measurement results along Z direction

McKay, David C., Christopher J. Wood, Sarah Sheldon, Jerry M. Chow, and Jay M. Gambetta. "Efficient Z Gates for Quantum Computing." *Physical Review A* 96, no. 2 (August 31, 2017): 022330. https://doi.org/10.1103/PhysRevA.96.022330.

An example of R_Y gate

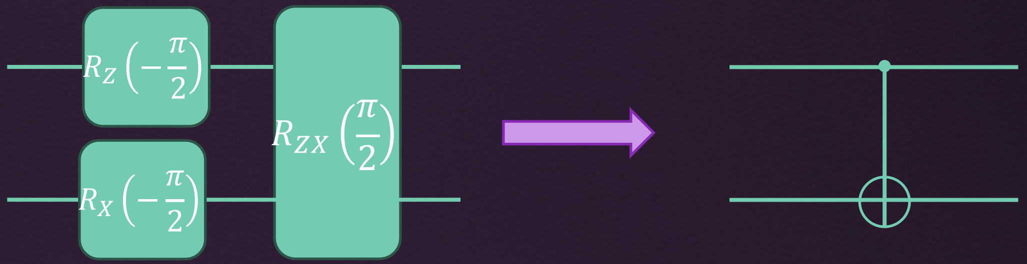
Implement a Y gate



In practice, the bottom approach is better.

Cross-resonance gate

Ideal cross-resonance gate: $\boxed{R_{ZX}(\theta) = e^{-\frac{i\theta \hat{Z}\hat{X}}{2}}}$



$$\hat{H} = \overbrace{\boxed{\Omega(t)\cos\left(\frac{\omega_2 t}{2} - \gamma\right)\left(\hat{a}_1 + \hat{a}_1^{\dagger}\right)}}^{\text{qubit 1 driving}} + \overbrace{\boxed{\omega_1 \hat{n}_1 + \frac{\alpha_1}{2}(\hat{n}_1 - 1)\hat{n}_1}}^{\text{qubit 1}} + \overbrace{\boxed{\omega_2 \hat{n}_2 + \frac{\alpha_2}{2}(\hat{n}_2 - 1)\hat{n}_2}}^{\text{qubit 2}} + \overbrace{\boxed{J\left(\hat{a}_1^{\dagger}\hat{a}_2 + \hat{a}_1\hat{a}_2^{\dagger}\right)}}^{\text{Interaction}}$$

Malekakhlagh, Moein, Easwar Magesan, and David C. McKay. "First-Principles Analysis of Cross-Resonance Gate Operation." *Physical Review A* 102, no. 4 (October 13, 2020): 042605. https://doi.org/10.1103/PhysRevA.102.042605.

QUANTUM ELEMENTS

Effective Hamiltonian using perturbation theory

In the rotation frame of $U(t) = e^{-i(\omega_1 \hat{n}_1 + \omega_2 \hat{n}_2)}$, the effective low-level Hamiltonian looks like

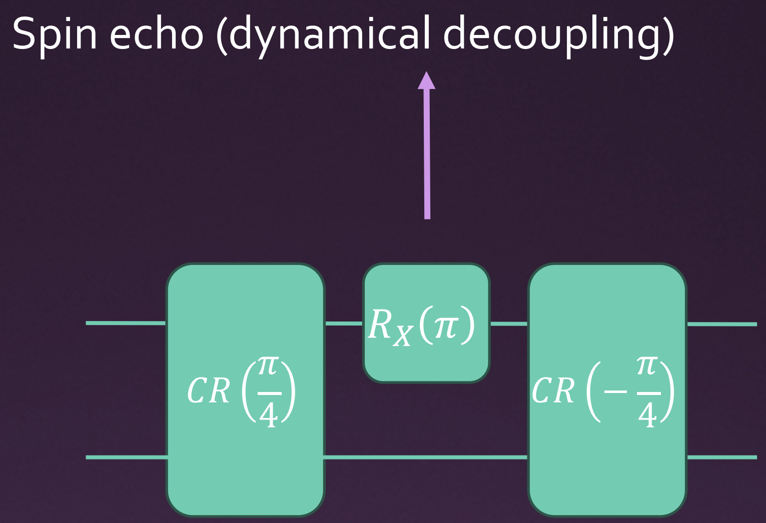
$$\hat{H}_{\text{eff}} = \Omega(t)\left\{\omega_{ix} \boxed{\frac{I_1\left[\cos(\gamma)\,\hat{X}_2 + \sin(\gamma)\,\hat{Y}_2\right]}{2}} + \omega_{zx}\frac{\hat{Z}_1\left[\cos(\gamma)\,\hat{X}_2 + \sin(\gamma)\,\hat{Y}_2\right]}{2}\right\} + \Omega^2(t)\omega_{zi}\cancel{\frac{\hat{Z}_1\hat{I}_2}{2} + \omega_{zz}\frac{\hat{Z}_1\hat{Z}_2}{2}}$$

Compensated by an R_X gate on qubit X

ignore or use dynamical decoupling (DD) to cancel

Echoed cross-resonance gate

127 Qubits	Status:	Online	Median ECR error:	9.034e-3
3.6% EPLG	Total pending jobs:	431 jobs	Median SX error:	2.675e-4
5K CLOPS	Processor type ⓘ:	Eagle r3	Median readout error:	1.600e-2
	Version:	1.2.38	Median T1:	223.52 us
	Basis gates:	ECR, ID, RZ, SX, X	Median T2:	105.32 us
	Your instance usage:	0 jobs		



Dos and Don'ts

- For the Redfield solver, limit the number of qubits to 5 or fewer.
- For Redfield solver, first use the `LinearExponential` algorithm for better simulation speed. Then experiment with other algorithms or increase the number of steps to verify result convergence.
- Use Ohmic bath (instead of Lindblad) for Redfield solver.
- Note that increasing the detuning between the driving frequency and the qubit frequency will slow down the simulation. Typically, these detunings are relatively small, ranging from KHz to MHz.
- To obtain the "noiseless" results:
  - Use Schrödinger equation solver;
  - Set the parasitic qubit-qubit coupling to zero;
  - Ensure the single qubit driving frequency matches the corresponding qubit frequency.

Q&A