

UNIVERSIDAD
NACIONAL
DE COLOMBIA

TESIS DE MAESTRÍA

Algoritmo cuántico para la reconstrucción de estados de espín un medio

Autor:
Cristian Andrey Galvis Flórez

Supervisora:
Dra. Karen Milena Fonseca
Romero

Tesis entregada como requerimiento para optar al
título de Magíster en Ciencias – Física

en el

Departamento de física
Grupo de óptica e información cuántica

2022

TESIS DE MAESTRÍA

Algoritmo cuántico para la reconstrucción de estados de espín un medio

Facultad de ciencias
Departamento de física

Magíster en Ciencias – Física

Por: Cristian Andrey Galvis Flórez

Yo, Karen Milena Fonseca Romero, declaro que esta tesis titulada, «Algoritmo cuántico para la reconstrucción de estados de espín un medio»; cumple con mi aprobación para su respectiva lectura y calificación.

Firma:

UNIVERSIDAD NACIONAL DE COLOMBIA - SEDE BOGOTÁ

Resumen

Facultad de ciencias
Departamento de física

Magíster en Ciencias – Física

Algoritmo cuántico para la reconstrucción de estados de espín un medio

Por: Cristian Andrey Galvis Flórez

Resumen:

La estimación de estados cuánticos es una tarea importante que se emplea en múltiples protocolos de información cuántica. En este trabajo se considera una familia de operadores unitarios de evolución dependientes de un parámetro (dos parámetros) que permiten la estimación de una componente del espín (todas las componentes del espín) de un sistema de dos niveles. La función de transferencia de tomografía cuántica ($qTTF$), que corresponde a la traza de la inversa de la matriz de información de Fisher, se usa para cuantificar el rendimiento del estimador. En este trabajo, se optimiza la $qTTF$ para los dos estimadores. El mínimo de la $qTTF$ del modelo de un parámetro se alcanza cuando el poder de entrelazamiento del operador unitario asociado es máximo. Los dos modelos son simulados en una unidad cuántica de procesamiento de IBM. Mientras que la implementación del modelo para estimación de una componente funciona satisfactoriamente, el modelo para la estimación total del espín muestra grandes errores debido a la profundidad del circuito asociado.

Palabras clave: Estimación de estado cuántico, función de transferencia de tomografía cuántica, Computación cuántica, IBM Quantum Experience.

Abstract

Facultad de ciencias
Departamento de física

Magíster en Ciencias – Física

Quantum algorithm for state reconstruction of one-half spin particles

By: Cristian Andrey Galvis Flórez

Abstract:

Quantum state estimation is an important task of many quantum information protocols. We consider a one-parameter (resp. two-parameter) family of unitary evolution operators which allow the estimation of a single spin component (resp. all spin components) of a two-level system. The quantum tomographic transfer function (qTTF), the average of the trace of the inverse of the Fisher information matrix, is used as a quantifier of tomographic performance. In this work, we optimize the qTTF of both estimation models. The minimum of the qTTF of the one-parameter model is attained when the entangling power of the associated unitary operator is maximum. Both models were run on an IBM quantum processing unit. While the implementation of the estimation of a single-spin component is quite satisfactory, the implementation of the whole spin estimation displays rather large errors due to the relatively large depth of the associated circuit.

Keywords: Quantum state estimation, Quantum tomographic transfer function, Quantum Computing, IBM Quantum Experience.

Agradecimientos

Inicialmente agradezco a la profesora Karen Milena Fonseca por la tutoría de éste trabajo. A pesar de que yo no tenía antecedentes como investigador en el campo de la información cuántica, ella accedió a supervisar este trabajo que nos ha ayudado a explorar el novedoso e interesante campo de la computación cuántica.

Además, quiero agradecer a mi familia, especialmente a mi madre Adelaida Florez. Su apoyo ha sido fundamental para el desarrollo de este trabajo. Su paciencia y confianza en esta vocación, a pesar de no tener un antecedente académico, solo se pueden entender desde los ojos y el amor de una madre, lo cual me hace sentir inmensamente agradecido.

Quiero agradecer especialmente a Daniel Saavedra, ya que su trabajo fue el primer escalón que tuve que subir en el desarrollo de esta maestría. Adicionalmente, quiero agradecer a Álvaro Martínez, cuya contribución en el entendimiento de los parámetros para caracterizar los estimadores fue fundamental.

También quiero agradecer a los compañeros del Grupo de Óptica e Información Cuántica (GOIC). Pese a que este trabajo se desarrolló en medio de una pandemia mundial que limitó el conocimiento físicamente entre los miembros del grupo, logramos realizar amistades importantes y contribuir con un pequeño grano de arena a la introducción de Colombia a la organización Qworld bajo el nombre de QColombia, un primer paso para el desarrollo de la computación cuántica en el país.

Finalmente quiero agradecer a mi segundo hogar, la Universidad Nacional de Colombia. Una institución de excelente calidad, que me ha permitido conocer y entender el universo desde la hermosa perspectiva de las ciencias físicas, y me ha permitido acceder a este título de Maestro en Ciencias.

Índice general

Resumen	V
Agradecimientos.	VII
Introducción	1
1. Computación cuántica	4
1.1. Bits cuánticos e información cuántica	4
1.1.1. Operadores	5
1.1.2. Entrelazamiento	6
1.2. Operador densidad	6
1.3. Evolución temporal	7
1.4. Medición en mecánica cuántica	8
1.4.1. Mediciones POVM	9
1.5. Cómputo con sistemas cuánticos	9
1.5.1. Compuertas de un qubit	10
1.5.2. Compuertas de múltiples qubits	11
1.5.3. Medición	13
1.6. Modelo cuántico de circuito	14
2. Tomografía de espín	16
2.1. Estimación de una componente del espín	16
2.1.1. Elementos POVM para la estimación de s_z	18
2.2. Estimación del estado total de espín	19
2.2.1. Elementos POVM para la estimación total del espín	21
2.2.2. Construcción del estimador	22
2.3. Caracterización del estimador	23
2.3.1. Parámetro de error de Fisher	25
2.3.2. Error del estimador de s_z	27
2.3.3. Error del estimador total de espín	31
3. Simulación en un computador cuántico	34
3.1. Qiskit	34
3.2. Estimación de una componente del espín	38
3.2.1. Construcción del circuito cuántico	38
3.2.2. Simulación en una QPU	38
3.3. Simulación del modelo de Saavedra	40
3.3.1. Construcción del circuito cuántico	40
3.3.2. Simulación en una QPU	40
Conclusiones y perspectivas	44
A. Expresiones explícitas de los coeficientes a_μ, b_μ, y c_μ	46

B. Equivalencia de la varianza de s_z y el parámetro de Fisher.	47
C. Equivalencia de la varianza del estimador \hat{s} y el parámetro de Fisher.	49

Índice de figuras

1.1.	Esfera de Bloch	10
1.2.	Representación gráfica de compuertas lógicas en un circuito cuántico.	11
1.3.	Compuertas H y X aplicadas sobre los qubits A y B respectivamente, representadas en un circuito cuántico. Esta representación se generaliza para más qubits.	12
1.4.	Representación de una CNOT en un circuito cuántico.	12
1.5.	Representación gráfica de la operación U controlada por el qubit A	13
1.6.	Representación gráfica del proceso de medición sobre un qubit en un circuito cuántico.	13
1.7.	Uno de los estados de la base de Bell representado como un circuito cuántico.	14
2.1.	Ilustración de la interacción de un sistema de interés S con un metro A para estimar una componente del espín.	16
2.2.	Esquema de medición del estado total de espín de la partícula S . Nuevamente, este esquema puede representarse en términos de qubits.	20
2.3.	Parámetro del error F^{-1} como función de α_1 para los valores $\theta = \frac{\pi}{2}, \frac{2\pi}{3}, \frac{3\pi}{4}$ y π . Como se puede ver, el error decrece cuando $\theta \rightarrow \pi$	28
2.4.	Entropía del sistema $E(\rho_S)$ como función de α_1 para los valores $\theta = \frac{\pi}{2}, \frac{2\pi}{3}, \frac{3\pi}{4}$ y π . Como se puede ver, el comportamiento de la entropía es similar al del parámetro del error presentado en la figura 2.3.	29
2.5.	Valor medio del parámetro del error $\overline{F^{-1}(\theta)}$ (línea sólida) y valor máximo del error F_{max}^{-1} (línea discontinua) como funciones del parámetro θ	30
2.6.	Comportamiento del parámetro de error de Fisher asociado al estimador total de espín respecto a los parámetros ξ que caracterizan el estado inicial. Para este gráfico se usan los valores óptimos del estimador $\theta_A^{min} \approx 3,45$ y $\theta_B^{min} \approx -8,42$	33
3.1.	Circuito cuántico para la construcción de un estado de Bell. Antes de la medición, el estado de los qubits 0 y 1 es $ \psi\rangle = \frac{1}{\sqrt{2}}(00\rangle + 11\rangle)$	35
3.2.	Topología de la QPU <code>ibmq_lima</code>	36
3.3.	Histograma de los resultados del circuito en la figura 3.1. Las barras negras muestran el resultado de la simulación clásica (resultado ideal), mientras que las barras azules muestran los resultados de la simulación en la QPU <code>ibmq_lima</code>	37
3.4.	Circuito cuántico para la estimación de una componente del espín del qubit S usando el qubit auxiliar A	38
3.5.	a) Simulación ideal del circuito 3.4 en el simulador local <code>qasm_simulator</code> . b) Simulación en la QPU <code>ibmq_lima</code> . Esta simulación muestra un sesgo respecto al resultado teórico, indicando la presencia de errores de hardware que afectan las mediciones. c) Simulación usando el mitigador de <code>qiskit</code> , no mejora los resultados.	38

3.6. Parámetro qTTF sobre N y varianza del estimador de s_z respecto a las N muestras. 40

3.7. Estructura general del circuito cuántico usado para la simulación del modelo de estimación total de espín. 41

3.8. Simulación del circuito asociado para la estimación del espín de un qubit con qubits auxiliares para el estado inicial $|\psi\rangle = \cos \frac{\pi}{8} |0\rangle + e^{i\frac{\pi}{3}} \sin \frac{\pi}{8} |1\rangle$. Se muestra la simulación esperada (negro), simulación en una QPU (azul oscuro), simulación en una QPU con mitigador (azul claro). 41

3.9. Parámetro qTTF sobre N y varianza del estimador total de espín respecto a las N muestras. 42

Índice de tablas

3.1. Estimación de s_z para los seis autoestados de los operadores de Pauli, en la QPU <code>ibmq_1ima</code> para los ángulos $\theta = \pi, 2\pi/3, \pi/2$. Cada estado fue simulado 5 veces con 1024 disparos en cada una, resultando así en estimaciones con 5×1024 disparos.	39
3.2. Estimación de estado total de espín para los seis autoestados de los operadores de Pauli, en la QPU <code>ibmq_1ima</code> para los ángulos óptimos $\theta_A \approx 3,45$ y $\theta_B \approx -8,42$. Cada estado fue simulado 5 veces con 1024 disparos en cada una, resultando así en estimaciones con 5×1024 disparos.	42

Dedicado a mi familia...

Introducción

La computación cuántica no tiene un inicio claro; sin embargo, hay un artículo de 1980 publicado por el físico estadounidense Paul Benioff (Benioff, 1980) en el cual propone un primer modelo para realizar una máquina de Turing con base en estados cuánticos. Recordemos que la máquina de Turing consta de una cinta con un cabezal que puede recorrer dicha cinta y cambiar los valores que hay en los espacios de la misma. Esta máquina permite realizar diferentes cálculos numéricos como lo sugirió Alan Turing (Turing, 1938).

Con la propuesta de una máquina de Turing cuántica, en 1982, Richard Feynmann (Feynman, 1982; Feynman, 1986) establece el modelo de circuito cuántico; con base en bits cuánticos, conocidos como *qubits* y los operadores que pueden actuar sobre ellos, a los que se llaman *compuertas cuánticas*. En 1995 Adriano Barenco, en compañía de varios científicos que han aportado al desarrollo de la computación cuántica como Charles Bennett, David DiVincenzo y Peter Shor, establecen un conjunto de compuertas elementales sobre las cuales se puede construir cualquier circuito cuántico (Barenco y col., 1995). Para este entonces ya se habían desarrollado diferentes algoritmos con base en este tipo de computación como lo son el algoritmo de Grover (Grover, 1996), que se utiliza para buscar elementos en una lista, o el de Shor (Shor, 1994), que es una alternativa para realizar factorización numérica. Los dos algoritmos presentan significativas mejoras respecto a sus contrapartes clásicas.

Luego de que se estableció un modelo teórico para la computación cuántica, empezó su desarrollo experimental. Un avance importante fue hecho en 1998, donde un grupo liderado por Issac Chuang en la universidad de California, logran ejecutar experimentalmente el algoritmo de Grover para un sistema con 4 qubits (Chuang, Gershenfeld y Kubinec, 1998). Estos desarrollos experimentales para el manejo de qubits, basados en diferentes sistemas físicos, han llevado la computación cuántica que conocemos hoy en día. El punto de despegue de la computación cuántica actual se puede decir que se da en 2019 cuando Google proclama haber alcanzado la supremacía cuántica. Aquí se muestra un procesador programable de 53 qubits que supera a un procesador clásico en una prueba específica. La prueba consiste en crear estados cuánticos aleatorios de 53 qubits y tomar $N = 10^6$ mediciones de cada circuito para tener la distribución de probabilidad de los posibles estados y así compararlo con los resultados teóricos. El problema con simular estados cuánticos en procesadores clásicos radica en que, para simular estados superpuestos, la cantidad de bits que se necesitan crece exponencialmente con el número de qubits del circuito. En los resultados que reporta Google, la simulación de un circuito con $m = 20$ ciclos toma alrededor de 200 segundos en el procesador cuántico, mientras que realizar esta tarea con millones de núcleos de computadores clásicos tomaría alrededor de 10.000 años, además, verificar la fidelidad de los resultados tomaría un millón de años más (Arute y col., 2019).

A partir de este punto ha empezado una revolución por parte de la industria para invertir por este tipo de computación. Hay diferentes tipos de procesadores cuánticos. Por ejemplo, D-wave tiene procesadores que funcionan con lo que se conoce como *Quantum Annealing* (Johnson y col., 2011; Jones, 2013). Xanadu, por otra parte, desarrolló

un software para el manejo de procesadores cuánticos ópticos (Killoran y col., 2019) y actualmente se encuentra cerca de desarrollar el primer procesador cuántico óptico programable (Arrazola y col., 2021). Por otra parte, IBM utiliza unidades cuánticas de procesamiento (QPU del inglés Quantum Processor Unit) con base en superconductores; uno de sus primeros test experimentales con la actual serie de procesadores se da en 2017 (Kandala y col., 2017). Luego, IBM presenta Qiskit (Asfaw y col., 2020), una herramienta de software que permite a los usuarios construir, simular y ejecutar circuitos cuánticos en procesadores cuánticos reales.

Ahora se introduce el problema de la tomografía de estados cuánticos, en el cual se busca estimar el estado de superposición de un sistema cuántico de la forma más precisa y usando la menor cantidad de recursos posibles. Este problema cobra importancia ya que, para evaluar el rendimiento de un algoritmo cuántico, es necesario, entre otras tareas, estimar el estado cuántico de la QPU (Banaszek, Cramer y Gross, 2013; James y col., 2001). En cualquier caso, la estimación de unos cuantos qubits es una tarea desafiante dado que, el número de parámetros necesarios para la estimación de un estado de N qubits, es del orden 2^{2N} .

Hoy en día, diferentes tipos de algoritmos cuánticos que buscan realizar tomografía de estados cuánticos son implementados en QPUs (Kiktenko, Kublikova y Fedorov, 2020; Gaikwad y col., 2022). Además de los clásicos métodos de reconstrucción tomográfica de un estado cuántico, en los cuales se realizan mediciones sobre múltiples copias del estado a estimar para generar un conjunto de datos que permite realizar la estimación, se han desarrollado diferentes técnicas de estimación de qubits. Algoritmos de aprendizaje de máquina clásico (Lohani y col., 2021), optimizaciones estocásticas (Ferrie, 2014) y técnicas de máxima entropía (Gupta y col., 2021; Gupta, Levine y Kais, 2021) son algunas de las técnicas que se han empleado para realizar estimación del estado de un qubit. En este trabajo, se usa un estimador de inversión lineal o un estimador de máxima verosimilitud (Matteo Paris, 2004; Teo, 2015) para realizar la estimación del estado de un qubit.

El problema de la tomografía se puede trasladar al caso específico de la estimación del estado de espín de una partícula de espín un medio. En 1986, Asher Peres (Peres, 1986) propone un modelo para estimar una componente del espín. Eso funciona al hacer interactuar un sistema de espín un medio con otra partícula de espín un medio, a la que llamaremos metro (metro A), que está en contacto con un reservorio. Peres muestra que, con la interacción apropiada, es posible estimar la componente z del espín de nuestro sistema a través de la medición del metro A.

Luego, en 2017, Martí Perarnau-Llobet propone un esquema en el cual, a partir de dos metros, es posible hacer una estimación de 2 componentes del espín del sistema (Perarnau-Llobet y Nieuwenhuizen, 2017). Finalmente, en 2019, Daniel Saavedra y Karen Fonseca proponen un modelo en el cual es posible hacer una tomografía total del estado de espín de una partícula usando 2 metros auxiliares (Saavedra y Fonseca-Romero, 2019). El modelo propuesto en (Saavedra y Fonseca-Romero, 2019) no se ha simulado en un procesador cuántico hasta la fecha.

En este trabajo se propone realizar una simulación de los modelos de estimación parcial y total del estado de espín de una partícula de espín un medio. Para esto se toman los modelos de Asher Peres y Daniel Saavedra y se ejecutan en un computador cuántico real para evaluar su rendimiento siguiendo la siguiente estructura. Primero se ilustra el marco teórico necesario para entender el modelo de estimación, y se define un criterio que permite caracterizar el rendimiento del estimador. Luego, se realiza la representación del problema de estimación en un circuito cuántico, el cual es ejecutado

múltiples veces en un simulador local (simulación clásica), luego se ejecuta en una QPU (simulación cuántica). Los resultados se comparan con lo que predice la teoría y así se analiza el rendimiento de los estimadores.

El trabajo se estructura de la siguiente manera. En el capítulo 1 se hace una breve introducción a la computación cuántica en la cual se manifiestan los elementos teóricos necesarios para la comprensión y elaboración de un circuito cuántico. En la parte final, se muestra el modelo cuántico de circuito que se utiliza en la posterior simulación de los modelos de estimación.

En el capítulo 2 se desarrollan los modelos teóricos para la estimación parcial y total del estado de espín de una partícula de espín un medio. Adicionalmente, se desarrolla el modelo teórico para la caracterización de los estimadores con base en la función de transferencia de tomografía cuántica (qTTF del inglés Quantum Tomographic Transfer Function). A través de la qTTF se encuentran los parámetros óptimos para los estimadores parcial y total del espín.

Finalmente, en el capítulo 3 se reporta la simulación de los modelos previamente propuestos en un computador cuántico de IBM. Para esto se ilustra brevemente el software que se va a utilizar para la simulación y cómo funciona con un ejemplo sencillo. Además, se muestra la topología de la QPU escogida para realizar las simulaciones de los modelos. Luego, se construyen los circuitos cuánticos que representan los modelos de estimación para finalmente realizar las respectivas simulaciones. Se implementan 2 tipos de simulación: una local, que se realiza en un procesador clásico y cuyos resultados son lo más cercano a lo ideal; por otra parte, se realiza la simulación en el computador cuántico, con el objetivo de compararlas y cuantificar el rendimiento de la simulación en un computador cuántico respecto a lo que idealmente se debería esperar. En el capítulo final se presentan las conclusiones y perspectivas futuras a tener en cuenta para el desarrollo de nuevos trabajos.

Capítulo 1

Computación cuántica

En este capítulo se realiza una descripción del modelo de computación cuántica de compuerta universal. Este modelo es el que se utiliza para realizar la simulación de los protocolos de estimación de espín presentados en este trabajo. Inicialmente se describen los qubits desde la teoría cuántica de la información y cómo su estado cambia bajo la acción de operadores cuánticos. Luego, se muestra el marco de las mediciones generalizadas, fundamental para entender el funcionamiento de los estimadores de espín. Finalmente, se muestra cómo, a través de qubits, es posible realizar cálculos computacionales usando el modelo cuántico de circuito. Este modelo se implementará posteriormente para la simulación de los problemas de estimación parcial y total del espín.

1.1. Bits cuánticos e información cuántica

La teoría clásica de la información se basa en lo que se conoce como un bit. Un bit es algún sistema cualquiera que puede tomar únicamente dos valores bien definidos a los que se conocen como "0" y "1".

