

Taller de Qiskit

Manejo de errores con Qiskit Ignis

Bruno Eduardo Ramírez Galindo

Facultad de Ciencias UNAM, QMexico

Qiskit Summer Jam México, Agosto 2021



Tabla de Contenido

① ¿Qué es Qiskit Ignis?

Errores en cómputo cuántico

Qiskit Ignis para caracterización y mitigación

② Caracterización de errores

Errores al aplicar compuertas

Relajación y decoherencia

Error ZZ

¿Para qué caracterizar?

③ Mitigando errores de medición

Errores de medición

Mitigación

④ Conclusiones e ideas



Tabla de Contenido

① ¿Qué es Qiskit Ignis?

Errores en cómputo cuántico

Qiskit Ignis para caracterización y mitigación

② Caracterización de errores

Errores al aplicar compuertas

Relajación y decoherencia

Error ZZ

¿Para qué caracterizar?

③ Mitigando errores de medición

Errores de medición

Mitigación

④ Conclusiones e ideas



«*Nobody's perfect.*»

- Hannah Montana

... y mucho menos las computadoras cuánticas.



Errores al aplicar compuertas

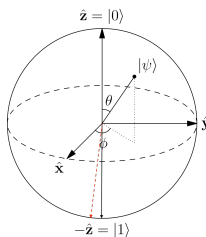
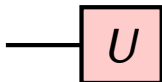


Fig: 1. Una compuerta mal aplicada. *Wikimedia commons.*

Cuando la aplicación de un pulso mal calibrado hace que nuestro qubit acabe en un lugar distinto del deseado.

- Compuertas de un qubit $X, Y, Z, U_1(\theta, \phi, \lambda)$.
- Compuertas de múltiples qubits: CX, CZ, CCX , etc.



Errores inherentes al qubit

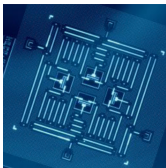


Fig: 2. Qubits superconductores. *Gambetta, J.*

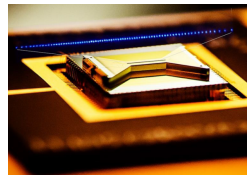


Fig: 3. Qubits de iones atrapados. *Munroe, C.*

Dependen de la «arquitectura» de nuestro qubit y qué tan aislado esté del ambiente. Todos los sistemas físicos tienen interacciones y comportamientos no deseados.

- Error ZZ en qubits superconductores
- Relajación
- Decoherencia



Errores al medir



Fig: 4. Error de flip aleatorio. *Electronics Hub*

Perturbación que afecta un estado medido de forma aleatoria antes de que se den los resultados de medición al usuario. Pueden ocurrir debido a fallos en los canales de comunicación clásica.



¿Que se hace con cada error?

Dependiendo del tipo de error, Qiskit Ignis nos permite hacer una de dos acciones:

- Caracterizar
 - Errores al aplicar compuertas
 - Errores inherentes al qubit
- Mitigar
 - Errores de medición

Además de lo anterior, se dedica mucho tiempo al diseño de códigos para la corrección de errores cuánticos. *Ej.:* Código de Shor, código de repetición, etc.



Lo que nos ofrece Qiskit Ignis

Entre sus herramientas, **Qiskit Ignis** nos brinda herramientas muy buenas para la caracterización y mitigación de errores.

Ignis trabaja muy de cerca con **Terra** y **Aer** para permitir la caracterización y mitigación de errores:

- **Terra**: Para el diseño cuánticos específicos que ayudan a caracterizar y mitigar errores.
- **Aer**: Para simular los modelos de ruido correspondientes a cada tipo de error, para ejecutar los circuitos.



Tabla de Contenido

① ¿Qué es Qiskit Ignis?

Errores en cómputo cuántico

Qiskit Ignis para caracterización y mitigación

② Caracterización de errores

Errores al aplicar compuertas

Relajación y decoherencia

Error ZZ

¿Para qué caracterizar?

③ Mitigando errores de medición

Errores de medición

Mitigación

④ Conclusiones e ideas



¿A qué nos referimos con «caracterización»?