Con esta idea en mente se desarrolló un equivalente cuántico del bit, popularmente llamado *qubit* (quantum bit). Un qubit es un sistema cuántico cualquiera de 2 niveles distinguibles bajo la medición del valor esperado de un operador. A estos estados los llamaremos $|0\rangle$ y $|1\rangle$. Estos dos estados son la base de un espacio de Hilbert de dimensión 2 y por lo tanto un estado cuántico en dicho espacio se puede representar como una combinación lineal de los elementos de la base (Diósi, 2007; Wilde, 2013). El principio de superposición nos permite describir un estado general de un qubit de la forma

$$|\psi\rangle = c_0 |0\rangle + c_1 |1\rangle. \quad (1.1)$$

Este es un estado normalizado, por lo tanto $|c_0|^2 + |c_1|^2 = 1$. Después de realizar una medición sobre este estado, se tiene una probabilidad $|c_0|^2$ de encontrarlo en el estado $|0\rangle$ y $|c_1|^2$ de encontrarlo en el estado $|1\rangle$. Estos estados, al formar una base, cumplen

$$\langle 0|0\rangle = 1, \quad (1.2)$$

$$\langle 1|1\rangle = 1, \quad (1.3)$$

$$\langle 0|1\rangle = \langle 1|0\rangle = 0. \quad (1.4)$$

Si tenemos un sistema con más de un qubit, las relaciones de ortonormalidad previamente mencionadas se satisfacen por cada uno de ellos. En general, esta relación de ortonormalidad se escribe de la forma

$$\langle m|m'\rangle = \delta_{mm'}. \quad (1.5)$$

Acá m y m' toman los valores 0 o 1.

Es posible hacer una representación vectorial de los qubits de la forma

$$|0\rangle \doteq \begin{pmatrix} 1 \\ 0 \end{pmatrix}; \quad |1\rangle \doteq \begin{pmatrix} 0 \\ 1 \end{pmatrix}. \quad (1.6)$$

1.1.1. Operadores

La evolución de un sistema cuántico cerrado se puede describir en términos de una transformación unitaria. Existen infinitas operaciones que se pueden aplicar sobre un sistema cuántico de dos niveles (Diósi, 2007; Wilde, 2013; Watrous, 2018). Acá mostraremos algunas operaciones importantes que luego se verán con frecuencia en el contexto de la computación cuántica y que son importantes para el desarrollo y entendimiento de protocolos cuánticos básicos.

Una operación importante es la identidad I . Esta transformación deja invariante el estado del sistema y se representa matricialmente de la forma

$$I \doteq \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}. \quad (1.7)$$

Otra transformación importante se representa por el operador $\sigma_x = X$. Esta operación se puede entender como una negación del estado; por lo tanto, en este caso lo que hace es intercambiar los elementos de la base, es decir

$$X|0\rangle \rightarrow |1\rangle, \quad X|1\rangle \rightarrow |0\rangle. \quad (1.8)$$

La negación se puede representar matricialmente de la forma

$$X \doteq \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}. \quad (1.9)$$

La siguiente transformación de interés se conoce como el cambio de fase relativa o "phase-flip". Se representa con el operador $\sigma_z = Z$. Esta cambia la fase relativa entre el estado $|0\rangle$ y el estado $|1\rangle$ en un factor de π , por lo tanto,

$$Z|0\rangle \rightarrow |0\rangle, \quad Z|1\rangle \rightarrow -|1\rangle. \quad (1.10)$$

Matricialmente se puede representar de la forma

$$Z \doteq \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}. \quad (1.11)$$

Otra transformación importante es la transformación de Hadamard, que se representa con el operador H . Este operador actúa sobre la base del espacio de la forma

$$H|0\rangle \rightarrow |+\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle), \quad H|1\rangle \rightarrow |-\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle). \quad (1.12)$$

El operador de Hadamard se representa matricialmente de la forma

$$H \doteq \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}. \quad (1.13)$$

Los estados $|+\rangle$ y $|-\rangle$ resultan ser los autoestados del operador $\sigma_x = X$. Por lo cual, es posible realizar mediciones para encontrar el valor esperado del operador y obtener dos estados distinguibles. Es decir, es una base de medición alternativa a la base computacional.

Entonces, el estado de superposición que describe un qubit se puede modificar aplicando operadores unitarios a conveniencia. La propiedad de superposición y la posibilidad de modificar estos estados, representa una ventaja respecto a un sistema de bits clásicos. Esta es una de las propiedades que hace que la computación cuántica tenga un mayor alcance a la computación clásica. La otra propiedad importante que permite tomar ventaja de los bits cuánticos es el entrelazamiento.

1.1.2. Entrelazamiento

El entrelazamiento es fundamental para el desarrollo de algoritmos cuánticos con varios qubits. Imaginemos una fuente de dos partículas cuánticas (A y B) emitidas en direcciones opuestas. Además, supongamos que por alguna razón (como la conservación del momento angular) estas partículas tienen que tener estados opuestos de momento angular. Si identificamos estos dos posibles estados de momento angular con los estados $|0\rangle$ y $|1\rangle$ y suponemos que los dos estados posibles para este sistema tienen la misma probabilidad, entonces, el estado de las dos partículas se puede escribir de la forma (Bouwmeester, Ekert y Zeilinger, 2001)

$$|\psi\rangle_{AB} = \frac{1}{\sqrt{2}} \left(|0\rangle_A |1\rangle_B + e^{i\chi} |1\rangle_A |2\rangle_B \right). \quad (1.14)$$

Aquí χ es una fase determinada por las propiedades internas de la fuente de este par de partículas. Por simplicidad tomamos $\chi = 0$. La ecuación (1.14) es una representación de un estado cuántico entrelazado de dos partículas. Es importante recalcar que ni la partícula A ni la partícula B tienen un estado definido; sin embargo, cuando se mide el estado de una partícula y se encuentra un resultado el cual es totalmente al azar, el estado de la otra partícula queda completamente definido. Entonces, cuando se mide la partícula A y por ejemplo tiene el estado $|0\rangle$, inmediatamente la partícula B se encuentra el estado $|1\rangle$; así mismo, cuando la partícula A es medida en el estado $|1\rangle$, la partícula B se encuentra en el estado $|0\rangle$. La misma lógica se aplica cuando la medición se hace inicialmente sobre la partícula B .

Cualquier fuente que produzca pares de partículas entrelazadas se conoce como una fuente EPR, por el famoso artículo de A. Einstein, B. Podolsky y N. Rosen (Einstein, Podolsky y Rosen, 1935). En este artículo los autores muestran que un estado entrelazado no satisface el principio de causalidad, por lo cual, indican que la descripción cuántica de fenómenos físicos está incompleta. Ellos proponen un conjunto de variables ocultas que expliquen estos resultados anómalos; sin embargo, se ha demostrado que esto no es necesario.

1.2. Operador densidad

Los estados cuánticos se pueden representar de una forma más general usando el operador densidad. Para el caso de un qubit que se encuentra en el estado $|\psi\rangle = c_0|0\rangle + c_1|1\rangle$ se define el operador densidad de la forma

$$\begin{aligned} \rho &= |\psi\rangle\langle\psi| \\ &= |c_0|^2 |0\rangle\langle 0| + c_0c_1^* |0\rangle\langle 1| + c_0^*c_1 |1\rangle\langle 0| + |c_1|^2 |1\rangle\langle 1|. \end{aligned} \quad (1.15)$$

Matricialmente se puede escribir de la forma

$$\rho = \begin{pmatrix} |c_0|^2 & c_0c_1^* \\ c_0^*c_1 & |c_1|^2 \end{pmatrix}. \quad (1.16)$$

Sin embargo, en la realidad un sistema cuántico de dos niveles, no se suele encontrar en un *estado puro*, es decir, un estado como en la ecuación (1.1). En cambio se sabe que el sistema se encuentra en una mezcla estadística de estados puros. Es decir, se tiene un conjunto de estados (Benenti, Casati y Strini, 2004)

$$\{|\psi_1\rangle, |\psi_2\rangle \dots |\psi_l\rangle\},$$

con probabilidades

$$\{p_1, p_2, \dots, p_l\},$$

que satisfacen la condición $\sum_k p_k = 1$. Cada uno de los estados $|\psi_k\rangle$ es puro, por lo que no necesariamente son ortogonales. Cuando un estado cuántico se encuentra en esta mezcla estadística de estados puros, lo identificamos como un *estado mezclado*.

Con esto en mente, el operador densidad se define de una forma más general como

$$\rho \doteq \sum_k p_k |\psi_k\rangle \langle \psi_k|. \quad (1.17)$$

En este punto definimos una base ortonormal $\{i\}$, con $i = 1, 2, \dots, d$ donde d es la dimensión del espacio de Hilbert \mathcal{H} . En este espacio es posible expresar los estados $|\psi_k\rangle$, y de la misma forma, podemos representar matricialmente el operador densidad. Los elementos de esta representación matricial son

$$\rho_{ij} = \langle i | \rho | j \rangle.$$

Con base en esto, también es posible definir el valor medio de un observable A por la medición del estado ρ como

$$\langle A \rangle \doteq \text{Tr}(\rho A) = \sum_k p_k \langle \psi_k | A | \psi_k \rangle. \quad (1.18)$$

Al igual que un estado puro, la evolución de un estado representado por el operador densidad también se puede representar con un operador unitario.

1.3. Evolución temporal

La evolución de un sistema cuántico cerrado se describe en términos de un operador unitario U . Este operador tiene que cumplir con la condición de unitariedad ($U^\dagger U = I$) ya que esta asegura que el estado resultante de la operación sigue cumpliendo la normalización con la unidad $\sum_k p_k = 1$.

La evolución temporal de un estado cuántico se describe por la ecuación de Schrödinger

$$i\hbar \frac{d}{dt} |\psi\rangle = H |\psi\rangle. \quad (1.19)$$

En el caso más general, se escribe la ecuación de evolución para el operador densidad de la forma

$$\frac{d}{dt} \rho(t) = \frac{1}{i\hbar} [H, \rho(t)], \quad (1.20)$$

conocida como la ecuación de von Neumann (Nielsen y Chuang, 2011; Sakurai y Napolitano, 2014).

De este modo, el operador de evolución asociado al hamiltoniano H es

$$U(t) = \mathcal{T} \left(e^{-\frac{i}{\hbar} \int_{t_0}^t H(t') dt'} \right), \quad (1.21)$$

donde se utiliza el operador de ordenamiento temporal de Dyson \mathcal{T} , el cual ordena poniendo a la derecha los operadores que actúan en tiempos más tempranos,

$$\mathcal{T}(H(t_1)H(t_2)) \doteq \begin{cases} H(t_1)H(t_2) & \text{if } t_1 \geq t_2, \\ H(t_2)H(t_1) & \text{if } t_2 > t_1. \end{cases} \quad (1.22)$$

Si el Hamiltoniano es independiente del tiempo, el operador de evolución toma la forma

$$U = e^{-\frac{i}{\hbar}Ht}. \quad (1.23)$$

Luego, la evolución de un sistema cuántico bajo la acción de un operador unitario se escribe de la forma

$$|\psi\rangle \xrightarrow{U} U|\psi\rangle, \quad (1.24)$$

o en el caso del operador densidad

$$\rho = \sum_k p_k |\psi_k\rangle \langle \psi_k| \xrightarrow{U} \sum_k p_k U|\psi_k\rangle \langle \psi_k| U^\dagger = U\rho U^\dagger. \quad (1.25)$$

Así, los operadores presentados en la sección 1.1 son operadores unitarios que describen adecuadamente la evolución de un sistema cuántico; en este caso, de 2 niveles.

1.4. Medición en mecánica cuántica

El proceso de medición de un estado cuántico se puede describir en términos de un conjunto de operadores $\{M_i\}$. El índice i se refiere a los posibles resultados de la medición cuando se hace el experimento. Estos operadores no son necesariamente autoadjuntos, sin embargo, si deben cumplir la relación de completez

$$\sum_i M_i^\dagger M_i = I. \quad (1.26)$$

Si el estado del sistema justo antes de la medición es $|\psi\rangle$, la probabilidad de obtener el resultado i al realizar el proceso de medición es

$$p_i \doteq \langle \psi | M_i^\dagger M_i | \psi \rangle. \quad (1.27)$$

Podemos ver fácilmente que, la relación de completez de los operadores de medición se debe al hecho de que la suma de probabilidades debe sumar uno

$$\sum_i p_i = \sum_i \langle \psi | M_i^\dagger M_i | \psi \rangle = \langle \psi | \sum_i M_i^\dagger M_i | \psi \rangle = \langle \psi | I | \psi \rangle = 1.$$

Una vez se realiza el proceso de medición, y se obtiene el i -ésimo resultado, el estado del sistema cambia y toma la forma

$$\frac{M_i |\psi\rangle}{\sqrt{\langle \psi | M_i^\dagger M_i | \psi \rangle}}. \quad (1.28)$$

El proceso de medición se puede describir también en términos del operador densidad del sistema. Si el sistema está en el estado ρ antes de ser medido, el estado posterior a la medición es

$$\frac{M_i \rho M_i^\dagger}{\text{Tr}[M_i \rho M_i^\dagger]}$$

con probabilidad

$$p_i = \text{Tr}[M_i \rho M_i^\dagger]. \quad (1.29)$$

Un caso especial de las mediciones generalizadas corresponde a los que se conocen como mediciones proyectivas; estas corresponden a operadores de medición M_i que son proyectores ortogonales, es decir,

$$M_i^\dagger = M_i \quad M_i M_j = \delta_{ij} M_i. \quad (1.30)$$

En este caso la relación de completitud toma la forma $\sum_i M_i = I$. Es importante resaltar que, una medición proyectiva descrita por las condiciones en (1.30), acompañada por operaciones unitarias se vuelve una medición generalizada en un espacio donde se añaden qubits auxiliares. Es decir, una medición generalizada es equivalente a una medición proyectiva en un espacio de Hilbert mayor (Benenti, Casati y Strini, 2004). Esto se conoce como el teorema de Newmark.

1.4.1. Mediciones POVM

El formalismo de mediciones generalizadas en mecánica cuántica implica un par de cosas importantes. En primer lugar, ofrece un marco teórico para describir la estadística de un proceso de medición a partir de un operador general. Por otra parte, nos indica cual es el estado después del proceso de medición (Nielsen y Chuang, 2011). Los operadores POVM (por sus siglas en inglés *Positive Operator-Valued Measurements*) se suelen usar para la descripción de experimentos en los que el sistema se mide únicamente una vez, por lo cual, no hay interés en el estado luego de la medición. Un ejemplo de esto, es la detección de un fotón por un fotomultiplicador; cuando se realiza la medición el fotón es destruido, por lo que no es posible hacer una medición posterior (Benenti, Casati y Strini, 2004).

En el caso de la computación cuántica, se espera tener mucho control sobre el proceso de medición que se ejecuta sobre los qubits, por lo cual, el formalismo de las mediciones POVM cobra gran importancia.

Una medición POVM se describe por un conjunto de operadores positivos E_i que se conocen como los elementos POVM y se definen en términos de los operadores de medición $\{M_i\}$ de la forma

$$E_i \doteq M_i^\dagger M_i. \quad (1.31)$$

Por la relación de completitud de los operadores de medición, los elementos POVM cumplen $\sum_i E_i = I$. Esto quiere decir que, los operadores E_i son suficientes para determinar las probabilidades de los posibles resultados de la medición a partir de la relación $p_i = \langle \psi | E_i | \psi \rangle$. de este modo, las mediciones proyectivas son un caso particular de POVM, en el que $E_i = M_i$ descritos en la ecuación (1.30).

1.5. Cómputo con sistemas cuánticos

Una vez introducidos los conceptos básicos de mecánica cuántica necesarios; contextualizamos estas herramientas para realizar cómputo con base a lo que llamamos qubits. Para esto introduciremos primero el modelo fundamental de computación, conocido como el modelo de circuito cuántico (Feynman, 1986). Por otra parte, se muestra que existe un pequeño conjunto de compuertas consideradas *universales*, con base a las cuales se puede expresar cualquier circuito cuántico con múltiples qubits.

El modelo de circuito, como lo veremos, consiste en un conjunto de *cables* que representan cada qubit. Sobre estos actúan *compuertas lógicas*; que pueden ser representadas gráficamente de diferentes formas como lo veremos adelante. Las compuertas lógicas pueden actuar sobre uno o múltiples qubits y, con base a estas, se pueden construir circuitos cuyos propósitos varían dependiendo del campo de aplicación. Se han construido diferentes algoritmos para el procesamiento de información, aprendizaje automatizado, y simulación de sistemas físicos como moléculas o sistemas de partículas.

1.5.1. Compuertas de un qubit

Como se introdujo previamente, un qubit puede representarse como en la ecuación (1.1). Las compuertas lógicas que se aplican sobre un qubit no pueden cambiar su normalización; por lo tanto, deben ser operadores unitarios. En el caso de un qubit, las compuertas lógicas se pueden representar como matrices 2×2 . Las operaciones asociadas a los 3 operadores de Pauli son de las más importantes. Las recordamos en su forma matricial

$$\sigma_x = X \doteq \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}; \quad \sigma_y = Y \doteq \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}; \quad \sigma_z = Z \doteq \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}.$$

Tres compuertas más que son muy importantes en cuanto a la construcción y entendimiento de circuitos cuánticos se refiere son la compuerta de Hadamard (H), La compuerta de fase (S) y la compuerta $\pi/8$ (T) (Nielsen y Chuang, 2011),

$$H \doteq \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}; \quad S \doteq \begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix}; \quad T \doteq \begin{pmatrix} 1 & 0 \\ 0 & e^{i\frac{\pi}{4}} \end{pmatrix}.$$

Los qubits pueden ser representados también como puntos en una esfera de radio uno si tomamos $c_0 = \cos(\theta/2)$ y $c_1 = e^{i\phi} \sin(\theta/2)$. Esta esfera es conocida como la esfera de Bloch y se muestra en la figura 1.1.

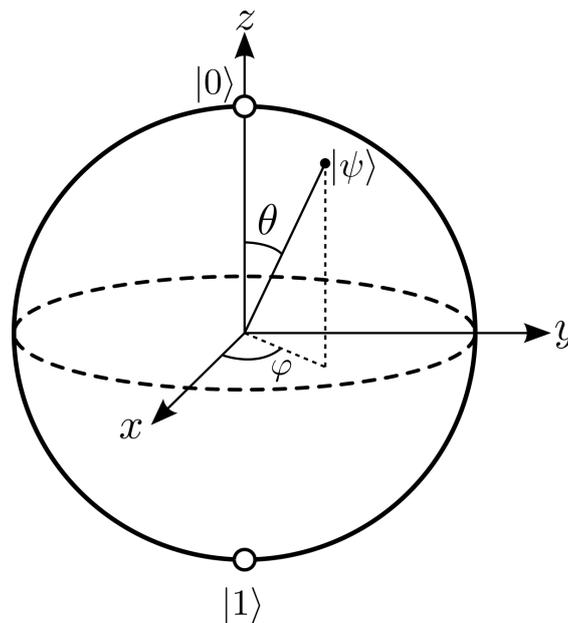


Figura 1.1: Esfera de Bloch

La esfera de Bloch ubica los estados $|0\rangle$ y $|1\rangle$ en sus polos, los cuales corresponden a la base computacional y adicionalmente corresponden a los autoestados del operador Z.

A partir de los operadores de Pauli es posible construir un conjunto de rotaciones sobre los ejes de la esfera de Bloch. Estas rotaciones se construyen al tomar los siguientes operadores exponenciales

$$\begin{aligned} R_x(\phi) &\doteq e^{-i\frac{\phi}{2}X} = \cos\frac{\phi}{2}I - i\sin\frac{\phi}{2}X = \begin{pmatrix} \cos\frac{\phi}{2} & -i\sin\frac{\phi}{2} \\ -i\sin\frac{\phi}{2} & \cos\frac{\phi}{2} \end{pmatrix}, \\ R_y(\phi) &\doteq e^{-i\frac{\phi}{2}Y} = \cos\frac{\phi}{2}I - i\sin\frac{\phi}{2}Y = \begin{pmatrix} \cos\frac{\phi}{2} & -\sin\frac{\phi}{2} \\ \sin\frac{\phi}{2} & \cos\frac{\phi}{2} \end{pmatrix}, \\ R_z(\phi) &\doteq e^{-i\frac{\phi}{2}Z} = \cos\frac{\phi}{2}I - i\sin\frac{\phi}{2}Z = \begin{pmatrix} e^{-i\frac{\phi}{2}} & 0 \\ 0 & e^{i\frac{\phi}{2}} \end{pmatrix}. \end{aligned}$$

Las rotaciones anteriores son rotaciones realizadas alrededor de los ejes definidos en la esfera de Bloch. Con esto ya podemos definir la compuerta U . Esta compuerta se define de la forma

$$U(\theta, \phi, \lambda) \doteq R_z(\phi)R_x(-\frac{\pi}{2})R_z(\theta)R_x(\frac{\pi}{2})R_z(\lambda) \doteq \begin{pmatrix} \cos\frac{\theta}{2} & -e^{-i\lambda}\sin\frac{\theta}{2} \\ e^{i\phi}\sin\frac{\theta}{2} & e^{i(\phi+\lambda)}\cos\frac{\theta}{2} \end{pmatrix}. \quad (1.32)$$

La compuerta U es la más general que se puede expresar para un qubit. De hecho, se puede demostrar que cualquier compuerta de un qubit corresponde a una escogencia particular de los ángulos que definen la compuerta U .

Los circuitos cuánticos, en general, tienen una representación gráfica en la cual las compuertas lógicas de un qubit son encontradas en forma de cajas con un símbolo característico de la compuerta que está actuando sobre el qubit. Las compuertas más importantes y su representación gráficas son mostradas en la figura (1.2).

$$\begin{aligned} \text{Hadamard} & \text{---} \boxed{H} \text{---} & \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \\ \text{Pauli-X} & \text{---} \boxed{X} \text{---} & \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \\ \text{Pauli-Y} & \text{---} \boxed{Y} \text{---} & \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix} \\ \text{Pauli-Z} & \text{---} \boxed{Z} \text{---} & \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \\ \text{Fase} & \text{---} \boxed{S} \text{---} & \begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix} \\ \frac{\pi}{8} & \text{---} \boxed{T} \text{---} & \begin{pmatrix} 1 & 0 \\ 0 & e^{i\pi/4} \end{pmatrix} \end{aligned}$$

Figura 1.2: Representación gráfica de compuertas lógicas en un circuito cuántico.

1.5.2. Compuertas de múltiples qubits

La representación de compuertas anteriormente mostrada se puede extender para representar compuertas de multiples qubits. Por ejemplo, consideremos un par de qubits A y B . Estos qubits son iniciados en el estado $|0\rangle$. Luego, supongamos que se aplica

$$\begin{aligned} |0\rangle_A &: \text{---} \boxed{H} \text{---} |+\rangle_A \\ |0\rangle_B &: \text{---} \boxed{X} \text{---} |1\rangle_B \end{aligned}$$

Figura 1.3: Compuertas H y X aplicadas sobre los qubits A y B respectivamente, representadas en un circuito cuántico. Esta representación se generaliza para más qubits.

una compuerta H sobre el qubit A y una compuerta X (también conocida como compuerta NOT) al qubit B . Esto se puede representar gráficamente como en la figura (1.3)

De acuerdo a las transformaciones vistas previamente, los estados finales de los qubits A y B son $|+\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$ y $|1\rangle$ respectivamente.

El estado global del sistema antes de aplicar las compuertas sería $|\psi\rangle_{AB} = |0\rangle \otimes |0\rangle_B = |0,0\rangle$. Luego, el operador de dos qubits que representa la transformación es $H_A \otimes X_B$, donde las compuertas H_A y X_B corresponden a operadores de un qubit que pueden ser representados con matrices 2×2 como vimos anteriormente; de este modo, el operador $H_A \otimes X_B$ es un operador que matricialmente se puede representar con una matriz 4×4 . El estado final sería entonces $|+\rangle_A \otimes |1\rangle = |+,1\rangle$.

Ahora introduciremos las compuertas controladas. Estas compuertas tienen la forma "Si A es verdadero, realizar B "; y no pueden ser representadas como el producto tensorial de otros operadores. Una compuerta de este tipo es la CNOT (negación controlada). La compuerta CNOT es una compuerta implementada sobre 2 qubits, donde uno de ellos es el *qubit de control* y el *qubit objetivo*. Si el qubit de control se encuentra en el estado $|1\rangle$, se aplica una compuerta NOT (X) al qubit objetivo. Por ejemplo, si el qubit de control es el qubit A y el objetivo es el B , la transformación seguiría de la forma

$$\begin{aligned} |00\rangle &\longrightarrow |00\rangle, \\ |01\rangle &\longrightarrow |01\rangle, \\ |10\rangle &\longrightarrow |11\rangle, \\ |11\rangle &\longrightarrow |10\rangle. \end{aligned}$$

Matricialmente se puede escribir como

$$CNOT \doteq \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}.$$

En un circuito cuántico, se representa como se muestra en la figura (1.4).

$$\begin{array}{l} A : \text{---} \bullet \text{---} \\ B : \text{---} \oplus \text{---} \end{array}$$

Figura 1.4: Representación de una CNOT en un circuito cuántico.

De una forma más general, es posible definir una operación controlada. Supongamos que queremos implementar la compuerta U sobre el qubit B controlada por el qubit A . Si la representación matricial de la compuerta U es de la forma

$$U \doteq \begin{pmatrix} u_{00} & u_{01} \\ u_{10} & u_{11} \end{pmatrix},$$

entonces la representación matricial de esta compuerta tiene la forma

$$CU \doteq \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & u_{00} & u_{01} \\ 0 & 0 & u_{10} & u_{11} \end{pmatrix}.$$

La operación se puede encontrar en un circuito cuántico representada como se muestra en la figura (1.5).

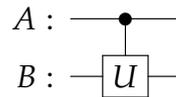
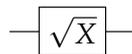


Figura 1.5: Representación gráfica de la operación U controlada por el qubit A .

Con lo anterior ya tenemos los elementos necesarios para definir un conjunto universal de compuertas sobre las que es posible expresar cualquier circuito cuántico de N qubits. Puede haber muchos conjuntos de compuertas para esto; en general, una compuerta controlada y un conjunto de compuertas de un qubit sirven como un conjunto de puertas universal. Por ejemplo, el que usan los procesadores de IBM para expresar sus circuitos consiste en la compuerta CNOT, y las compuertas de un qubit $\{I, R_z(\theta), SX, X\}$ (IBM, s.f.) La compuerta SX o \sqrt{X} se representa matricialmente como

$$SX \doteq \frac{1}{2} \begin{pmatrix} 1+i & 1-i \\ 1-i & 1+i \end{pmatrix},$$

y se representa en un circuito de la forma



Con el conjunto de compuertas $\{CNOT, I, R_z(\theta), SX, X\}$ es posible representar cualquier circuito cuántico.

1.5.3. Medición

La parte final necesaria para completar un circuito es realizar una medición. Muchas veces, el proceso de medición se usa para condicionar compuertas que se implementan en otros qubits posteriores a la medición. Un ejemplo de esto es el protocolo de teleportación. El proceso de medición hecho sobre un qubit en el estado $|\psi\rangle$ en un circuito cuántico, se representa como se muestra en la figura (1.6).



Figura 1.6: Representación gráfica del proceso de medición sobre un qubit en un circuito cuántico.

El proceso de medición se realiza sobre la base computacional. Es importante tener en cuenta que, de acuerdo con teorema de Newmark, una medición generalizada hecha

sobre un qubit se puede expresar como una medición proyectiva hecho en un espacio mayor. En este caso sería necesario introducir qubits adicionales, los cuales se conocen como *qubits de ancila*. Por lo tanto, es posible expresar mediciones generalizadas en un circuito cuántico realizando la transformaciones necesarias previamente.

Los resultados de mediciones en un computador cuántico real muestran los resultados obtenidos en un número N finito de ejecuciones del circuito. Estos resultados dan cuenta de las probabilidades que tienen los qubits de estar en un determinado estado. Por ejemplo, vamos a considerar un sistema de dos qubits que se encuentra en un estado entrelazado de la base de Bell. En particular, estamos interesados en el estado

$$|\psi\rangle_{AB} = \frac{1}{\sqrt{2}} (|00\rangle + |11\rangle).$$

Este estado se puede construir con el circuito de la figura (1.7).

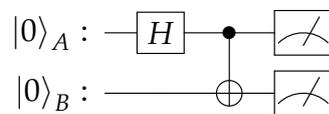


Figura 1.7: Uno de los estados de la base de Bell representado como un circuito cuántico.

Al realizar la medición, se tienen los posibles resultados $\{00, 01, 10, 11\}$ que tienen probabilidades $\{50\%, 0\%, 0\%, 50\%\}$. Los resultados de la medición se muestran como un histograma que registra los conteos obtenidos por cada uno de los posibles resultados. Si realizamos $N = 1024$ repeticiones del experimento, no necesariamente vamos a obtener 512 conteos del resultado 00 y 512 del estado 11. Esto se debe a la varianza de la medición; a medida que el número de repeticiones aumenta, la distribución se acerca más al resultado exacto.

Todos los elementos anteriores son suficientes para entender la lógica del modelo cuántico de circuitos. Los circuitos cuánticos de compuerta universal pueden usarse para diferentes aplicaciones. Para el caso concreto de este trabajo, utilizaremos un circuito cuántico que represente un sistema de tres partículas de espín un medio que interactúan de una forma particular descrita posteriormente para determinar el estado total de espín de una de ellas. En el siguiente capítulo abordaremos teóricamente el modelo que se desea simular en un procesador cuántico real.

1.6. Modelo cuántico de circuito

El concepto de computador cuántico es equivalente al modelo cuántico de circuito para la computación. Este modelo se puede resumir en 5 elementos clave que lo componen

1. **Recursos clásicos:** Aunque este no es un elemento necesario, componentes clásicas en un circuito cuántico pueden ayudar a mejorar su eficiencia. Por ejemplo, en el caso de algoritmos para corrección de errores. Además, los resultados de las mediciones deben ser guardados en registros clásicos.
2. **Un espacio de Hilbert suficientemente grande:** Si un computador cuántico consta de n qubits, entonces se necesita un espacio de Hilbert de dimensión 2^n para representarlos.

3. **Habilidad para preparar estados en la base computacional:** Se asume que es posible preparar cualquier estado de la base computacional de n qubits $|x_1, x_2, \dots, x_n\rangle$ en cuando mucho n pasos.
4. **Posibilidad de implementar compuertas cuánticas:** Como discutimos, un computador cuántico funciona aplicando un conjunto de compuertas lógicas sobre los qubits deseados con algún objetivo. Un computador cuántico debe tener la posibilidad de ejecutar, al menos en una muy buena aproximación, un conjunto de compuertas que funcionen como base para representar cualquier circuito.
5. **Posibilidad de ejecutar mediciones sobre un qubit:** Debe ser posible realizar mediciones en la base computacional sobre uno o más qubits.

Capítulo 2

Tomografía de espín

En este capítulo se describen teóricamente los modelos de estimación parcial (Peres, 1986) y total (Saavedra y Fonseca-Romero, 2019) de las componentes del vector de Bloch que describe el estado cuántico de superposición de un sistema de dos niveles. Luego de realizar la descripción de los modelos, se presenta el concepto de la función de transferencia de tomografía cuántica (qTTF), la cual es el criterio escogido para caracterizar los estimadores del espín. Una vez definido un criterio de caracterización, se realiza la optimización del estimador para elegir el parámetro o parámetros que, independientemente del estado inicial a estimar, tienen un mejor rendimiento al realizar la tarea de estimación bien sea parcial o total del espín.

2.1. Estimación de una componente del espín

En este trabajo hay un particular interés por los sistemas cuánticos de dos niveles, como es el caso del espín de un electrón, la polarización de un fotón y muchos más ejemplos. Esto se debe a que un sistema cuántico de este tipo se puede representar fácilmente utilizando qubits.

Un problema particular para este tipo de sistemas es su estimación. Un sistema cuántico en general se encuentra en una superposición de sus diferentes estados posibles. Para saber cuál es la probabilidad de cada estado en la superposición es necesario realizar medidas sobre el sistema de interés bien sea directa o indirectamente.

En esta sección abordaremos el problema de la estimación de una componente del espín de una partícula de espín un medio que llamamos *Sistema* (S). Esta estimación se puede hacer a través de mediciones indirectas haciendo uso de un estado auxiliar que llamamos metro (A) y que interactúan como se muestra en la figura 2.1.

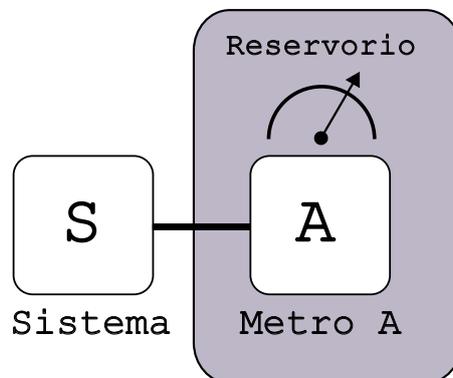


Figura 2.1: Ilustración de la interacción de un sistema de interés S con un metro A para estimar una componente del espín.

Ahora se hace la descripción de la interacción entre el sistema S y el metro A . Para esto, con base al trabajo de Asher Peres (Peres, 1986), se puede definir un Hamiltoniano de interacción entre el sistema y el metro de la forma

$$H \doteq g\Pi_1^S \otimes \Pi_1^A = g|1\rangle\langle 1|^S \otimes |1\rangle\langle 1|^A. \quad (2.1)$$

El parámetro g es un parámetro de acople. En general el parámetro de acople puede depender del tiempo. Sin embargo, el caso de un parámetro constante funciona perfectamente para realizar tomografía de una componente del espín.

El operador de evolución temporal asociado al Hamiltoniano de interacción es entonces

$$U(\theta) \doteq e^{-iHT/\hbar} = e^{-i\theta\Pi_1^S \otimes \Pi_1^A} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & e^{-i\theta} \end{pmatrix}. \quad (2.2)$$

El operador de evolución se ha escrito en la base computacional de dos qubits escrita en orden lexicográfico, es decir, $\{|00\rangle, |01\rangle, |10\rangle, |11\rangle\}$. Además, encontramos el parámetro $\theta = gT/\hbar$; esto suponiendo que la interacción se da durante un tiempo T . Este parámetro, en el trabajo de Peres, toma el valor $\theta = \pi$ Sin embargo, acá tomaremos un ángulo arbitrario y mostraremos por qué esa escogencia permite realizar una estimación más precisa. Dada la forma de definir la interacción y la base computacional que elegimos, el operador de evolución resultante corresponde a una compuerta controlada, por lo cual, la simulación del sistema en términos de un circuito cuántico resulta bastante sencilla. Esta perspectiva la abordaremos en el capítulo siguiente.

Una vez definida la interacción entre el sistema y el metro, podemos construir el esquema para estimar la componente z del vector de espín del sistema S . Esta estimación está asociada al valor esperado del operador de Pauli σ_z . Para este experimento, el sistema S se encuentra en un estado cuántico puro y arbitrario de la forma $|\psi\rangle = c_0|0\rangle + c_1|1\rangle$; es decir, c_0 y c_1 son desconocidos. Por otra parte, se tiene control sobre el estado inicial del metro A . Este metro debe ser iniciado en el estado $|+\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$, que corresponde al autoestado de autovalor $+1$ del operador de Pauli σ_x . De este modo el estado inicial del sistema y el metro ($S + A$) se describe de la forma $|\psi(0)\rangle = |\psi_0^S\rangle \otimes |+\rangle^A$. Luego del tiempo T , el sistema evoluciona de la forma

$$\begin{aligned} |\psi(\theta)\rangle &= U(\theta)|\psi(0)\rangle = \frac{1}{\sqrt{2}} \left(c_0|0^S\rangle|0^A\rangle + c_0|0^S\rangle|1^A\rangle + c_1|1^S\rangle|0^A\rangle + c_1e^{-i\theta}|1^S\rangle|1^A\rangle \right) \\ &= \left(c_0|0^S\rangle + \frac{1}{2}c_1(1 + e^{-i\theta})|1^S\rangle \right) |+\rangle^A + \frac{1}{2}c_1(1 - e^{-i\theta})|1^S\rangle |-\rangle^A \\ &= |d_0\rangle |+\rangle^A + |d_1\rangle |-\rangle^A. \end{aligned}$$

Donde se puede ver de forma clara que $|d_0\rangle = c_0|0^S\rangle + c_1e^{-i\theta/2}\cos(\theta/2)|1^S\rangle$ y $|d_1\rangle = c_1ie^{-i\theta/2}\sin(\theta/2)|1^S\rangle$.

La decisión de escribir el estado final en términos de los estados $|\pm^A\rangle$ no es casualidad. La medición del metro A no se hace en la base computacional si no que se hace en la base de σ_x ; es decir, lo que se medirá son los estados $|\pm^A\rangle$. De este modo, las probabilidades de medir el metro A en los estados $|+\rangle^A$ y $|-\rangle^A$ son, respectivamente,

$$P_0 = \langle d_0|d_0\rangle = |c_0|^2 + |c_1|^2 \cos^2 \frac{\theta}{2}, \quad (2.3)$$

$$P_1 = \langle d_1|d_1\rangle = |c_1|^2 \sin^2 \frac{\theta}{2}. \quad (2.4)$$

Para construir el estimador de s_z , que corresponde a la componente z del vector de Bloch asociado al espín de la sistema, tomamos el valor esperado

$$\begin{aligned}\langle \sigma_z \rangle &= s_z = \langle \psi_0^s | \sigma_z | \psi_0^s \rangle, \\ &= |c_0|^2 - |c_1|^2.\end{aligned}\quad (2.5)$$

Con el resultado anterior y teniendo en cuenta las ecuaciones (2.3) y (2.4) el estimador queda de la forma

$$s_z = \csc^2 \frac{\theta}{2} \left(P_0 - P_1 - \cos^2 \left(\frac{\theta}{2} \right) \right). \quad (2.6)$$

Una vez definido el estimador para s_z , solo necesitamos un montaje experimental a partir del cual, con múltiples mediciones, de puedan estimar P_0 y P_1 . Un escenario perfecto para ser simulado en un computador cuántico.

2.1.1. Elementos POVM para la estimación de s_z

En esta sección deduciremos brevemente los elementos POVM correspondientes a las mediciones del metro A en el estado $|+^A\rangle$ o $|-^A\rangle$. Para esto, se encuentran los operadores de Kraus asociados a la medición de cada metro (Benenti, Casati y Strini, 2004). Teniendo en cuenta que el estado final del sistema tras realizar la medición del metro A en el estado $|+^A\rangle$ es $|d_0\rangle$; reescribimos el estado final de la forma

$$\begin{aligned}|d_0\rangle &= c_0 |0^S\rangle + c_1 e^{-i\theta/2} \cos(\theta/2) |1^S\rangle \\ &= \left(|0^S\rangle \langle 0^S| + e^{-i\frac{\theta}{2}} \cos \frac{\theta}{2} |1^S\rangle \langle 1^S| \right) |\psi_0^S\rangle.\end{aligned}\quad (2.7)$$

De la escritura anterior, identificamos el operador de Kraus M_0 que es

$$M_0 = |0^S\rangle \langle 0^S| + e^{-i\frac{\theta}{2}} \cos \frac{\theta}{2} |1^S\rangle \langle 1^S| = \begin{pmatrix} 1 & 0 \\ 0 & e^{-i\frac{\theta}{2}} \cos \frac{\theta}{2} \end{pmatrix}. \quad (2.8)$$

El operador de medición permite entonces definir el elemento POVM asociado que es

$$E_0 \doteq M_0 M_0^\dagger = \begin{pmatrix} 1 & 0 \\ 0 & \cos^2 \frac{\theta}{2} = \frac{1}{2} (1 + \cos^2 \frac{\theta}{2}) I + \frac{1}{2} \sin^2 \frac{\theta}{2} \sigma_z \end{pmatrix}. \quad (2.9)$$

Una vez definido el elemento POVM ya podemos calcular fácilmente las probabilidades de cada estado final del sistema. En el caso del elemento E_0 se tiene que la probabilidad de tener el estado (2.7) al medir el metro A es

$$\langle \psi_0 | E_0 | \psi_0 \rangle = P_0 = |c_0|^2 + |c_1|^2 \cos^2 \frac{\theta}{2} = \frac{1}{2} \left(1 + \cos^2 \frac{\theta}{2} \right) + \frac{1}{2} \sin^2 \frac{\theta}{2} s_z, \quad (2.10)$$

que corresponde al resultado obtenido en (2.3). El estado final del sistema, luego de la medición de A es

$$\begin{aligned}|\psi_{f0}\rangle &= \frac{E_0}{\sqrt{\langle \psi_0 | E_0 | \psi_0 \rangle}} |\psi_0\rangle \\ &= \frac{1}{\sqrt{P_0}} |d_0\rangle.\end{aligned}$$

De acá es fácil notar que los estados $|d_0\rangle$ y $|d_1\rangle$ en general no son estados normalizados.