Algunos errores no son tan fáciles de corregir programando, pero podemos aspirar a conocerlos bien. Podemos darnos una idea de la magnitud de su efecto midiendo algunos números (**parámetros característicos**), cuyo significado varía según el tipo de error: ángulos, tiempos, frecuencias.

$$d\theta, d\phi, \xi, T_1, T_2$$

La **caracterización** consta de en la determinación experimental de los **parámetros característicos**. Esto nos es posible gracias a:

```
1 from qiskit.ignis.characterization import *
```



Procedimiento de caracterización general

Notaremos que todos los procedimientos de

```
1 from qiskit.ignis.characterization import *
```

permiten la obtención de un parámetro característico mediante un proceso similar:

- Generar **una serie** de circuitos cuánticos que hacen evidente el error.

```
1 error_circuits(*args)
```

- Ejecutar los circuitos generados (con ruido) y acceder a los conteos de medición de los distintos circuitos.
- Procesar y ajustar los datos de conteos a alguna función $P(\epsilon, n)$ según el tipo de error, ajustamos ϵ al controlar n en el circuito y observar $P(\epsilon, n)$.

```
1 ErrorFitter(*args)
```



Errores en compuertas

Al aplicar compuertas de uno y dos qubits, por ejemplo X y CX . Puede suceder cuando un pulso está mal calibrado en duración, amplitud o frecuencia.

Idealmente un pulso induce una rotación de θ en el eje X :

$$U(\theta, \sigma_x) = e^{-i\theta\sigma_x/2}$$

Pero lo que sucede *en realidad* es algo del estilo:

$$U(\theta + d\theta, \sigma_x) = e^{-i(\theta + d\theta)\sigma_x/2} \rightarrow \text{Err. amplitud}$$

$$U(\theta, d\phi\sigma_x + d\phi\sigma_y) = e^{-i(\theta)(d\phi\sigma_x + d\phi\sigma_y)/2} \rightarrow \text{Err. ángulo}$$



Errores en compuertas con Qiskit

Caracterizando error de ángulo

Parámetro $\Rightarrow d\theta$, un ángulo

$$P_1(d\theta, n) \approx \frac{1}{2} + \frac{(-1)^n d\theta}{2} (2n + 1)$$

Controlamos, n las repeticiones de una secuencia de compuertas, en un circuito

```
1 amp_cal_1q_circuits(max_reps, qubits)
```

Una vez ejecutados los circuitos, queremos ajustar $P_1(d\theta, n)$ para $d\theta$ con

```
1 AmpCalFitter(backend_result, xdata, ...)
```

Similar para los errores de amplitud y en compuertas CX



Relajación

Relajación: Proceso mediante el cual el estado $|1\rangle$ tiende a regresar al estado $|0\rangle$, debido a la tendencia de ocupar el estado de menor energía. **Fenómeno inherente al qubit.**

Gracias, a esto la probabilidad de observar $|1\rangle$ decae con el paso del tiempo, dependiendo del «ritmo de relajación».

$$P_1(T_1, t) = e^{-\frac{t}{T_1}}$$

Existe un tiempo de espera T_1 para el cual encontrar el estado excitado ya es poco probable

$$P_1(T_1, T_1) = e^{-\Gamma_1 t} = e^{-T_1/T_1} = e^{-1} = 1/e \approx 0,368$$

T_1 , tiempo de relajación



Relajación con Qiskit

Caracterizando relajación

Parámetro $\Rightarrow T_1$, un tiempo

$$P_1(T_1, t) = e^{-\frac{t}{T_1}}$$

Controlamos t , el tiempo que transcurre entre la preparación de $|1\rangle$ y la medición, con circuitos

```
1 t1_circuits(num_gates, gate_time, ...)
```

Tras ejecutar los circuitos se ajusta $P_1(T_1, t)$ obtenemos T_1 con

```
1 T1Fitter(backend_result, xdata, ...)
```



Decoherencia con Qiskit

Caracterizando decoherencia

Parámetro $\Rightarrow T_2$, un tiempo

$$P_0(T_2, t) = \frac{1}{2}(1 + e^{-\frac{t}{T_2}})$$

*Notemos que ahora es P_0 y no P_+ , podemos pasar de una a otra con una compuerta H