Realizando el mismo procedimiento cuando se obtiene el resultado $|-\rangle$ en la medición del metro A ; el estado final se reescribe de la forma

$$\begin{aligned} |d_1\rangle &= c_1 i e^{-i\theta/2} \sin(\theta/2) |1^S\rangle \\ &= \left(i e^{-i\frac{\theta}{2}} \sin \frac{\theta}{2} |1^S\rangle \langle 1^S| \right) |\psi_0\rangle. \end{aligned}$$

Se identifica el operador de Kraus M_1

$$M_1 = i e^{-i\frac{\theta}{2}} \sin \frac{\theta}{2} |1^S\rangle \langle 1^S| = \begin{pmatrix} 0 & 0 \\ 0 & i e^{-i\frac{\theta}{2}} \sin \frac{\theta}{2} \end{pmatrix}. \quad (2.11)$$

Este operador permite definir el elemento POVM asociado a esta medición

$$\begin{aligned} E_1 &\doteq M_1 M_1^\dagger = \begin{pmatrix} 0 & 0 \\ 0 & \sin^2 \frac{\theta}{2} \end{pmatrix} \\ E_1 &\doteq \frac{1}{2} \sin^2 \frac{\theta}{2} I - \frac{1}{2} \sin^2 \frac{\theta}{2} \sigma_z. \end{aligned} \quad (2.12)$$

Con esto es posible encontrar, nuevamente, la probabilidad de esta medición. De forma similar al caso anterior se tiene

$$\langle \psi_0 | E_1 | \psi_0 \rangle = P_1 = |c_1|^2 \sin^2 \frac{\theta}{2} = \frac{1}{2} \sin^2 \frac{\theta}{2} (1 - s_z), \quad (2.13)$$

El cual corresponde al resultado en la ecuación (2.4). El estado final del sistema luego de la medición del metro A en el estado $|-\rangle$ es

$$\begin{aligned} |\psi_{f1}\rangle &= \frac{E_1}{\sqrt{\langle \psi_0 | E_1 | \psi_0 \rangle}} |\psi_0\rangle \\ &= \frac{1}{\sqrt{P_1}} |d_1\rangle. \end{aligned}$$

Ya completado el esquema para la estimación de la componente z del espín con la ayuda de un metro auxiliar solo queda realizar el experimento y hacer la elección del mejor parámetro θ para realizar la estimación. Esta elección se justificará posteriormente con base en el parámetro del error de Fisher; el resultado se obtendrá corresponde al del artículo de Peres (Peres, 1986), $\theta = \pi$. Este estado manifiesta la condición de máximo entrelazamiento entre el qubits S y el metro A ; esto de acuerdo con la medida de entrelazamiento conocida como concurrencia (Wootters, 1998).

2.2. Estimación del estado total de espín

Luego de haber hecho una construcción de un estimador para la componente z del espín de una partícula de espín un medio haciendo uso de un metro auxiliar. Se puede pensar en estimar el estado total de espín haciendo uso de más metros auxiliares siguiendo un modelo similar. De hecho, haciendo uso de un metro auxiliar (Perarnau-Llobet y Nieuwenhuizen, 2017) es posible hacer la estimación de dos componentes. Sin embargo, si se define la interacción adecuada entre el sistema de interés y un par de metros auxiliares, es posible hacer una estimación total del estado de espín de la partícula S . Este modelo es propuesto por Saavedra (Saavedra y Fonseca-Romero, 2019) en el cual, tenemos un sistema de interés que llamamos S y un par de metros adyacentes a este (metros A y B) como se muestra en la figura 2.2.

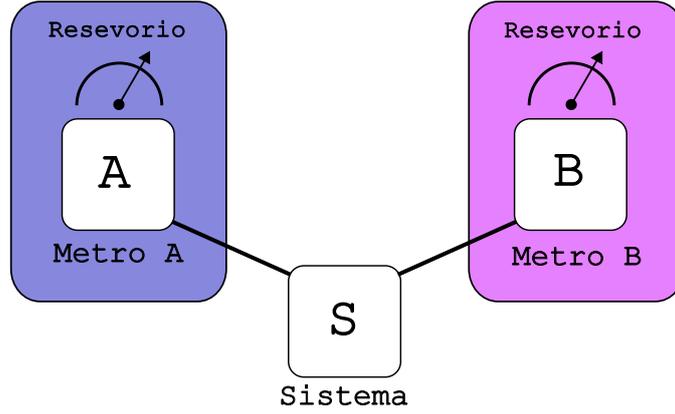


Figura 2.2: Esquema de medición del estado total de espín de la partícula S . Nuevamente, este esquema puede representarse en términos de qubits.

De igual manera al modelo de estimación anterior, los metros A y B son sistemas cuánticos de 2 niveles que pueden representarse como qubits. La interacción Hamiltoniana entre el sistema y los metros se construye inspirada en el modelo de Peres usado anteriormente. En la base computacional, la interacción tiene la forma,

$$H = \lambda_A \Pi_0^S \otimes \Pi_0^A \otimes I^B + \lambda_B \Pi_+^S \otimes I^A \otimes \Pi_0^B. \quad (2.14)$$

Acá, nuevamente, los operadores $\Pi_i^M = |i\rangle \langle i|^M$ corresponden al proyector asociado al estado $i = 0, +$ de la partícula $M = S, A, B$; el operador I^M corresponde al operador identidad del espacio de Hilbert de la partícula M .

En este caso los parámetros λ_A y λ_B son los parámetros de acople en la interacción entre el sistema y el metro correspondiente. En principio estas interacciones dependen del tiempo, pero para los propósitos de este trabajo se pueden tomar como parámetros constantes. De este modo, el operador de evolución asociado en el tiempo T toma la forma

$$U(\theta_A, \theta_B) = e^{-iHT/\hbar} = \exp\left(-i\theta_A \Pi_0^S \otimes \Pi_0^A \otimes I^B - i\theta_B \Pi_+^S \otimes I^A \otimes \Pi_0^B\right), \quad (2.15)$$

que depende de los parámetros $\theta_A = \lambda_A T/\hbar$ y $\theta_B = \lambda_B T/\hbar$.

De forma similar a la estimación de una componente del espín, los metros se inician en el estado $|+\rangle$, mientras que el sistema es iniciado en un estado arbitrario $|\psi_0\rangle = c_0 |0\rangle + c_1 |1\rangle$. El sistema completo luego de la interacción tiene la forma

$$|\Psi_f\rangle = U(\theta_A, \theta_B) |\Psi(0)\rangle = |\psi_0^S\rangle \otimes |+\rangle^A \otimes |+\rangle^B. \quad (2.16)$$

Teniendo en cuenta que $\sum_{i=0,1} |i\rangle \langle i| = I$, el estado final se puede escribir de la forma

$$|\Psi_f\rangle = \sum_{i,j=0,1} U_{ij} |\psi_0^S\rangle |i^A j^B\rangle; \quad (2.17)$$

que corresponde a una escritura diagonal por bloques del operador de evolución, es decir

$$U \doteq \begin{pmatrix} U_{00} & 0 & 0 & 0 \\ 0 & U_{01} & 0 & 0 \\ 0 & 0 & U_{10} & 0 \\ 0 & 0 & 0 & U_{11} \end{pmatrix}.$$

Acá los ceros corresponden a matrices 2×2 llenas de ceros. Los operadores U_{ij} son, tomando $\theta_C = \sqrt{\theta_A^2 + \theta_B^2}$,

$$\begin{aligned} U_{00} &= e^{-i\theta_A(I^S + \sigma_z^S)/2 - i\theta_B(I^S + \sigma_x^S)/2}, \\ &= e^{-i(\theta_A + \theta_B)/2} \left(\cos \frac{\theta_C}{2} I^S - i \frac{\sin \frac{\theta_C}{2}}{\theta_C} (\theta_A \sigma_z^S + \theta_B \sigma_x^S) \right), \\ U_{01} &= e^{-i\theta_A(I^S + \sigma_z^S)/2} = e^{-i\theta_A \Pi_+^S}, \\ U_{10} &= e^{-i\theta_B(I^S + \sigma_x^S)/2} = e^{-i\theta_B/2} \left(\cos(\theta_B/2) I^S - i \sin(\theta_B/2) \sigma_x^S \right), \\ U_{11} &= I^S. \end{aligned}$$

Al igual que en la estimación de s_z , luego de la interacción entre los metros y el sistema, se realiza un cambio en la base de medición de A y B . Cambiando el estado final para medir los metros en la base de σ_x se tiene

$$\begin{aligned} |\Psi_f\rangle &= \frac{1}{2} \sum_{i,j=0,1} U_{ij} |\psi_0^S\rangle \sum_{k,l=\pm} |k^A l^B\rangle \langle k^A l^B | i^A j^B\rangle \\ &= \sum_{k,l=\pm} |d_{kl}\rangle |k^A l^B\rangle, \end{aligned}$$

donde los estados $|d_{kl}\rangle$ corresponden a los estados finales del sistema luego de la medición de los metros

$$|d_{kl}\rangle = \frac{1}{2} \sum_{i,j=0,1} \langle k^A l^B | i^A j^B\rangle U_{ij} |\psi_0^S\rangle. \quad (2.18)$$

Estos estados finales no están normalizados; las probabilidades de obtener cada uno de estos estados son

$$p_{kl} = \langle d_{kl} | d_{kl}\rangle. \quad (2.19)$$

De este modo, en cada medición el estado del sistema queda determinado completamente luego de la medición.

2.2.1. Elementos POVM para la estimación total del espín

De forma similar a la sección anterior, los operadores de Kraus pueden ser definidos a partir del estado final del sistema. Para esto se escribe primero el estado final del sistema de la forma

$$|d_{kl}\rangle = \left(\frac{1}{2} \sum_{i,j=0,1} \langle k^A l^B | i^A j^B\rangle U_{ij} \right) |\psi_0^S\rangle = M_{kl} |\psi_0^S\rangle. \quad (2.20)$$

Los productos $\langle k|i\rangle$ se pueden escribir, teniendo en cuenta que $|k\rangle = \frac{1}{\sqrt{2}} (|0\rangle + k|1\rangle)$, de la forma $\langle k|0\rangle = \frac{1}{\sqrt{2}}$ y $\langle k|-\rangle = \frac{k}{\sqrt{2}}$. Teniendo en cuenta lo anterior, los operadores de Kraus M_{kl} se pueden escribir de la forma

$$M_{kl} = \frac{1}{4} (U_{00} + lU_{01} + kU_{10} + klU_{11}). \quad (2.21)$$

Los elementos POVM son entonces $E_{kl} = M_{kl}^\dagger M_{kl}$, que se reducen a

$$E_{kl} = \frac{1}{16} \sum_{ij} \left[U_{ij}^\dagger U_{i,j} + lU_{ij}^\dagger U_{i,1-j} + kU_{ij}^\dagger U_{1-i,j} + klU_{ij}^\dagger U_{1-i,1-j} \right], \quad (2.22)$$

Como los operadores U_{ij} son unitarios, los elementos POVM pueden escribirse de la forma

$$M_{kl} = \frac{1}{4}I + kA + lB + klC, \quad (2.23)$$

donde

$$A = \frac{1}{16} \sum_{ij} U_{i,j}^\dagger U_{1-i,j}, \quad B = \frac{1}{16} \sum_{ij} U_{i,j}^\dagger U_{i,1-j}, \quad \text{y} \quad C = \frac{1}{16} \sum_{ij} U_{i,j}^\dagger U_{1-i,1-j}.$$

A partir de los operadores de Kraus es posible encontrar también las probabilidades de obtener cada uno de los posibles resultados, que son

$$p_{kl} = \text{Tr} \left[M_{kl} \rho_0^S M_{kl}^\dagger \right] = \text{Tr} \left[E_{kl} \rho_0^S \right], \quad (2.24)$$

En general, el operador densidad inicial ρ_0^S puede ser puro o mezclado. Para el ejemplo propuesto $\rho_0 = |\psi_0^S\rangle \langle \psi_0^S|$.

2.2.2. Construcción del estimador

La estimación de un estado de espín puede hacerse de diferentes maneras. En particular, el método de inversión lineal (Teo, 2015; Matteo Paris, 2004) y el de máxima verosimilitud (Řeháček y col., 2007) serán detallados a continuación.

Para estimar un estado cuántico se necesita disponer de un conjunto suficientemente grande de copias de un sistema arbitrario, en este caso S . Luego, se define un conjunto de observables N que se medirán sobre el conjunto de copias previamente mencionado. Los resultados de las mediciones de estos observables permiten realizar la construcción de un estimador del estado inicial S . Para eso, recordamos que el operador densidad de un estado cuántico puede escribirse en términos de las componentes del vector de Bloch de la forma

$$\rho_0 = \frac{1}{2} \sum_{\mu=0}^3 s_\mu \sigma_\mu. \quad (2.25)$$

Acá $s_0 = 1$ y s_μ , $\mu = 1, 2, 3$, corresponden a las componentes del vector de Bloch $\vec{s} = (s_x, s_y, s_z)$ que describe el estado ρ en las direcciones x, y, z respectivamente. Adicionalmente, los operadores $\sigma_0 = I$ y σ_μ , $\mu = 1, 2, 3$ corresponden a los operadores de Pauli $\sigma_x, \sigma_y, \sigma_z$ respectivamente. El vector de Bloch cumple que $|\vec{s}| \leq 1$; la igualdad se cumple en el caso de que el estado descrito por el vector sea puro. La representación gráfica de este vector corresponde a la descrita en la sección 1.1 en la figura 1.1.

Para este caso, el conjunto de observables que permiten estimar el vector de Bloch del sistema corresponde a los que se obtienen a partir de los elementos POVM definidos en la ecuación (2.23). Teniendo en cuenta la escritura explícita de las probabilidades p_{kl} en la ecuación (2.24) y que el estado inicial puede escribirse como en la ecuación (2.25), las probabilidades de medición pueden escribirse en términos de los operadores POVM y las componentes del vector de Bloch de la forma

$$p_{kl} = \frac{1}{4} s_0 + \sum_{\mu=0}^3 (a_\mu k + b_\mu l + c_\mu kl) s_\mu. \quad (2.26)$$

Los coeficientes a_μ, b_μ y c_μ están dados por

$$a_\mu = \frac{1}{2} \text{Tr} [A\sigma_\mu], \quad (2.27)$$

$$b_\mu = \frac{1}{2} \text{Tr} [B\sigma_\mu], \quad (2.28)$$

$$c_\mu = \frac{1}{2} \text{Tr} [C\sigma_\mu]; \quad (2.29)$$

Las expresiones explícitas de estos coeficientes se encuentran en el apéndice A.

Definiendo el vector de probabilidades $\mathbf{P} = (p_{00}, p_{01}, p_{10}, p_{11})^T$, y el vector del estado inicial de espín $\mathbf{S} = (s_0, s_1, s_2, s_3)^T = (s_0, \vec{s})^T$, la ecuación (2.26) se escribe matricialmente de la forma $\mathbf{P} = \mathbb{T}\mathbf{S}$ donde \mathbb{T} es una matriz que se define implícitamente en la ecuación (2.26) y se conoce como la *matriz de medición*. Esta matriz depende del montaje experimental que se usa para realizar la estimación; en este caso depende de la interacción entre el sistema S y los metros A y B . Escribiendo explícitamente esta igualdad se tiene

$$\begin{pmatrix} p_{++} \\ p_{+-} \\ p_{-+} \\ p_{--} \end{pmatrix} = \begin{pmatrix} \frac{1}{4} + a_0 + b_0 + c_0 & a_1 + b_1 + c_1 & a_2 + b_2 + c_2 & a_3 + b_3 + c_3 \\ \frac{1}{4} + a_0 - b_0 - c_0 & a_1 - b_1 - c_1 & a_2 - b_2 - c_2 & a_3 - b_3 - c_3 \\ \frac{1}{4} - a_0 + b_0 - c_0 & -a_1 + b_1 - c_1 & -a_2 + b_2 - c_2 & -a_3 + b_3 - c_3 \\ \frac{1}{4} - a_0 - b_0 + c_0 & -a_1 - b_1 + c_1 & -a_2 - b_2 + c_2 & -a_3 - b_3 + c_3 \end{pmatrix} \begin{pmatrix} 1 \\ s_1 \\ s_2 \\ s_3 \end{pmatrix}.$$

Por lo tanto, para estimar las componentes del vector \mathbf{S} es necesario que la matriz \mathbb{T} sea invertible. En caso de cumplirse esto, las componentes del vector de Bloch se pueden escribir de la forma

$$\mathbf{S} = \mathbb{T}^{-1}\mathbf{P}. \quad (2.30)$$

Por lo tanto, con el montaje experimental que represente la interacción en la ecuación (2.15) y realizando suficientes mediciones para estimar el vector de probabilidades \mathbf{P} , podemos realizar la estimación del vector de espín que describe el sistema S de interés.

Para este caso, el montaje experimental se va a simular a través de un circuito cuántico. Por otra parte las mediciones que se realizarán para estimar \mathbf{P} dependen del simulador utilizado; en general un circuito cuántico se suele ejecutar 1024 veces para obtener una estimación acertada de los resultados.

2.3. Caracterización del estimador

Ya se ha construido un modelo teórico que funciona para realizar la estimación parcial y total de las componentes del vector de Bloch que describe el estado inicial de espín de una partícula con espín un medio. Sin embargo, hay diferentes estimadores posibles; de hecho, hay infinitos valores para los parámetros del operador de evolución que se pueden elegir y que permiten realizar tomografía. En esta sección se va a ilustrar un criterio que permite juzgar el rendimiento de un estimador; para esto es necesario introducir el parámetro de error de Fisher (Lehmann y Cassella, 1998; Kay, 1993).

Como vimos en la sección anterior, ya hemos construido un estimador para la componente s_z del espín, en la ecuación (2.6), y para el vector total de espín, ecuación (2.30). En los dos casos, los estimadores dependen de las mediciones de los metros y los parámetros que median la interacción entre el metro y el sistema. Las mediciones de los metros se realizan para realizar la estimación de los parámetros p_k en cualquiera de los dos casos. Con base a estas estimaciones se construye el estimador de las componentes

de espín.

Consideramos que el circuito cuántico que simula el problema se va a ejecutar N veces. Luego, se tendrán N eventos independientes de detección en los metros asociados al montaje; cada una de las posibilidades se detectará n_k veces formando así un conjunto de detecciones $\mathbf{n} = \{n_1, \dots, n_m\}$. Para el caso de estimación de la componente s_z , $m = 2$; en el caso de la estimación total del espín, $m = 4$, donde $\sum_k^m n_k = N$. El conjunto de mediciones tiene una distribución multinomial de la forma

$$P(\mathbf{n}|\mathbf{P}) = N! \prod_{r=1}^m \frac{1}{n_r!} [p_r]^{n_r}. \quad (2.31)$$

Esta distribución se debe a que, al construir el circuito, las probabilidades de medición de cada uno de los posibles estados de los metros tiene una probabilidad fija. De esta forma, el estimador de las componentes p_k , con base a los resultados de la medición, toma la forma

$$\hat{p}_k = \frac{n_k}{N}. \quad (2.32)$$

Acá la notación \hat{p} se usa para diferenciar los valores teóricos de sus estimaciones; en este caso es claro que si $N \rightarrow \infty \Rightarrow \hat{p} \rightarrow p_k$.

Una vez definido el estimador \hat{p}_k se construye el estimador para las componentes del vector de espín

$$\hat{s}_z = \csc^2 \frac{\theta}{2} \left(\hat{p}_0 - \hat{p}_1 - \cos^2 \left(\frac{\theta}{2} \right) \right), \quad (2.33)$$

$$\hat{s}_\mu = \sum_{k=0}^4 \left[\mathbb{T}^{-1}(\theta_A, \theta_B) \right]_{\mu k} \hat{p}_k. \quad (2.34)$$

Para la estimación de una y tres componentes del espín respectivamente; es importante notar que los estimadores dependen de un conjunto de parámetros Θ , que en el caso de la estimación de s_z son $\Theta = \{\theta\}$ y en el caso de la estimación total $\Theta = \{\theta_A, \theta_B\}$.

Como los estimadores \hat{p}_k son variables aleatorias, los estimadores \hat{s}_μ también lo son. De este modo, se puede asociar al estimador un sesgo y un error cuadrático medio, definidos respectivamente de la forma (Kay, 1993)

$$b(\hat{s}_\mu) = \mathbb{E}[\hat{s}_\mu] - s_\mu \quad (2.35)$$

y

$$\text{ECM}(\hat{s}_\mu) = \mathbb{E} \left[(\hat{s}_\mu - s_\mu)^2 \right]. \quad (2.36)$$

El símbolo $\mathbb{E}[\cdot]$ se refiere al valor esperado de la variable aleatoria en cuestión. Un buen estimador de espín debería entonces tener valores lo más pequeños posibles del sesgo y del error cuadrático medio.

Para el cálculo del sesgo del estimador debemos calcular primero $\mathbb{E}[\hat{s}_\mu]$. Para el caso de los estimadores construidos anteriormente se tiene que el valor esperado es

$$\mathbb{E}[\hat{s}_\mu] = \mathbb{E} \left[\sum_{k=0}^4 \left[\mathbb{T}^{-1}(\Theta) \right]_{\mu k} \hat{p}_k \right] = \sum_{k=0}^4 \left[\mathbb{T}^{-1}(\Theta) \right]_{\mu k} \mathbb{E}[\hat{p}_k]$$

Cuando se tiene una distribución de probabilidad multinomial, como es el caso de los dos estimadores presentados en este trabajo, es $\mathbb{E}[\hat{p}_\mu] = p_\mu$ (Lehmann y Cassella, 1998), por lo tanto

$$\mathbb{E}[\hat{s}_\mu] = s_\mu,$$

Lo cual ocurre para los dos estimadores, por lo tanto $b(\hat{s}_\mu) = 0$.

Cuando el sesgo de un estimador es nulo se le denomina un estimador *insesgado*. Esto implica que el rendimiento de los estimadores recae completamente en sus errores cuadráticos medios.

Como $\mathbb{E}[\hat{s}_\mu] = s_\mu$ entonces el error cuadrático medio puede escribirse de la forma $\text{ECM}(\hat{s}_\mu) = \mathbb{E}[(\hat{s}_\mu - \mathbb{E}[\hat{s}_\mu])^2]$. Esta forma del ECM corresponde a la varianza del estimador \hat{s}_μ

$$\text{Var}(\hat{s}_\mu) = \mathbb{E}[(\hat{s}_\mu - \mathbb{E}[\hat{s}_\mu])^2]. \quad (2.37)$$

Esta definición de la varianza corresponde también a partir de la matriz de covarianza de los estimadores. Las componentes de esta matriz son

$$\text{Cov}(\hat{s}_\mu, \hat{s}_\nu) = \mathbb{E}[(\hat{s}_\mu - \mathbb{E}[\hat{s}_\mu])(\hat{s}_\nu - \mathbb{E}[\hat{s}_\nu])]. \quad (2.38)$$

Por lo tanto, las varianzas corresponden a los elementos de la diagonal de la matriz de covarianza.

Con esta definición, podemos construir casos de medición ideales para encontrar el conjunto de parámetros que optimizan el rendimiento del estimador. Para realizar esto debemos introducir el parámetro del error de Fisher, el cual está íntimamente relacionado con las varianzas del estimador y con base al cual se puede establecer un criterio para indicar que estimador es mejor.

2.3.1. Parámetro de error de Fisher

Ya definida la matriz de covarianza de los estimadores, dados que estos son insesgados y son estimados a partir de distribuciones multinomiales, la matriz de covarianza cumple con la cota de Cramér-Rao (Kay, 1993).

$$\text{Cov}(\hat{\mathbf{s}}, \hat{\mathbf{s}}) \geq \mathbb{J}^{-1}(\mathbf{s}), \quad (2.39)$$

con \mathbb{J} conocida como la matriz de información de Fisher. Las componentes de la matriz de Fisher se calculan a partir de la relación (Teo, 2015)

$$J_{\mu\nu}(\mathbf{s}) = -\mathbb{E}\left[\frac{\partial^2 \ln[P(\mathbf{n}|\mathbf{P})]}{\partial s_\mu \partial s_\nu}\right]. \quad (2.40)$$

Donde $P(\mathbf{n}|\mathbf{P})$ corresponde a la distribución multinomial de la ecuación (2.31). Utilizando la regla de la cadena y teniendo en cuenta que $\frac{\partial}{\partial s_\mu} = \sum_k \frac{\partial p_k}{\partial s_\mu} \frac{\partial}{\partial p_k}$ se tiene

$$\begin{aligned} J_{\mu\nu}(\mathbf{s}) &= -\mathbb{E}\left[\left(\sum_{k=1}^m \frac{\partial p_k}{\partial s_\mu} \frac{\partial}{\partial p_k}\right) \left(\sum_{l=1}^m \frac{\partial p_l}{\partial s_\nu} \frac{\partial \ln[P(\mathbf{n}|\mathbf{P})]}{\partial p_l}\right)\right], \\ &= -\sum_{k,l=1}^m \frac{\partial p_k}{\partial s_\mu} \frac{\partial p_l}{\partial s_\nu} \mathbb{E}\left[\frac{\partial^2 \ln[P(\mathbf{n}|\mathbf{P})]}{\partial p_k \partial p_l}\right]. \end{aligned}$$

Teniendo en cuenta que

$$\begin{aligned} \frac{\partial \ln[P(\mathbf{n}|\mathbf{P})]}{\partial p_k} &= \frac{1}{P(\mathbf{n}|\mathbf{P})} \frac{\partial [P(\mathbf{n}|\mathbf{P})]}{\partial p_k}, \\ &= \frac{1}{P(\mathbf{n}|\mathbf{P})} N! \prod_{r=1}^m \frac{1}{n_r!} \frac{n_k}{p_k} [p_r]^{n_r}, \\ &= \frac{1}{P(\mathbf{n}|\mathbf{P})} P(\mathbf{n}|\mathbf{P}) \frac{n_k}{p_k}. \end{aligned}$$

Entonces $\frac{\partial \ln[P(\mathbf{n}|\mathbf{P})]}{\partial p_k} = \frac{n_k}{p_k}$, por lo tanto,

$$\begin{aligned} J_{\mu\nu}(\mathbf{s}) &= - \sum_{k,l=1}^m \frac{\partial p_k}{\partial s_\mu} \frac{\partial p_l}{\partial s_\nu} \mathbb{E} \left[-\frac{n_k}{p_k^2} \right] \delta_{kl}, \\ &= \sum_{k,l=1}^m \frac{\partial p_k}{\partial s_\mu} \frac{\partial p_l}{\partial s_\nu} \left[\frac{N p_k}{p_k^2} \right] \delta_{kl}. \end{aligned}$$

Finalmente las componentes de la matriz de Fisher son

$$J_{\mu\nu}(\mathbf{s}) = N \sum_{k=1}^m \frac{1}{p_k} \frac{\partial p_k}{\partial s_\mu} \frac{\partial p_k}{\partial s_\nu}. \quad (2.41)$$

Se puede ver que las componentes de la matriz de Fisher disminuyen cuando $N \rightarrow \infty$, de hecho, dado que el estimador es asintóticamente eficiente esto implica a su vez que

$$\lim_{N \rightarrow \infty} \text{Cov}(\hat{\mathbf{s}}, \hat{\mathbf{s}}) = \mathbb{J}^{-1}(\mathbf{s}),$$

por lo cual con un número suficientemente grande de ejecuciones del circuito es posible realizar una buena estimación. Sin embargo, nos interesa caracterizar el estimador independientemente del número de ejecuciones que se hagan, por lo tanto se define ahora la matriz \mathbb{F} con base a la matriz de Fisher por la relación

$$\mathbb{J}^{-1} = \frac{1}{N} \mathbb{F}^{-1}, \quad (2.42)$$

cuyas componentes son entonces

$$F_{\mu\nu} = \sum_{k=1}^m \frac{1}{p_k} \frac{\partial p_k}{\partial s_\mu} \frac{\partial p_k}{\partial s_\nu}. \quad (2.43)$$

La matriz \mathbb{F} depende únicamente del estado a estimar y el montaje experimental asociado para realizar la estimación; por esto se toma la traza de la inversa de esta matriz para caracterizar el error. A este parámetro se le nombra *parámetro de error de Fisher* y tiene la forma

$$\Delta(\Theta, \mathbf{s}) = \text{Tr} \left[\mathbb{F}^{-1}(\Theta, \mathbf{s}) \right]. \quad (2.44)$$

Ahora, con base al parámetro de error de Fisher podemos caracterizar el rendimiento de un estimador independientemente del número de muestras N que se tengan para realizar la tomografía del estado de espín. Para esto introducimos la función de transferencia de tomografía cuántica qTTF (quantum tomographic transfer function) (Řeháček, Teo y Hradil, 2015). En este trabajo, la qTTF se define como la media del parámetro del error de Fisher respecto a los posibles estados iniciales del sistema, la cual depende de los parámetros del montaje para realizar la estimación de las componentes de espín. En este caso, los posibles estados iniciales del sistema se pueden escribir en términos de un conjunto de parámetros ξ ; por lo tanto se tiene $\mathbf{s} = \mathbf{s}(\xi)$. La qTTF se define de la forma

$$qTTF(\Theta) = \frac{1}{V} \int_{\xi} \Delta(\Theta, \mathbf{s}(\xi)) d\xi. \quad (2.45)$$

Donde $V = \int_{\xi} d\xi$.

Una vez definida la qTTF, debemos minimizar su valor para así encontrar el mejor estimador, por lo cual, el procedimiento cambia para cada uno de los estimadores definidos. Una vez se encuentran los mejores valores para los estimadores, estos se usarán para realizar las simulaciones de los correspondientes circuitos cuánticos que representen el problema.

2.3.2. Error del estimador de s_z

Una vez construido un criterio que permite caracterizar el rendimiento de un estimador para las componentes de espín, se pasa a seleccionar el mejor estimador en cada caso. En primero lugar consideramos el estimador de la componente s_z del espín. Primero tomamos el valor del parámetro de Fisher asociado a este operador. Para esto, tomamos la ecuación (2.43); en este caso $m = 2$ ya que hay 2 posibles resultados de esta medición, como se puede ver en la ecuación (2.6). Por otra parte, los índices μ, ν son los que están relacionados a las componentes de espín que se van a estimar. En este caso solo se tiene s_0 y s_z , sin embargo, tomamos la normalización $s_0 = 1$. Esto hace que solo se considere el caso $\mu = \nu = 3$, lo que implica que la matriz de Fisher solamente tiene una componente que aporta la información necesaria para caracterizar el estimador. Las derivadas $\partial p_k / \partial s_\mu$ se calculan a partir de las probabilidades encontradas con los elementos POVM, es decir las ecuaciones (2.10) y (2.13). Además, el conjunto de parámetros que caracteriza el estimador en este caso solo corresponde a $\Theta = \{\theta\}$. Recordando que lo que nos interesa es el valor inverso de este parámetro, se tiene

$$F^{-1}(\theta, \xi) = \frac{4p_0(\xi)p_1(\xi)}{\sin^4 \frac{\theta}{2}}. \quad (2.46)$$

Es importante recordar que F^{-1} cumple la desigualdad de Cramér-Rao. Es decir, F^{-1} es la cota inferior de la varianza del estimador para s_z . Por lo tanto, en el límite $N \rightarrow \infty$, se debe cumplir que

$$\lim_{N \rightarrow \infty} \text{Var}(s_z) = \frac{1}{N} F^{-1}(\theta, \xi). \quad (2.47)$$

En el apéndice (B) se muestra dicha igualdad. Una vez calculado el parámetro de error de Fisher, nos interesa estudiar su dependencia con el estado inicial del sistema. Para esto es necesario parametrizar las constantes c_0 y c_1 que definen un estado inicial. La parametrización usada en este trabajo para un sistema de dos niveles (Tilma y Sudarshan, 2002) está dada por los parámetros $\xi = \{\alpha_1, \alpha_2, \alpha_3\}$, donde $\alpha_1 \in [0, \pi/2]$, y $\alpha_2, \alpha_3 \in [0, \pi]$, y parametrizan el estado inicial siguiendo

$$c_0 = e^{i\alpha_3 + i\alpha_2} \cos \alpha_1, \quad c_1 = e^{i\alpha_3 - i\alpha_2} \sin \alpha_1.$$

Para el análisis del parámetro del error, podemos sacar el término asociado a la fase global $e^{i\alpha_3}$. Así los parámetros $\xi = \{\alpha_1, \alpha_2\}$ representan el estado inicial del sistema de la forma

$$c_0 = e^{i\alpha_2} \cos \alpha_1, \quad c_1 = e^{-i\alpha_2} \sin \alpha_1. \quad (2.48)$$

El diferencial de volumen de esta parametrización es $dV = \sin(2\alpha_1) d\alpha_1 d\alpha_2$, por lo cual el volumen asociado al espacio de los parámetros ξ es

$$\int_0^{\frac{\pi}{2}} \sin(2\alpha_1) d\alpha_1 \int_0^\pi d\alpha_2 = \pi.$$

Al utilizar la parametrización anterior, el parámetro del error toma la forma

$$F^{-1}(\theta, \alpha_1, \alpha_2) = 4 \sin^2(\alpha_1) (\csc^2(\theta/2) - \sin^2(\alpha_1)).$$

Es importante notar que la dependencia con respecto al parámetro α_2 desaparece. Esto se debe a que el parámetro α_2 está relacionado con la fase relativa entre los estados $|0\rangle$ y $|1\rangle$; una fase relativa diferente no afecta el valor esperado del espín en dirección z .

Una vez definido el parámetro del error graficamos su comportamiento respecto al estado, es decir, respecto al parámetro α_1 . Esto para diferentes valores del parámetro θ

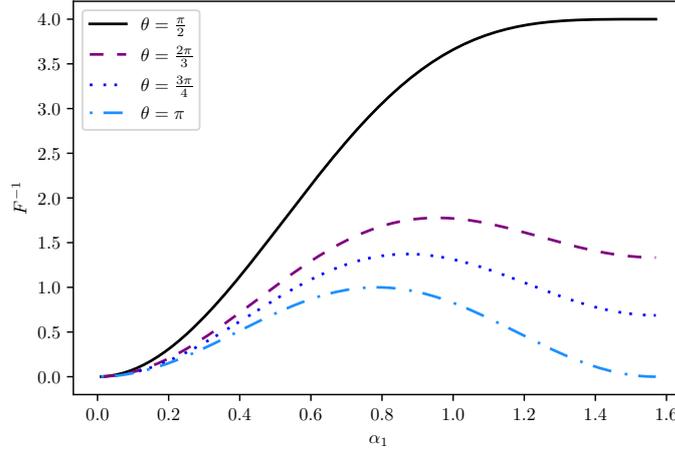


Figura 2.3: Parámetro del error F^{-1} como función de α_1 para los valores $\theta = \frac{\pi}{2}, \frac{2\pi}{3}, \frac{3\pi}{4}$ y π . Como se puede ver, el error decrece cuando $\theta \rightarrow \pi$.

que caracteriza el estimador. En la figura 2.3 se puede ver el comportamiento del parámetro de error de Fisher en función de α_1 para diferentes valores de θ .

Del gráfico anterior ya es posible concluir que el valor $\theta = \pi$ es el que tiene un menor error para la estimación de s_z ; es claro que el área bajo la curva (que es proporcional al valor medio) es menor para este parámetro, por lo cual, es el valor más óptimo para el estimador. Otro aspecto importante que podemos notar en el gráfico es que siempre tiene un valor máximo y que el error en la estimación no es uniforme para cualquier estado inicial. Dependiendo del estado que se quiere estimar puede haber un mayor o menor error en dicha estimación.

El valor máximo, dado un valor de θ fijo satisface

$$\frac{\partial F^{-1}}{\partial \alpha_1} = 4 \sin(2\alpha_1) (\csc^2(\theta/2) - 2 \sin^2 \alpha_1) = 0. \quad (2.49)$$

Si $0 \leq \theta \leq \pi/2$ el valor máximo del error se da siempre en $\alpha_1 = \pi/2$ donde $F^{-1} = 4(\csc^2(\theta/2) - 1)$. Por otra parte, cuando $\pi/2 \leq \theta \leq \pi$, el valor máximo del error cambia con el parámetro θ y se da cuando $\alpha_1 = \arcsin(\frac{1}{\sqrt{2}} \csc(\theta/2))$. En este caso el valor del parámetro del error es $F^{-1} = \csc^4(\theta/2)$. Además, se puede ver que hay valores de α_1 para los cuales el valor del parámetro del error se anula.

Para interpretar mejor estos resultados se hace uso del concepto de entropía de von Neumann (Benenti, Casati y Strini, 2004). Es importante tener en cuenta que el estado asociado al sistema y el metro luego de la interacción es puro y se puede escribir de la forma

$$|\psi_{SA}\rangle = \frac{1}{\sqrt{2}} \left(c_0 |0\rangle |0\rangle + c_0 |0\rangle |1\rangle + c_1 |1\rangle |0\rangle + c_1 e^{-i\theta} |1\rangle |1\rangle \right).$$

Acá el primer índice del ket se refiere al sistema mientras que el segundo valor corresponde al metro. El entrelazamiento puede ser cuantificado con base a la entropía bien sea del operador densidad asociado al sistema $\rho_S = \text{Tr}_A(|\psi_{SA}\rangle \langle \psi_{SA}|)$, o al operador densidad del metro $\rho_A = \text{Tr}_S(|\psi_{SA}\rangle \langle \psi_{SA}|)$.

En este caso se toma ρ_S , por lo cual, la entropía de Von Neumann del sistema luego de la interacción con el metro está dada por

$$E(\rho_S) = -\text{Tr}(\rho_S \log \rho_S) = -\lambda_+ \log \lambda_+ - \lambda_- \log \lambda_-, \quad (2.50)$$

donde $\lambda_{\pm} = (1/2)(1 \pm \sqrt{x})$ y $x = 1 - \sin^2(2\alpha_1) \sin^2(\theta/2)$. En la figura 2.4 se muestra el comportamiento de la entropía respecto al parámetro α_1 del sistema.

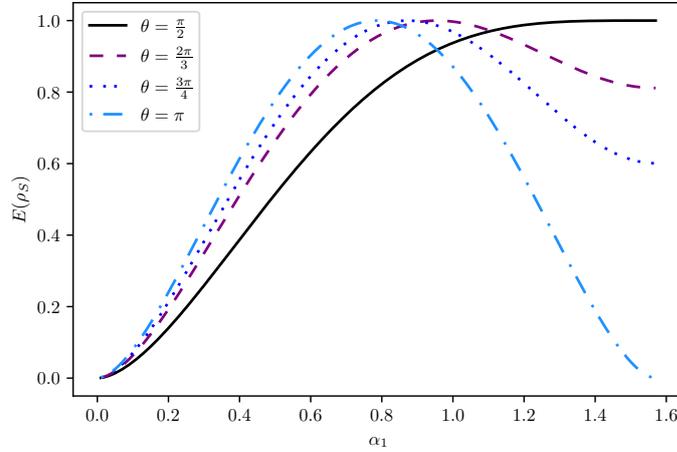


Figura 2.4: Entropía del sistema $E(\rho_S)$ como función de α_1 para los valores $\theta = \frac{\pi}{2}, \frac{2\pi}{3}, \frac{3\pi}{4}$ y π . Como se puede ver, el comportamiento de la entropía es similar al del parámetro del error presentado en la figura 2.3.

Aquí se puede ver que la entropía, no solo tiene un comportamiento cualitativamente similar al parámetro del error F^{-1} ; sino que a su vez, se anula para las mismas combinaciones de los parámetros α_1 y θ .

En los casos $\alpha = 0$ para cualquier θ , el error y la entropía se hacen cero; esto pasa porque, con $\alpha_1 = 0$ se tiene que el estado inicial, antes de la interacción (2.2) es $|0^S\rangle | +^A\rangle$. Este estado queda invariante después de la interacción, por lo tanto, se tiene total certeza de que cuando se realice la medición del metro A solo se va a encontrar un posible valor, es decir, no hay incertidumbre en el posible valor que va a tomar el metro después de la medición. Por otra parte, cuando el estado inicial es $\alpha_1 = \frac{\pi}{2}$ y $\theta = \pi$, el estado inicial del sistema es $|1^S\rangle | +^A\rangle$, luego de la interacción (2.2) el estado pasa a ser $|1^S\rangle | -^A\rangle$. Acá, nuevamente, se tiene total certeza de que las mediciones sobre el metro A siempre corresponderán a un único estado, lo cual se puede interpretar por el hecho de que la entropía es cero.

De la misma forma se pueden interpretar los resultados del parámetro del error F^{-1} ; Para las combinaciones de α y θ previamente discutidas se tiene también que el error es cero. En esos casos el error se anula por el hecho de que, al haber una sola medición posible, se puede estimar con total certeza el valor esperado de σ_z .

Ahora el interés es caracterizar el estimador únicamente respecto al parámetro del que este depende, es decir, θ . Para esto tomamos el valor medio de F^{-1} . El valor medio corresponde también a la qTTF definida anteriormente en la ecuación (2.45) y queda de la forma

$$qTTF = \overline{F^{-1}}(\theta) = \frac{1}{\pi} \int_0^{\pi/2} d\alpha_1 \int_0^{\pi} d\alpha_2 F^{-1}(\alpha_1, \alpha_2, \theta) \sin(2\alpha_1) \quad (2.51)$$

$$qTTF = \frac{2}{3} (2 + \cos \theta) \csc^2 \frac{\theta}{2}$$

La función anterior permite caracterizar el estimador únicamente respecto al parámetro θ teniendo en cuenta su contribución media para la estimación de cualquier estado. En la figura 2.5 se muestra el comportamiento de la qTTF asociada al estimador de s_z .

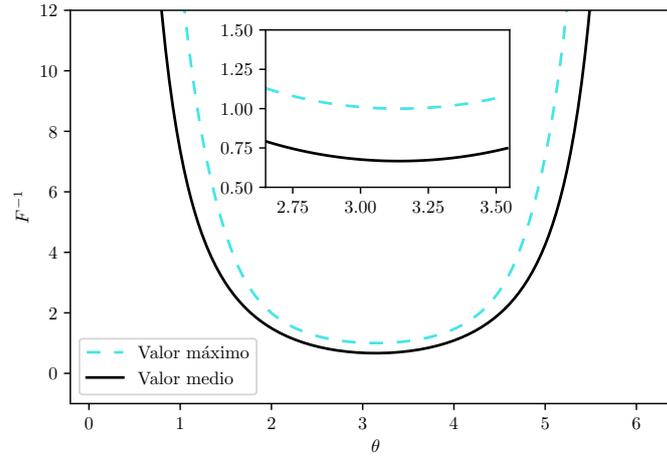


Figura 2.5: Valor medio del parámetro del error $\overline{F^{-1}}(\theta)$ (línea sólida) y valor máximo del error F_{max}^{-1} (línea discontinua) como funciones del parámetro θ .

El resultado corresponde a una función monóticamente decreciente en el intervalo $\theta \in (0, \pi]$ que alcanza su valor mínimo en $\theta = \pi$. Nuevamente, se justifica la escogencia de $\theta = \pi$ como el valor óptimo del estimador.