Nuevamente controlamos t , el tiempo que transcurre entre la preparación de $|+\rangle$ y la medición, con circuitos*

```
1 t2star_circuits(num_gates, gate_time, ...)
2 t2_circuits(num_gates, gate_time, ...)
```

Tras ejecutar los circuitos se ajusta $P_0(T_2, t)$ obtenemos T_2 con*

```
1 T2StarFitter(backend_result , xdata, ...)
2 T2Fitter(backend_result , xdata, ...)
```



Error ZZ

Presente cuando hay dos **qubits superconductores acoplados**, desplazamiento de la energía correspondiente al estado $|11\rangle$. Provoca entrelazamiento no deseado. **Fenómeno inherente al qubit.**

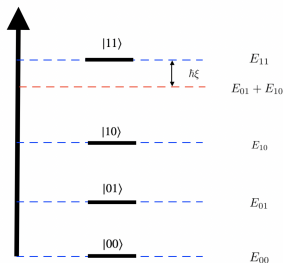


Fig: 5. Desplazamiento de la energía del estado $|11\rangle$

$$H = \hbar\omega_0(1 - \sigma_{Z,0})/2 + \hbar\omega_1(1 - \sigma_{Z,1})/2 + \hbar\xi|11\rangle\langle 11|$$



Error ZZ

El desplazamiento de energía podemos calcularlo:

$$\xi = \omega_{11} - \omega_{10} - \omega_{01}$$

Las frecuencias involucradas podemos calcularlas haciendo *Experimentos de Ramsey*. Observamos oscilaciones de la probabilidad de observar el estado excitado:

$$P_{1,Q_1=|0\rangle}(\omega_{10}, t) = A \cos(\omega_{10}t + \phi)$$

$$P_{1,Q_1=|1\rangle}(\omega_{01}, t) = A \cos(\omega_{01}t + \phi)$$



Error ZZ con Qiskit

Caracterizando error ZZ

Parámetro $\Rightarrow \xi \Rightarrow \omega_{10}$ y ω_{01} , frecuencias*

$$P_{1,Q_1=|0\rangle} = A \cos(\omega_{10}t + \phi), \quad P_{1,Q_1=|1\rangle} = A \cos(\omega_{01}t + \phi)$$

*Recordemos que por $E = \hbar\omega$, también podemos interpretar ξ como una energía

Controlamos t , el tiempo transcurrido en los experimentos de Ramsey, para observar las oscilaciones. Usamos:

```
1 zz_circuits(num_gates, gate_time, ...)
```

Tras ejecutar los circuitos se ajusta $P_{1,\omega_{ij}}(\omega_{ij}, t)$ obtenemos ω_{ij} con

```
1 ZZFitter(backend_result, xdata, ...)
```



¿Para qué caracterizar?

Siempre es bueno saber que ocurre un error y qué tanto está afectando, esto nos permitirá hacer algo para arreglarlo:

- Errores en compuertas: calibrar parámetros de pulso.
- Error ZZ: cambiar la arquitectura del qubit, calibrar acoplamientos.
- Relajación y decoherencia: Mejorar sistemas de aislamiento.

Caracterizar siempre sirve para saber que qubits o dispositivos evitar.



Tabla de Contenido

① ¿Qué es Qiskit Ignis?

Errores en cómputo cuántico

Qiskit Ignis para caracterización y mitigación

② Caracterización de errores

Errores al aplicar compuertas

Relajación y decoherencia

Error ZZ

¿Para qué caracterizar?

③ Mitigando errores de medición

Errores de medición

Mitigación

④ Conclusiones e ideas



Acerca de los errores de medición

Tras hacer una medición aún queda algo más por hacer, mostrar los resultados a quien los analizará. En este proceso de «presentar» los resultados se pueden generar errores. Puede que al medir el bit 0 se muestre al usuario 1 y viceversa.



Fig: 5. Un error común.