Además de esto, la figura 2.5 muestra la divergencia del valor medio del error cuando $\theta \rightarrow \{0, 2\pi\}$, lo cual indica que el estimador pierde su eficiencia con estos valores. Esto se da porque para esos valores el operador de evolución en la ecuación (2.2) tiene una acción nula sobre el conjunto sistema-metro; en otras palabras, no hay interacción. Si no hay interacción entre el sistema y el metro no es posible estimar propiedades del sistema a partir de una medición realizada sobre el metro.

De acá se hace evidente que el entrelazamiento juega un papel importante en cuanto a la calidad del estimador. El poder de entrelazamiento de un operador de evolución U respecto a una medida de entrelazamiento E tiene la forma (Zanardi, Zalka y Faoro, 2000)

$$e_p(U) = \overline{E(U|\psi_1\rangle \otimes |\psi_2\rangle)}^{\psi_1, \psi_2}, \quad (2.52)$$

donde el promedio se toma sobre todos los posibles estados puros y separables. Para el caso del operador de evolución $U(\theta)$ de la ecuación (2.2), el poder de entrelazamiento se halla como el de una compuerta controlada (Rezakhani, 2004). Para el caso específico se tiene que

$$e_p(U(\theta)) = \frac{1}{9} (1 - \cos \theta). \quad (2.53)$$

Cuando θ varía entre 0 y π , el poder de entrelazamiento varía desde su valor mínimo 0 hasta su valor máximo $2/9$. Es importante ver que la qTTF $\overline{F^{-1}}(\theta)$ disminuye a medida que el poder de entrelazamiento de la compuerta $U(\theta)$ aumenta. Cuando el poder de entrelazamiento es máximo, la qTTF alcanza su valor óptimo.

2.3.3. Error del estimador total de espín

En este caso realizamos el mismo procedimiento que se realizó para el estimador de s_z . Se toma la ecuación (2.43) donde $m = 4$ ya que hay 4 posibles resultados de la medición de los 2 metros como se muestra en la ecuación (2.26). Los índices μ, ν son los que se relacionan con las derivadas respecto a las componentes de espín. Acá, nuevamente, se considera la normalización $s_0 = 1$, por lo cual las componentes que tienen un aporte al derivar son $\mu, \nu = 1, 2, 3$. De este modo, la Matriz de Fisher es una matriz 4×4 que otorga la información suficiente para caracterizar el estimador total de espín. Para esto, nuevamente, consideramos las probabilidades de medición de cada uno de los 4 estados posibles de la ecuación (2.26). El conjunto de parámetros que caracteriza el estimador construido en la sección 2.2 es $\Theta = \{\theta_A, \theta_B\}$. Para el cálculo de la matriz de información de Fisher reescribimos las probabilidades p_{kl} de la forma

$$p_{kl} = \left(\frac{1}{4} + a_0 k + b_0 l + c_0 kl \right) s_0 + \sum_{\mu=1}^3 (a_\mu k + b_\mu l + c_\mu kl) s_\mu.$$

Las cuatro ecuaciones asociadas a las diferentes probabilidades son

$$\begin{aligned} p_{00} &= \left(\frac{1}{4} + a_0 + b_0 + c_0 \right) s_0 + \sum_{\mu=1}^3 (a_\mu + b_\mu + c_\mu) s_\mu, \\ p_{01} &= \left(\frac{1}{4} + a_0 - b_0 - c_0 \right) s_0 + \sum_{\mu=1}^3 (a_\mu - b_\mu - c_\mu) s_\mu, \\ p_{10} &= \left(\frac{1}{4} - a_0 + b_0 - c_0 \right) s_0 + \sum_{\mu=1}^3 (-a_\mu + b_\mu - c_\mu) s_\mu, \\ p_{11} &= \left(\frac{1}{4} - a_0 - b_0 + c_0 \right) s_0 + \sum_{\mu=1}^3 (-a_\mu - b_\mu + c_\mu) s_\mu. \end{aligned} \tag{2.54}$$

Estas 4 ecuaciones pueden escribirse matricialmente de la forma

$$\mathbf{P} = \left(\frac{1}{4} \mathbf{1} + \mathbb{V}^T \mathbf{D}_0 \right) s_0 + \mathbb{V}^T \mathbb{D} \mathbf{s},$$

donde $\mathbf{1} = (1, 1, 1, 1)^T$, $\mathbf{D}_0 = (a_0, b_0, c_0)^T$, y $\mathbf{s} = (s_1, s_2, s_3)^T$. Además, las matrices \mathbb{V} y \mathbb{D} se definen de la forma

$$\mathbb{D} = \begin{pmatrix} a_1 & b_1 & c_1 \\ a_2 & b_2 & c_2 \\ a_3 & b_3 & c_3 \end{pmatrix}, \quad \mathbb{V} = \begin{pmatrix} 1 & 1 & -1 & -1 \\ 1 & -1 & 1 & -1 \\ 1 & -1 & -1 & 1 \end{pmatrix}.$$

Las componentes de la matriz de Fisher son entonces

$$F_{\mu\nu} = \sum_{i=1}^4 \frac{1}{p_i} \frac{\partial p_i}{\partial s_\mu} \frac{\partial p_i}{\partial s_\nu}.$$

Como se considera la normalización $s_0 = 1$, el resultado se escribe de la forma

$$F_{\mu\nu} = \sum_{i=1}^4 \frac{1}{p_i} \left(\mathbb{V}^T \mathbb{D} \right)_{k\mu} \left(\mathbb{V}^T \mathbb{D} \right)_{k\nu},$$

que se escribe matricialmente como

$$\mathbf{F} = \mathbf{D}^T \left(\mathbb{V} \mathbf{P}^{-1} \mathbb{V}^T \right) \mathbb{D}, \tag{2.55}$$

donde la matriz \mathbb{P} tiene la forma

$$\mathbb{P} = \begin{pmatrix} p_{00} & 0 & 0 & 0 \\ 0 & p_{01} & 0 & 0 \\ 0 & 0 & p_{10} & 0 \\ 0 & 0 & 0 & p_{11} \end{pmatrix}.$$

El parámetro de error de Fisher se define de acuerdo con la ecuación (2.44) tomando la forma

$$\Delta(\theta_A, \theta_B, \mathbf{s}) = \text{Tr} \left[\mathbb{F}^{-1}(\theta_A, \theta_B, \mathbf{s}) \right]. \quad (2.56)$$

Nuevamente, \mathbb{F}^{-1} representa la cota inferior, en este caso, de la matriz de covarianza asociada al estimador de $\hat{\mathbf{s}}$. En el apéndice C se muestra la equivalencia de este parámetro con la matriz de covarianza asociada en el límite $N \rightarrow \infty$.

Con el parámetro del error de Fisher es posible definir la qTTF asociada al estimador de acuerdo con la ecuación (2.45), en este caso, se tiene

$$qTTF(\theta_A, \theta_B) = \frac{1}{\pi} \int_0^{\frac{\pi}{2}} \int_0^{\pi} \Delta(\theta_A, \theta_B, \alpha_1, \alpha_2) d\alpha_1 d\alpha_2. \quad (2.57)$$

La optimización de la qTTF para el estimador total de espín no se puede hacer fácilmente de forma analítica, por lo cual se hace uso de la librería SciPy (Virtanen y col., 2020); se emplea el algoritmo de optimización *Nelder-mead*, un algoritmo de frecuentemente usado en problemas de optimización no lineal donde la derivación de la función objetivo no es conocida. La optimización resulta en los valores $\theta_A^{min} \approx 3,45$ y $\theta_B^{min} \approx -8,42$; con un valor de $qTTF_{opt} \approx 17$. Para realizar esta optimización es necesario iniciar los parámetros θ_A y θ_B en valores aleatorios dentro de su respectivo rango, múltiples veces. De los múltiples resultados, se pueden encontrar diferentes valores mínimos y máximos para el parámetro del error.

Para estudiar la dependencia del error del estimador respecto al estado inicial del sistema se usa nuevamente la parametrización de la ecuación (2.48). Una forma analítica explícita del parámetro del error resulta larga y con pocas posibilidades de interpretación; en cambio, en la figura 2.6 se muestra un gráfico dónde se aprecia la dependencia del error para un conjunto de parámetros Θ fijos respecto al estado inicial del sistema.

Se puede ver que el estimador total del espín no tiene el mismo error para todos los estados; al igual que el estimador de una componente. Esto apoya la decisión de elegir el valor medio del parámetro de Fisher sobre todos los posibles estados como indicador del montaje óptimo para la estimación.

El valor máximo del parámetro de Fisher se da para los valores $\alpha_1 \approx 0,40$ y $\alpha_2 \approx 2,01$; dónde el parámetro de Fisher toma el valor $\Delta \approx 22,07$. Por otra parte, el valor mínimo del parámetro de Fisher se da cuando $\alpha_1 \approx 1,17$ y $\alpha_2 = 0,48$, tomando el valor $\Delta \approx 11,93$.

Una vez definidos los estimadores óptimos se hace la simulación en una unidad cuántica de procesamiento (QPU), popularmente conocidas como computadores cuánticos. Esta implementación se realizará y discutirá en el siguiente capítulo.

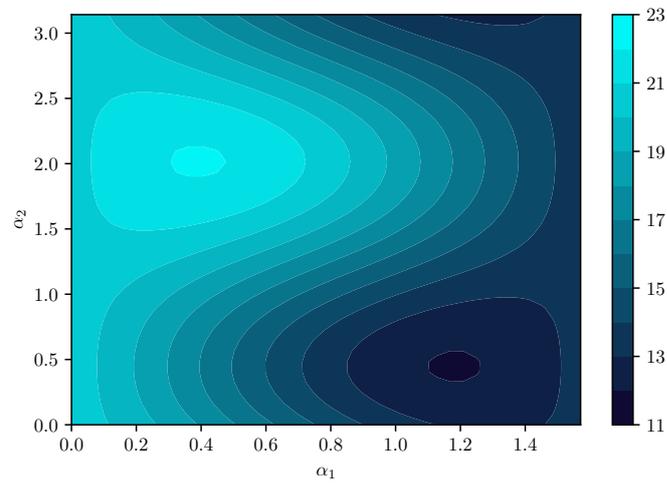


Figura 2.6: Comportamiento del parámetro de error de Fisher asociado al estimador total de espín respecto a los parámetros ζ que caracterizan el estado inicial. Para este gráfico se usan los valores óptimos del estimador $\theta_A^{min} \approx 3,45$ y $\theta_B^{min} \approx -8,42$.

Capítulo 3

Simulación en un computador cuántico

En el capítulo anterior se encontraron los montajes óptimos que permiten realizar una estimación parcial y total del espín de un partícula de espín un medio. En este capítulo se simularán estos modelos para ver su rendimiento en un computador cuántico real. En primer lugar se escoge la herramienta para realizar la simulación. Para el objetivo de este trabajo se hace uso de Qiskit (Asfaw y col., 2020), una herramienta para el desarrollo de circuitos cuánticos con base en el modelo de compuertas universales. Luego, se construyen los circuitos cuánticos asociados a los modelos de Peres y Saavedra para estimación parcial y total del espín respectivamente. Una vez construidos los circuitos, se realizan simulaciones locales, que simulan los resultados ideales, y simulaciones en una QPU de IBM. Los resultados de las estimaciones se muestran al final de cada sección para así concluir este trabajo.

3.1. Qiskit

En el año 2019 IBM lanza al público *The IBM Quantum Experience* (IBM, s.f.). Consiste en una plataforma para hacer uso de sus QPUs a través de la herramienta de software Qiskit. Esta permite construir circuitos cuánticos que pueden ser ejecutados en un simulador clásico local, usualmente el `qasm_simulator`, o, en uno de los computadores cuánticos de los que dispone IBM. Para la fecha de escritura de este documento, IBM ofrece acceso de forma gratuita a 8 de sus 43 unidades cuánticas de procesamiento. Las unidades gratuitas a las que se tiene acceso disponen entre 1 y 7 qubits. Para el desarrollo de este trabajo, se necesita una unidad de al menos 3 qubits, por lo cual, no es necesario solicitar servicio en una unidad paga.

La librería `qiskit` está disponible en el lenguaje de programación de Python. En un principio IBM exploró la posibilidad de realizar una versión para los lenguajes Swift y Java pero su desarrollo se detuvo.

Como se explicó en el capítulo 1, un circuito cuántico para computación cuántica basada en compuertas universales, consiste en un conjunto de qubits sobre los que actúan diferentes compuertas. La realización de circuitos cuánticos con `qiskit` se hace, a grandes rasgos, de la siguiente forma

1. Se inicia un conjunto de *registros cuánticos*, cuya cantidad corresponde al número de qubits que harán parte del circuito; a su vez, se inicializan un conjunto de *registros clásicos* en donde se guardará el resultado de las mediciones realizadas sobre los qubits.

2. Se definen las compuertas que actúan sobre el circuito. Los detalles para la implementación de estas compuertas se puede consultar en la referencia (Asfaw y col., 2020).
3. Finalmente, se realizan las mediciones correspondientes a los qubits que se desean medir y se almacenan en los registros clásicos previamente creados. Los resultados se consultan en forma de diccionarios de Python y pueden ser utilizados para realizar gráficos u otros procedimientos deseables.

Una guía de instalación de Qiskit fue realizada durante el desarrollo de este trabajo para el workshop Qbronze ofrecido por QColombia en marzo del año 2022 (Galvis Florez, 2022).

Un ejemplo sencillo para ilustrar el funcionamiento de Qiskit es la creación de un estado de Bell y realizar la medición respectiva. Para esto se inicia un registro cuántico de dos qubits y un registro clásico de dos bits para guardar los resultados. Con estos dos, podemos iniciar un circuito de dos qubits; por defecto, los qubits inician en el estado $|00\rangle$. Luego, aplicamos una compuerta de Hadamard sobre el primer qubit para que el siguiente estado sea $\frac{1}{\sqrt{2}}(|00\rangle + |10\rangle)$. Finalmente, se aplica una CNOT controlada por el primer qubit (sobre el que se aplicó la compuerta Hadamard) y apuntando al segundo. De esta forma, el estado final del sistema es $\frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$, que corresponde a un estado de Bell. Finalmente medimos los qubits y verificamos su resultado. El código que realiza la construcción anterior es

```
## Importar librerías necesarias
from qiskit import QuantumRegister, ClassicalRegister
from qiskit import QuantumCircuit, execute, Aer
from qiskit.visualization import plot_histogram
import matplotlib.pyplot as plt

## Iniciar los registros necesarios y el circuito
qr=QuantumRegister(2)
cr=ClassicalRegister(2)
qc=QuantumCircuit(qr, cr)

## Aplicar compuertas
qc.h(qr[0])
qc.cx(qr[0], qr[1])

## Medir
qc.measure(qr[0], cr[0])
qc.measure(qr[1], cr[1])
```

El circuito asociado al estado de Bell descrito anteriormente se ve en la figura 3.1.

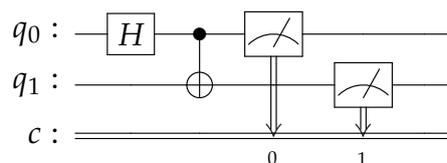


Figura 3.1: Circuito cuántico para la construcción de un estado de Bell.

Antes de la medición, el estado de los qubits 0 y 1 es

$$|\psi\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle).$$

Ahora se pasa a la ejecución del circuito, para esto se tienen en cuenta dos posibilidades. La primera es ejecutar el circuito en un simulador local, es decir, un algoritmo clásico que simula con números pseudo-aleatorios los resultados de la medición en un computador cuántico real. Para este caso se hace uso del simulador `aer_simulator`. Este ejecuta el circuito 1024 veces y entrega un conteo de las veces que se obtiene cada uno de los posibles resultados; los resultados de estas simulaciones corresponderían al caso de un computador cuántico *ideal*.

Por otra parte está la ejecución del circuito en una QPU. Para el desarrollo de este trabajo se hizo uso del simulador `ibmq_lima`. Esta QPU consta de 5 qubits lógicos que tienen una arquitectura como la que se muestra en la figura 3.2.

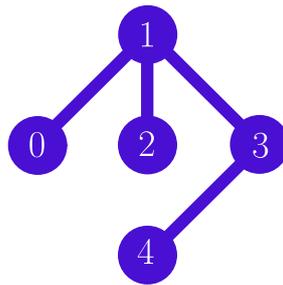


Figura 3.2: Topología de la QPU `ibmq_lima`.

La topología de una QPU usualmente cambia respecto a otra, y es importante tenerla en cuenta cuando se desea ejecutar cualquier circuito allí. Además, otro factor importante es la calibración de las compuertas. IBM muestra la fidelidad de las compuertas singulares y de múltiples qubits (CNOT) en cada una de sus QPU. Esta información también complementa la decisión de escoger el dispositivo y con cual conjunto de qubits se va a ejecutar un circuito (IBM, *s.f.*).

La ejecución del circuito cuántico mostrado anteriormente se realiza siguiendo un código similar al siguiente. Se define el simulador que se va a usar, bien sea uno clásico o uno cuántico, se ejecuta el circuito y luego se guardan los resultados para ser graficados como se muestra a continuación

```
## Definir el simulador clasico
simulator = Aer.get_backend('aer_simulator')

## Ejecutar del circuito
job=execute(qc, backend=simulator,disparos=1024)

## Guardar el resultado
result_sim=job.result()
counts_sim = result_sim.get_counts(qc)

## Cargar el proveedor de acceso a la QPU
IBMQ.load_account()
provider=IBMQ.get_provider('ibmq-q')
qcom=provider.get_backend('ibmq_lima')

## Ejecutar el circuito
job=execute(qc, backend=qcom,disparos=1024)

## Monitor que indica el tiempo restante
job_monitor(job)

result_qpu=job.result()
```

```
counts_qpu = result_qpu.get_counts(qc)
plot_histogram([counts_sim, counts_qpu], legend=['Simulación', 'IBM QPU'],
               , title='Histograma estado de Bell')
```

El gráfico que resulta de las simulaciones se muestra en la figura 3.3. Se grafican en forma de histogramas, donde las barras muestran las frecuencias $f_k = n_k/N$ asociadas al resultado k de la medición.

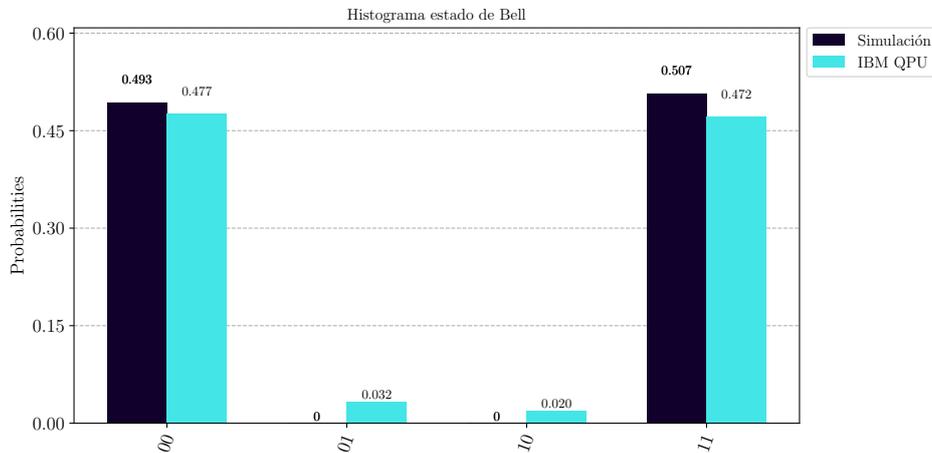


Figura 3.3: Histograma de los resultados del circuito en la figura 3.1. Las barras negras muestran el resultado de la simulación clásica (resultado ideal), mientras que las barras azules muestran los resultados de la simulación en la QPU `ibmq_1ima`.

De la figura 3.3 es importante notar un par de cosas. En primer lugar, respecto a la simulación clásica (barras negras), es importante notar que los resultados no son exactamente 50% para el estado $|00\rangle$ y $|11\rangle$, sino que varían un poco en torno a este valor, con probabilidades 49,3% y 50,7% respectivamente. Esto sucede porque el número de ejecuciones del circuito es finito $N = 1024$. Sin embargo, se cumple el hecho de que, los únicos estados posibles son $|00\rangle$ y $|11\rangle$.

Por otra parte, respecto a la simulación en la QPU, se puede ver que hay algunos resultados en los estados $|01\rangle$ y $|10\rangle$. Esto se debe a que las QPU de las que disponemos hoy en día tienen errores de hardware que alteran los resultados de la medición (Jattana y col., 2020; Unruh, 1995). Hay dos tipos de errores que se tienen en cuenta para estas mediciones. Primero, los conocidos como SPAM (State Preparation and Measurement), que corresponden a los errores por la inicialización medición de un qubit. Además, se encuentran los errores de las compuertas que se aplican sobre los qubits. De este modo, a mayor número de compuertas tenga un circuito, mayores errores se acumularán y el resultado estará más lejos de lo deseado. Un estimado de la probabilidad de errores resultantes de la medición se puede obtener si se tiene en cuenta la fidelidad de la compuerta (Johnstun y Van Huele, 2021).

Es importante tener en cuenta que cuando un circuito se ejecuta en una QPU, no se emplean directamente las compuertas que lo componen; en cambio, cada QPU tiene un conjunto de compuertas universales sobre las cuales se puede expresar cualquier circuito. En el caso de los procesadores de IBM, las compuertas base son $CNOT$, I , R_z , SX y X . El circuito se expresa en términos de estas compuertas universales y luego es ejecutado.

3.2. Estimación de una componente del espín

3.2.1. Construcción del circuito cuántico

La simulación del modelo para la estimación total del espín se hace dividiendo el circuito en cuatro partes. La primera consiste en la inicialización de los qubits. Para el caso de estimación de s_z , se inician los qubits S (el sistema que se va a estimar) y A (el metro a través del que se realiza la estimación). El qubit S inicia en un estado arbitrario mediante una rotación general que depende de los parámetros ζ y el qubit A se inicia en el estado $|+\rangle$ a través de una compuerta Hadamard. Luego, viene la aplicación del operador de evolución de la ecuación (2.2). Por la forma del operador es fácil reconocer que se trata de una compuerta de fase controlada por el qubit A y con un ángulo θ . Después de esto, es necesario cambiar la base de medición del metro A a la base de σ_x . La transformación que va de la base computacional a la de σ_x es mediada por una compuerta de Hadamard. Finalmente, se realiza la medición del metro y se guarda en un bit clásico, con el resultado de múltiples simulaciones del circuito (1024 simulaciones o disparos), se realiza la estimación de s_z . El circuito descrito anteriormente se muestra en la figura 3.4.

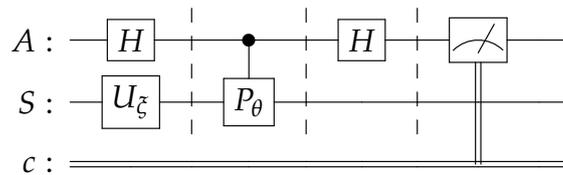


Figura 3.4: Circuito cuántico para la estimación de una componente del espín del qubit S usando el qubit auxiliar A .

3.2.2. Simulación en una QPU

Ahora, para entender el efecto de la simulación en un computador cuántico se escoge un estado inicial particular para el qubit S , en este caso $|\psi\rangle_S = \frac{1}{2}|0\rangle + \frac{\sqrt{3}}{2}|1\rangle$. Los resultados se muestran en la figura 3.5.

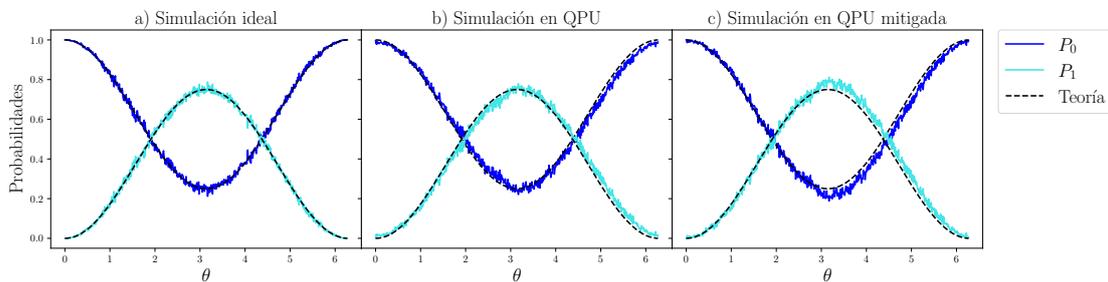


Figura 3.5: a) Simulación ideal del circuito 3.4 en el simulador local `qasm_simulator`. b) Simulación en la QPU `ibmq_lima`. Esta simulación muestra un sesgo respecto al resultado teórico, indicando la presencia de errores de hardware que afectan las mediciones. c) Simulación usando el mitigador de `qiskit`, no mejora los resultados.

La simulación ideal muestra resultados que varían alrededor del valor teórico de la simulación. Esto pasa ya que el número de ejecuciones es finito, $N = 1024$. A medida

que $N \rightarrow \infty$, la varianza baja y los resultados de la simulación se acercan más a la curva teórica.

La gráfica que corresponde a la simulación en una QPU muestra pequeñas separaciones de las cantidades P_0 y P_1 respecto a las curvas teóricas dadas por las ecuaciones (2.3) y (2.4). Esto se debe a las imperfecciones de hardware al ejecutar el circuito; tales como la fidelidad de las compuertas, decoherencia, efectos disipativos y errores en la inicialización y medición del qubit. El mitigador de errores de qiskit se implementa en el tercer gráfico. Este mitigador tiene en cuenta únicamente errores de tipo SPAM. Sin embargo, se puede ver que no mejoran la simulación del modelo de estimación, por lo tanto, no se hará uso del mitigador.

Los resultados de la figura 3.5 muestran que, la simulación del problema de estimación en una QPU es sesgada, ya que el modelo no tiene en cuenta los errores de hardware de la QPU. Un mejor modelo podría tener en cuenta correcciones a estos errores en la simulación. Sin embargo, el objetivo de este trabajo va dirigido al rendimiento del estimador de s_z como una función del parámetro θ en un hardware real. Así, el análisis continua suponiendo que el estimador es insesgado pese a los efectos de hardware.

Ahora se pasa a hacer un cálculo experimental de la qTTF. Esto se hace usando un conjunto de estados X llamados, en ingles *2-design* (Bendersky, Pastawski y Paz, 2009). Estos estados satisfacen

$$\int d\psi \langle \psi | O_1 | \psi \rangle \langle \psi | O_2 | \psi \rangle = \frac{1}{|X|} \sum_{\psi \in X} \langle \psi | O_1 | \psi \rangle \langle \psi | O_2 | \psi \rangle, \quad (3.1)$$

para todos los operadores O_1, O_2 . Acá $|X|$ representa el número de elementos de X . Se puede ver que F^{-1} se puede escribir en términos de $\langle \psi | O_1 | \psi \rangle \langle \psi | O_2 | \psi \rangle$, donde $O_1 = \frac{1}{2} (1 + \cos^2 \frac{\theta}{2}) + \frac{1}{2} \sin^2 \frac{\theta}{2} \sigma_z$ y $O_2 = \frac{1}{2} \csc^2 \frac{\theta}{2} (1 - \sigma_z)$. Para este experimento se usa $X = \{ |\psi_{z_0}\rangle, |\psi_{z_1}\rangle, |\psi_{x_0}\rangle, |\psi_{x_1}\rangle, |\psi_{y_0}\rangle, |\psi_{y_1}\rangle \} = \left\{ |0\rangle, |1\rangle, \frac{|0\rangle+|1\rangle}{\sqrt{2}}, \frac{|0\rangle-|1\rangle}{\sqrt{2}}, \frac{|0\rangle+i|1\rangle}{\sqrt{2}}, \frac{|0\rangle-i|1\rangle}{\sqrt{2}} \right\}$, es decir, el conjunto de los autoestados de los 3 operadores de Pauli. Estos seis estados son suficientes para caracterizar los resultados experimentales y así mismo el rendimiento del estimador.

En la tabla 3.1 se muestran los resultados de la simulación en la QPU `ibmq_1ima` para tres valores diferentes de θ . Los resultados corresponden cualitativamente a lo que se esperaba del experimento, incluyendo el hecho de que el estimador con $\theta = \pi$ tiene una menor varianza

Tabla 3.1: Estimación de s_z para los seis autoestados de los operadores de Pauli, en la QPU `ibmq_1ima` para los ángulos $\theta = \pi, 2\pi/3, \pi/2$. Cada estado fue simulado 5 veces con 1024 disparos en cada una, resultando así en estimaciones con 5×1024 disparos.

Estado	s_z	$s_z e (\theta = \pi)$	$s_z e (\theta = 2\pi/3)$	$s_z e (\theta = \pi/2)$
$ \psi_{z_0}\rangle$	1,0	$0,95 \pm 0,06$	$0,96 \pm 0,04$	$0,94 \pm 0,06$
$ \psi_{z_1}\rangle$	-1,0	$-0,92 \pm 0,08$	$-0,8 \pm 0,2$	$-0,8 \pm 0,2$
$ \psi_{x_0}\rangle$	0,0	$-0,04 \pm 0,05$	$0,00 \pm 0,01$	$0,03 \pm 0,09$
$ \psi_{x_1}\rangle$	0,0	$0,1 \pm 0,1$	$0,1 \pm 0,1$	$0,2 \pm 0,2$
$ \psi_{y_0}\rangle$	0,0	$0,00 \pm 0,02$	$-0,03 \pm 0,05$	$-0,01 \pm 0,02$
$ \psi_{y_1}\rangle$	0,0	$0,06 \pm 0,07$	$0,1 \pm 0,1$	$0,1 \pm 0,1$

Con los resultados de las estimaciones, se pueden construir N muestras para realizar estimaciones de s_z como se describe en el apéndice B. Esto con el objetivo de mostrar como la qTTF es una cota inferior de la varianza de la estimación (Desigualdad de

Cramér-Rao). En la figura 3.6 se compara el valor promedio de las varianzas para los seis autoestados de los operadores de Pauli y la qTTF sobre N respecto al número de muestras. Del gráfico se puede ver como la cota de Cramér-Rao se satisface dado que $\overline{\text{Var}(s_z)} \leq \overline{qTTF}/N$. Cuando $N \rightarrow \infty$ se alcanza la condición de igualdad de la cota.

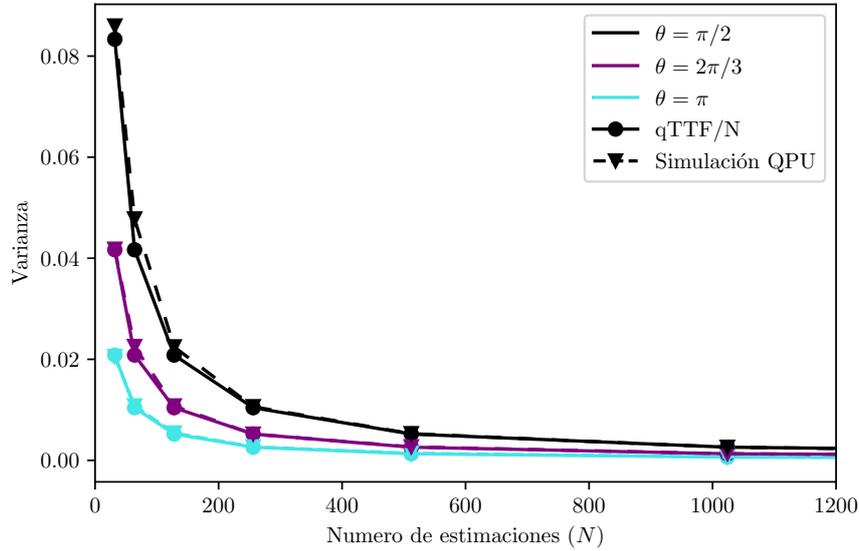


Figura 3.6: Parámetro qTTF sobre N y varianza del estimador de s_z respecto a las N muestras.

3.3. Simulación del modelo de Saavedra

3.3.1. Construcción del circuito cuántico

La simulación del modelo de Saavedra (Saavedra y Fonseca-Romero, 2019) para la estimación total del espín sigue una lógica similar a la de la estimación de una componente.

Primero se inician los 3 qubits necesarios para la simulación del modelo. Los metros A y B se inician con una compuerta de Hadamard que los lleva al estado $|+\rangle$ mientras que el qubit S se inicia con una rotación U que depende de los parámetros ξ . Luego, se aplica el operador de evolución temporal de la ecuación (2.15). Acá hay un detalle importante, y es que el operador de evolución no se puede representar fácilmente en términos de compuertas cuánticas, por esto, se hace uso de la clase `HamiltonianGate` de Qiskit. esta clase escribe el operador de evolución temporal asociado a un hamiltoniano en términos de compuerta cuánticas. Después de aplicar el operador de evolución, se aplica sobre los metros una compuerta Hadamard para llevarlos a la base de σ_x , que es la base de interés para la medición. Finalmente, se realiza la medición de los metros para realizar la estimación del estado inicial del qubit S .

El circuito correspondiente a la descripción anterior se muestra en la figura 3.7.

3.3.2. Simulación en una QPU

Para hacer una prueba preliminar de cómo funciona el circuito simulado en una QPU real, se inicia el sistema en el estado inicial tal que $c_0 = \cos \frac{\theta}{2}$ y $c_1 = e^{i\phi} \sin \frac{\theta}{2}$,

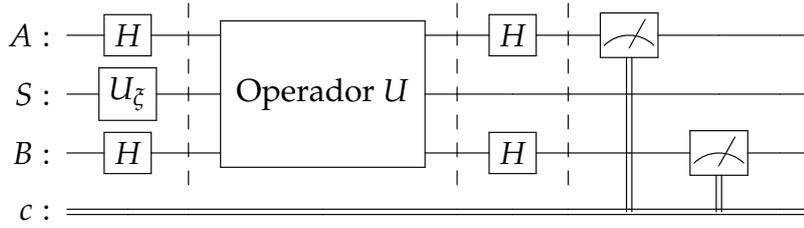


Figura 3.7: Estructura general del circuito cuántico usado para la simulación del modelo de estimación total de espín.

con $\theta = \frac{\pi}{4}$ y $\phi = \frac{\pi}{3}$. En la figura 3.8 Se muestran los resultados de la simulación en el simulador clásico `qasm_simulator`, la simulación en la QPU `ibmq_lima` y haciendo el uso del mitigador de errores SPAM de `qiskit`.

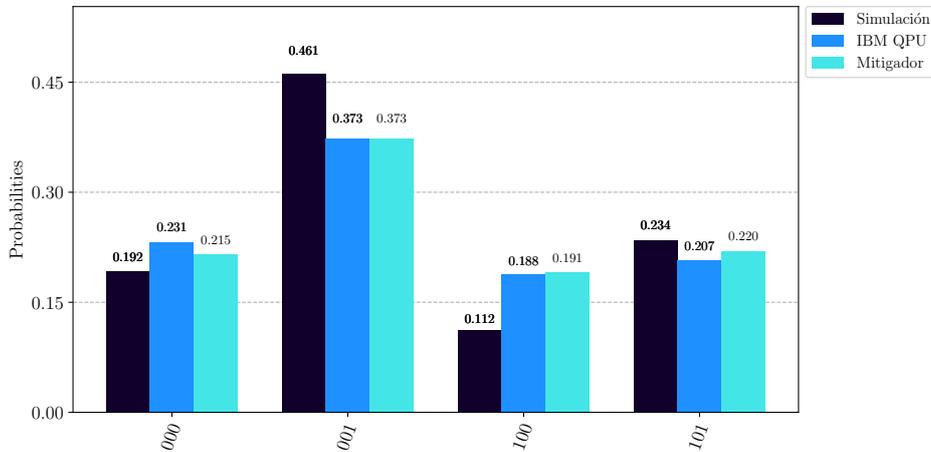


Figura 3.8: Simulación del circuito asociado para la estimación del espín de un qubit con qubits auxiliares para el estado inicial $|\psi\rangle = \cos \frac{\pi}{8} |0\rangle + e^{i\frac{\pi}{3}} \sin \frac{\pi}{8} |1\rangle$. Se muestra la simulación esperada (negro), simulación en una QPU (azul oscuro), simulación en una QPU con mitigador (azul claro).

Los resultados muestran que la simulación en la QPU no coincide con los resultados esperados de la simulación ideal. Se pueden ver errores porcentuales ($\frac{p_k - \hat{p}_k}{p_k} \times 100\%$) de hasta el 19%; como para el estado '100'.

Además, se puede ver que el mitigador, nuevamente, no realiza una mejora significativa sobre los resultados; si bien los resultados '000' y '101' muestran una mejora, los estados '100' y '001' no. Así, teniendo en cuenta el efecto del mitigador en la sección anterior, y que hay una gran diferencia entre los resultados mitigados respecto a lo que se espera de la simulación, no se hará uso de este en los siguientes resultados.

Para estudiar la eficacia del estimador, nuevamente se escogen los 6 autoestados de los operadores de Pauli $\{|\psi_{z_0}\rangle, |\psi_{z_1}\rangle, |\psi_{x_0}\rangle, |\psi_{x_1}\rangle, |\psi_{y_0}\rangle, |\psi_{y_1}\rangle\}$. Cada estado fue estimado promediando los resultados de 5 ejecuciones del circuito con 1024 disparos en cada una. La fidelidad entre el estado inicial del sistema ρ_0^S y el estado estimado ρ_e^S se calcula de la forma

$$\mathcal{F}(\rho_0^S, \rho_e^S) = [\text{Tr} \sqrt{\sqrt{\rho_0^S} \rho_e^S \sqrt{\rho_0^S}}]^2. \quad (3.2)$$

En la tabla 3.2 se muestran las estimaciones de las componentes x, y, z del vector de Bloch que describe el estado inicial del sistema y su fidelidad respecto al estado real.

Tabla 3.2: Estimación de estado total de espín para los seis autoestados de los operadores de Pauli, en la QPU `ibmq_1ima` para los ángulos óptimos $\theta_A \approx 3,45$ y $\theta_B \approx -8,42$. Cada estado fue simulado 5 veces con 1024 disparos en cada una, resultando así en estimaciones con 5×1024 disparos.

Estado	s_x	s_{xe}	s_y	s_{ye}	s_z	s_{ze}	Fidelidad
$ \psi_{z0}\rangle$	0.0	$-0,60 \pm 0,08$	0.0	$0,2 \pm 0,1$	1.0	$0,75 \pm 0,07$	85 %
$ \psi_{z1}\rangle$	0.0	$-0,7 \pm 0,2$	0.0	$0,34 \pm 0,07$	-1.0	$0,3 \pm 0,3$	37 %
$ \psi_{x0}\rangle$	1.0	$-0,3 \pm 0,3$	0.0	$0,2 \pm 0,1$	0.0	$0,7 \pm 0,2$	32 %
$ \psi_{x1}\rangle$	-1.0	$-0,75 \pm 0,07$	0.0	$0,1 \pm 0,1$	0.0	$0,5 \pm 0,2$	92 %
$ \psi_{y0}\rangle$	0.0	$-0,75 \pm 0,04$	1.0	$0,17 \pm 0,08$	0.0	$0,63 \pm 0,06$	55 %
$ \psi_{y1}\rangle$	0.0	$-0,5 \pm 0,2$	-1.0	$0,2 \pm 0,1$	0.0	$0,74 \pm 0,08$	41 %

La baja fidelidad de las estimaciones, en el peor de los casos de hasta un 32 %, es una consecuencia de la cantidad de compuertas cuánticas necesarias para hacer dicha simulación. Aún usando el optimizador de `qiskit`, el circuito transpilado en términos de las compuertas base de la QPU tiene 44 compuertas *CNOT* y aproximadamente 82 compuertas singulares, cantidad que varía respecto al estado inicial que se va a estimar. La cantidad tan grande de compuertas hace que, por cada compuerta implementada, se vayan acumulando errores en el circuito. Consecuentemente, el circuito tiene una profundidad¹ muy grande. El circuito transpilado que representa la simulación tiene una profundidad de 83.

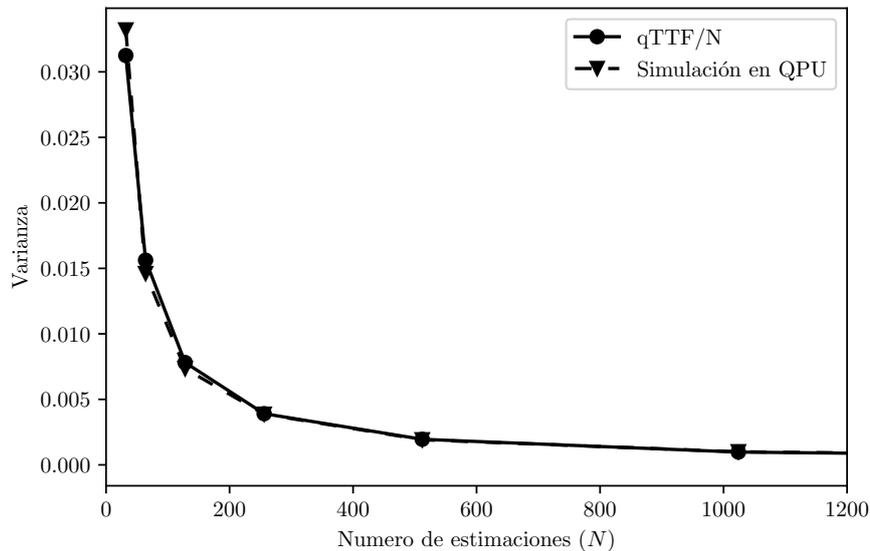


Figura 3.9: Parámetro qTTF sobre N y varianza del estimador total de espín respecto a las N muestras.

¹El número más largo de compuertas, desde la entrada hasta la salida del circuito, moviéndose a lo largo del cable de un qubit.