- Suelen suceder debido a ruido en canales clásicos
- ¡Se puede corregir si los conocemos bien!

```
1 from qiskit.ignis.mitigation.measurement import
```



La idea detrás de la mitigación

Si sospechamos que nuestras mediciones son erróneas, podemos hacer algo para detectarlas...atrapar a nuestro sistema diciendo mentiras.

- Generar circuitos con resultados predecibles.

```
1 complete_meas_cal(qubit_list, qr, cr, ...)
```

- Ejecutarlos y compararlos con nuestras predicciones, recabar información detallada de cada qubit.

```
1 CompleteMeasFitter(results, state_labels, ...)
```

- Utilizar la información recabada para anular los errores.



Primeros dos pasos visualizados

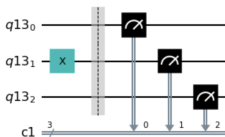


Fig: 6. Un circuito simple.

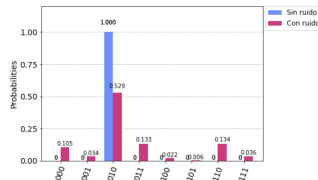


Fig: 7. Conteos del circuito con y sin ruido.

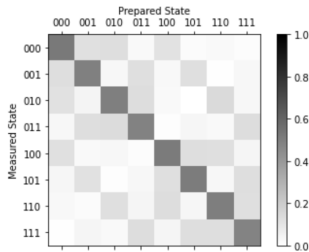


Fig: 8. Datos detallados de qué qubit falla y cuánto.



El último paso

Si conocemos los resultados esperados y los obtenidos, podemos modelar, considerando los diccionarios de conteos como «vectores de conteos». El vector conteos erróneo \vec{c}_e está relacionado con el vector ideal \vec{c}_i mediante:

$$\vec{c}_e = M\vec{c}_i$$

Donde M es la matriz que codifica las probabilidades de fallos, ¡algo que ya tenemos! Podemos aplicar la inversa.

$$\vec{c}_i = M^{-1}\vec{c}_e$$

```
1 r_i = CompleteMeasFitter(results, ...).filter.apply(r_e)
```



Mitigación Tensorial

A veces el ruido solo afecta a un conjunto específico de qubits, en este caso el proceso de mitigación cambia y de hecho es más fácil.

```
1 from qiskit.ignis.mitigation.measurement import (tensored_meas_cal, TensoredMeasFitter)
```

- Generamos menos circuitos.
- Generamos una matriz de calibración más pequeña.

$$M = M_{Loc} \otimes M_{Ideal}$$

El cómputo de los conteos corregidos es más rápido.



Tabla de Contenido

① ¿Qué es Qiskit Ignis?

Errores en cómputo cuántico

Qiskit Ignis para caracterización y mitigación

② Caracterización de errores

Errores al aplicar compuertas

Relajación y decoherencia

Error ZZ

¿Para qué caracterizar?

③ Mitigando errores de medición

Errores de medición

Mitigación

④ Conclusiones e ideas



Conclusiones e ideas

En conclusión, hemos aprendido:

- Hay al menos tres tipos de errores en cómputo cuántico.
- Algunos errores se caracterizan y otros se mitigan.
- **Ignis** (en conjunto con otras componentes de Qiskit) proporciona las herramientas para hacer lo correspondiente con cada error.

Algunas ideas de proyectos para el QSJM:

- Hacer la caracterización y mitigación completa para un dispositivo en específico.
- Caracterizar algunos errores y pensar en qué aspectos del dispositivo físico podemos cambiar para eliminarlos.



¡Gracias!



Referencias

- 1 Gambetta, J. M., Chow, J. M., & Steffen, M. (2017). Building logical qubits in a superconducting quantum computing system. npj Quantum Information, 3(1), 1-7.
- 2 Ball, P. (2018). Ion-based commercial quantum computer is a first. Physics World, December, 17.
- 3 Glosser.ca, (Diciembre 19 de 2012), Bloch sphere; a geometrical representation of a two-level quantum system. Consultado en https://en.wikipedia.org/wiki/File:Bloch_Sphere.svg#filelinks
- 4 Error Correction and Detection Codes: CRC, Hamming, Parity. (Julio 31 de 2019). Consultado en <https://www.electronicshub.org/error-correction-and-detection-codes/>