El hecho de que hayan errores en la simulación con un circuito que tiene una profundidad tan grande, hace evidente que el tiempo de coherencia del estado es finito. Cuando el tiempo de ejecución es muy largo, el estado que se encuentra en el procesador se disipa por interacción con el ambiente, por lo cual, se desconoce el estado final y las mediciones no corresponden a lo esperado (Jattana y col., 2020; Unruh, 1995). El mitigador de errores de qiskit no tiene efecto alguno respecto a los errores asociados a las compuertas implementadas en el circuito, por lo cual, no es una alternativa útil para mejorar la simulación.

En la figura 3.9 se toman N muestras de estimaciones totales de espín sobre los 6 autoestados de los operadores de Pauli y se compara el valor medio de la qTTF sobre N respecto al valor medio de la traza de la matriz de covarianza.

De la figura se puede ver de forma clara como la desigualdad de Cramér-Rao $\overline{\text{Cov}(s, s')} \geq \overline{qTTF}/N$ siempre es satisfecha, lo cual indica que la qTTF es un buen estimador de la varianza.

Conclusiones y perspectivas

En este trabajo se hizo una revisión de dos modelos para estimación del estado de un qubit: el primero, que depende de un único parámetro y permite hacer la estimación de una componente del estado inicial de un qubit; el segundo, que depende únicamente de dos parámetros, permite hacer una estimación total del vector de Bloch que describe el estado inicial del qubit asociado. Los dos modelos se basan en mediciones realizadas sobre uno o dos metros auxiliares que interactúan con el qubit de interés.

Además, se hizo una introducción a la simulación de sistemas cuánticos a través de circuitos cuánticos. Para esto se introdujo el modelo de computación cuántica basado en compuertas universales y se mostró el software llamado qiskit, cuyas herramientas son suficientes para realizar simulaciones cuánticas de los modelos propuestos en circuitos cuánticos que pueden ser ejecutados en simuladores clásicos y cuánticos.

Se definió el error del estimador como la traza de la inversa de la matriz de información de Fisher; la cual depende tanto del estado inicial del sistema (parámetros ζ) que se quieren estimar, como de los parámetros Θ que median la interacción entre el sistema y los metros. El valor promedio del error, sobre todos los posibles estados iniciales del sistema que son puros, se conoce como la función de transferencia tomográfica cuántica (qTTF). Esta fue usada como la medida global del error que caracteriza los estimadores y es independiente del estado inicial. La qTTF muestra grandes variaciones respecto a los parámetros que definen las interacciones en los modelos. Además, es significativamente más grande en el modelo para realizar estimación total del estado del qubit, en comparación con el modelo para realizar la estimación de una componente del vector de Bloch.

Para el caso del estimador de una componente, la optimización hecha sobre la qTTF indica que, para cualquier estado inicial, el menor valor del error en la estimación se da cuando se escoge $\Theta = \pi$. En este caso coincide con el hecho que el poder de entrelazamiento asociado al operador de evolución alcanza su valor máximo. Esto sugiere que el entrelazamiento entre el sistema y el metro es un factor importante en la calidad del estimador.

Por otra parte, para el caso de la estimación total, se encuentran los valores óptimos para el estimador $\theta_A \approx 3,45$ y $\theta_B \approx -8,42$. El error asociado al estimador muestra considerables fluctuaciones respecto al estado inicial del sistema, variando entre 11 y 23 dependiendo del estado inicial. Por otra parte, el valor medio del error es $\Delta = 17$; este es 25 veces mayor al valor medio del estimador para una componente del espín. Así, se hacen necesarias más muestras para alcanzar una estimación adecuada del estado total respecto a la estimación de una única componente del vector de Bloch.

Los dos modelos se pueden simular en una unidad cuántica de procesamiento (QPU) de IBM. En cuanto al modelo para la estimación de una componente, la dependencia de las probabilidades P_0 y P_1 respecto al parámetro de acople entre el metro y el qubit corresponde con el modelo de estimación de s_z ; por lo que la simulación en un computador cuántico real funciona adecuadamente. Se presentan pequeñas desviaciones respecto a

la curva teórica que describe dichas probabilidades. Esto son consecuencia de errores en la implementación de compuertas cuánticas, inicialización y medición de los qubits del circuito. El mitigador de IBM, no ofrece una mejora considerable en la simulación del modelo ya que solo tiene en cuenta errores de la preparación del estado y su medición (SPAM).

En cuanto al modelo para estimar completamente el estado inicial del qubit, la simulación en una QPU muestra errores por encima del 10% respecto a los resultados teóricos. Los grandes errores de esta simulación se atribuyen a la gran cantidad de compuertas necesarias para realizar la simulación del circuito y, consecuentemente, la profundidad del mismo; adicionalmente, esto hace que la coherencia del estado se pierda por la interacción con el ambiente debido al tiempo necesario para simular un circuito tan largo, por lo que la simulación de este modelo en una QPU no muestra unos resultados satisfactorios. La gran cantidad de compuertas CNOT y compuertas de un qubit necesarias para simular el circuito repercuten en las bajas fidelidades del estado estimado respecto al estado esperado. La fidelidades varían entre 30% y 90%, según los resultados obtenidos en la tabla 3.2.

La metodología que se emplea en este trabajo puede ser aplicada a otros modelos para realizar estimación, por ejemplo, modelos que implementen más qubits o diferentes interacciones entre el sistema y los metros. Sin embargo, en otros modelos el proceso de optimización de la qTTF para encontrar los mejores parámetros del estimador se hace más costoso, computacionalmente hablando, cuando el número de parámetros aumenta; demandando así la necesidad de otro tipo de optimizadores. Adicionalmente, un análisis más profundo respecto al rol de recursos cuánticos en el rendimiento del estimador, tal como el entrelazamiento, puede ser abarcado en un futuro trabajo.

Apéndice A

Expresiones explícitas de los coeficientes a_μ , b_μ , y c_μ

Después de haber definido el ángulo auxiliar $\theta_C = \sqrt{\theta_A^2 + \theta_B^2}$; los coeficientes a_μ , b_μ , y c_μ son

$$a_0 = \frac{1}{8} \cos\left(\frac{\theta_A}{2}\right) \left[\cos\left(\frac{\theta_A}{2}\right) + \frac{\theta_B \sin\left(\frac{\theta_B}{2}\right) \sin\left(\frac{\theta_C}{2}\right)}{\theta_C} + \cos\left(\frac{\theta_B}{2}\right) \cos\left(\frac{\theta_C}{2}\right) \right],$$

$$a_1 = \frac{1}{8} \sin\left(\frac{\theta_A}{2}\right) \left[\sin\left(\frac{\theta_B}{2}\right) \cos\left(\frac{\theta_C}{2}\right) - \frac{\theta_B \cos\left(\frac{\theta_B}{2}\right) \sin\left(\frac{\theta_C}{2}\right)}{\theta_C} \right],$$

$$a_2 = -\frac{\theta_A \sin\left(\frac{\theta_A}{2}\right) \sin\left(\frac{\theta_B}{2}\right) \sin\left(\frac{\theta_C}{2}\right)}{8\theta_C},$$

$$a_3 = -\frac{1}{8} \sin\left(\frac{\theta_A}{2}\right) \left[\frac{\theta_A \cos\left(\frac{\theta_B}{2}\right) \sin\left(\frac{\theta_C}{2}\right)}{\theta_C} + \sin\left(\frac{\theta_A}{2}\right) \right],$$

$$c_0 = \frac{1}{32} \left[4 \cos\left(\frac{\theta_C}{2}\right) \cos\left(\frac{\theta_A}{2} + \frac{\theta_B}{2}\right) + \cos(\theta_A - \theta_B) + \cos(\theta_A) + \cos(\theta_B) + 1 \right],$$

$$c_1 = \frac{1}{8} \left[\cos\left(\frac{\theta_A}{2}\right) \sin\left(\frac{\theta_B}{2}\right) \sin\left(\frac{\theta_A - \theta_B}{2}\right) - \frac{\theta_B \sin\left(\frac{\theta_C}{2}\right) \sin\left(\frac{\theta_A + \theta_B}{2}\right)}{\theta_C} \right],$$

$$c_2 = -\frac{1}{8} \sin\left(\frac{\theta_A}{2}\right) \sin\left(\frac{\theta_B}{2}\right) \sin\left(\frac{\theta_A - \theta_B}{2}\right).$$

Los demás coeficientes se relacionan con los anteriores a través de

$$\begin{aligned} b_0(\theta_A, \theta_B) &= a_0(\theta_B, \theta_A), & b_1(\theta_A, \theta_B) &= a_3(\theta_B, \theta_A), \\ b_2(\theta_A, \theta_B) &= -a_2(\theta_B, \theta_A), & b_3(\theta_A, \theta_B) &= a_1(\theta_B, \theta_A), \\ b_3(\theta_A, \theta_B) &= c_1(\theta_B, \theta_A). \end{aligned}$$

Apéndice B

Equivalencia de la varianza de s_Z y el parámetro de Fisher.

De la simulación, se tiene un conjunto de observaciones $S = \{s_k\}_{k=1}^N$, donde $s_k = s_z(P_0, P_1, \theta)$. Acá s_k toma los valores s_0 con probabilidad $p_0 = p$ y s_1 con probabilidad $p_1 = 1 - p$ (asumiendo $s_0 > s_1$); asociado a $s_z(1, 0, \theta)$ y $s_z(0, 1, \theta)$ respectivamente. Cada variable aleatoria s_k tiene una distribución de probabilidad tipo Bernoulli, mientras el set S tiene una distribución de probabilidad binomial.

La media de estas observaciones es

$$\bar{s} = \frac{1}{N} \sum_{k=1}^N s_k = \frac{n_0}{N} s_0 + \frac{n_1}{N} s_1,$$

donde n_0 es el número de observaciones de s_0 y n_1 es el número de observaciones de s_1 . Luego \bar{s} representa el estimador del valor esperable de la variable aleatoria s_k respecto a la distribución de probabilidad de Bernoulli. Por otra parte,

$$\sigma^2 = \frac{1}{N-1} \sum_{k=1}^N (s_k - \bar{s})^2,$$

es el estimador de la varianza de la variable aleatoria s_k respecto a la distribución de probabilidad de Bernoulli. Se puede escribir de la forma

$$\sigma^2 = \frac{N}{N-1} f_0 f_1 (s_0 - s_1)^2,$$

donde $f_0 = \frac{n_0}{N}$ y $f_1 = \frac{n_1}{N}$. Cuando $N \rightarrow \infty$, $f_0 \rightarrow P_0 = p$ y $f_1 \rightarrow P_1 = 1 - p$. Entonces, $\bar{s} \rightarrow s_0 p + s_1 (1 - p)$ y $\sigma^2 \rightarrow p(1 - p)(s_0 - s_1)^2$, que corresponden a los valores teóricos de la media y la varianza de una distribución de Bernoulli respectivamente.

Ahora se construye el estimador para el parámetro p , el cuál es el parámetro que define la distribución de probabilidad. De la media del conjunto de observaciones S , teniendo en cuenta el límite $N \rightarrow \infty$, se tiene

$$\frac{1}{N} \sum_{k=1}^N s_k = s_0 p + s_1 (1 - p).$$

Así, definimos el estimador \hat{p} del parámetro p como

$$\frac{1}{N} \sum_{k=1}^N s_k = s_0 \hat{p} + s_1 (1 - \hat{p}) \Rightarrow \hat{p} = \frac{1}{s_0 - s_1} \left[\frac{1}{N} \sum_{k=1}^N s_k - s_1 \right].$$

De este estimador podemos verificar fácilmente que el valor esperado con respecto a la distribución de probabilidad de Bernoulli es $\mathbb{E}[\hat{p}] = p$, por lo cual es un estimador

insesgado. De la ecuación (2.10) podemos escribir

$$p = \frac{1}{2} \left(1 + \cos^2 \frac{\theta}{2} \right) + \left(\frac{1}{2} \sin^2 \frac{\theta}{2} \right) s_z = \alpha + \beta s_z,$$

donde s_z es la componente del espín en dirección z . Luego, definimos el estimador para s_z . De hecho, este estimador tiene la forma

$$\hat{s}_z = \frac{1}{\beta(s_0 - s_1)} \left[\frac{1}{N} \sum_{k=1}^N s_k - s_1 \right] - \frac{\alpha}{\beta},$$

que, a su vez, es un estimador insesgado. El valor teórico de la varianza del estimador \hat{s}_z es

$$\begin{aligned} \text{Var}(\hat{s}_z) &= \text{Var} \left(\frac{1}{\beta(s_0 - s_1)} \left[\frac{1}{N} \sum_{k=1}^N s_k - s_1 \right] - \frac{\alpha}{\beta} \right) \\ &= \text{Var} \left(\frac{1}{\beta(s_0 - s_1)} \left[\frac{1}{N} \sum_{k=1}^N s_k - s_1 \right] \right) \\ \Rightarrow \text{Var}(\hat{s}_z) &= \frac{1}{N^2 \beta^2 (s_0 - s_1)^2} \text{Var} \left(\sum_{k=1}^N s_k \right), \end{aligned}$$

donde se han usado las propiedades $\text{Var}(X + \alpha) = \text{Var}(X)$ y $\text{Var}(\alpha X) = \alpha^2 \text{Var}(X)$ con α constante. Como s_k son variables aleatorias independientes, se tiene $\text{Var}(\sum_k s_k) = \sum_k \text{Var}(s_k)$. Por lo tanto, teniendo en cuenta que $\text{Var}(s_k) = \sigma^2 = p(1-p)(s_0 - s_1)^2$, se tiene

$$\text{Var}(\hat{s}_z) = \frac{1}{N \beta^2} p(1-p) = \frac{1}{N} \frac{4p_0 p_1}{\sin^4 \frac{\theta}{2}}.$$

Como se puede ver, en esta ecuación se puede identificar el valor correspondiente al parámetro de error de Fisher para la estimación de s_z dado en la ecuación (2.46).

Apéndice C

Equivalencia de la varianza del estimador \hat{s} y el parámetro de Fisher.

De forma similar a como se hizo para el estimador s_z , se tiene un conjunto de observaciones $S^i = \{s_k^i\}_{k=1}^N$ para la i -ésima componente del espín. Estas observaciones se obtienen con la medición de los metros A y B . Luego, se puede definir el valor promedio en cada dirección del espín de la forma

$$\bar{s}^i = \frac{1}{N} \sum_{k=1}^N s_k^i = \frac{n_{00}}{N} s_{00}^i + \frac{n_{01}}{N} s_{01}^i + \frac{n_{10}}{N} s_{10}^i + \frac{n_{11}}{N} s_{11}^i,$$

donde $i = x, y, z$. Acá hay que tener en cuenta que $s^i = s^i(n_{00}, n_{01}, n_{10}, n_{11})$. De este modo, $s_{00}^i = s^i(N, 0, 0, 0)$ y así con los demás resultados. Por conveniencia cambiaremos la notación $n_{00} \rightarrow n_1, n_{01} \rightarrow n_2, n_{10} \rightarrow n_3$ y $n_{11} \rightarrow n_4$.

La varianza de estas variables aleatorias son, en cada dirección,

$$\text{Var}(s_k^i) = \sigma^{i2} = \frac{1}{N-1} \sum_{k=1}^N (s_k^i - \bar{s}^i)^2.$$

Esto se puede escribir de la forma

$$\sigma^{i2} = \frac{N}{N-1} \left(\sum_{l=1}^4 f_l (s_l^i - \sum_{m=1}^4 s_m^i f_m)^2 \right).$$

Los índices n, l corresponden a los 4 posibles resultados de la medición, así mismo, $f_l = \frac{n_l}{N}$.

Cuando $N \rightarrow \infty$, se tiene que

$$\sigma^{i2} \rightarrow \sum_{l=1}^4 p_l (s_l^i - \sum_{m=1}^4 p_m s_m^i)^2.$$

En este límite, el valor medio de la estimación en la dirección i se escribe de la forma

$$\bar{s}^i \rightarrow p_1 s_1^i + p_2 s_2^i + p_3 s_3^i + p_4 s_4^i.$$

Esta relación se puede escribir de la forma matricial $\vec{\bar{s}} = \mathbb{S} \vec{p}$, donde

$$\vec{\bar{s}} = \begin{pmatrix} 1 \\ s^x \\ s^y \\ s^z \end{pmatrix}, \quad \mathbb{S} = \begin{pmatrix} 1 & 1 & 1 & 1 \\ s_1^x & s_2^x & s_3^x & s_4^x \\ s_1^y & s_2^y & s_3^y & s_4^y \\ s_1^z & s_2^z & s_3^z & s_4^z \end{pmatrix}, \quad \vec{p} = \begin{pmatrix} p_1 \\ p_2 \\ p_3 \\ p_4 \end{pmatrix}.$$

y se tiene en cuenta que $\sum_{k=1}^4 p_k = 1$. Por otra parte es importante tener en cuenta que $\vec{s} = \mathbb{T}^{-1} \vec{p}$, de la ecuación (2.30). Por lo tanto, el estimador $\vec{\bar{s}}$ es

$$\hat{s} = \mathbb{T}^{-1} \mathbb{S}^{-1} \vec{s} = \mathbb{G} \vec{s}.$$

Su i -ésima componente de la varianza $\text{Var}(\hat{s}) = \text{Var}(\mathbb{G} \vec{s})$ es

$$\begin{aligned} \text{Var}(\hat{s}^i) &= \text{Var}\left(\sum_{j=0}^4 G_{ij} \bar{s}_j\right) = \sum_{j=0}^4 G_{ij}^2 \text{Var}\left(\frac{1}{N} \sum_{k=1}^N s_k^j\right) \\ &= \sum_{j=0}^4 G_{ij}^2 \frac{1}{N^2} \sum_{k=1}^N \sigma^j{}^2. \end{aligned}$$

En términos de los resultados de la medición se tiene

$$\text{Var}(\hat{s}^i) = \frac{1}{N} \sum_{j=0}^4 G_{ij}^2 \sigma^j{}^2 = \frac{1}{N} \left(\sum_{j=0}^4 G_{ij}^2 \sum_{l=1}^4 p_l (s_l^j - \sum_{m=1}^4 p_m s_m^j)^2 \right),$$

donde $i, j = 0, x, y, z$, es decir, representan componentes del vector de espín. La componente 0 satisface la normalización $s^0 = 1$, luego, es claro que $\text{Var}(\hat{s}^0) = 0$. Por otra parte, los índices l, m corresponden a los cuatro posibles resultados de la medición del los metros, 00, 01, 10 y 11. Se puede mostrar numéricamente que la suma de los términos en paréntesis para los índices $i = x, y, z$, corresponde al parámetro del error de Fisher en la ecuación (2.56).

Bibliografía

- Arrazola, J. M. y col. (2021). «Quantum circuits with many photons on a programmable nanophotonic chip». *Nature* 591.7848, págs. 54-60. ISSN: 1476-4687. DOI: [10 . 1038 / s41586-021-03202-1](https://doi.org/10.1038/s41586-021-03202-1). URL: <https://doi.org/10.1038/s41586-021-03202-1>.
- Arute, Frank y col. (2019). «Quantum supremacy using a programmable superconducting processor». *Nature* 574.7779, págs. 505-510. ISSN: 1476-4687. DOI: [10 . 1038 / s41586-019-1666-5](https://doi.org/10.1038/s41586-019-1666-5). URL: <https://doi.org/10.1038/s41586-019-1666-5>.
- Asfaw, Abraham y col. (2020). *Learn Quantum Computation Using Qiskit*. URL: <http://community.qiskit.org/textbook>.
- Banaszek, K, M Cramer y D Gross (2013). «Focus on quantum tomography». *New Journal of Physics* 15.12, pág. 125020. DOI: [10 . 1088/1367-2630/15/12/125020](https://doi.org/10.1088/1367-2630/15/12/125020). URL: <https://doi.org/10.1088/1367-2630/15/12/125020>.
- Barenco, Adriano y col. (1995). «Elementary gates for quantum computation». *Phys. Rev. A* 52 (5), págs. 3457-3467. DOI: [10 . 1103/PhysRevA . 52 . 3457](https://link.aps.org/doi/10.1103/PhysRevA.52.3457). URL: <https://link.aps.org/doi/10.1103/PhysRevA.52.3457>.
- Bendersky, Ariel, Fernando Pastawski y Juan Pablo Paz (2009). «Selective and efficient quantum process tomography». *Phys. Rev. A* 80 (3), pág. 032116. DOI: [10 . 1103 / PhysRevA . 80 . 032116](https://link.aps.org/doi/10.1103/PhysRevA.80.032116). URL: <https://link.aps.org/doi/10.1103/PhysRevA.80.032116>.
- Benenti, Giuliano, Giulio Casati y Giuliano Strini (2004). *Principles of Quantum Computation and Information*. Vol. 2. WORLD SCIENTIFIC. DOI: [10 . 1142 / 5528](https://www.worldscientific.com/doi/pdf/10.1142/5528). eprint: <https://www.worldscientific.com/doi/abs/10.1142/5528>. URL: <https://www.worldscientific.com/doi/abs/10.1142/5528>.
- Benioff, Paul (1980). «The computer as a physical system: A microscopic quantum mechanical Hamiltonian model of computers as represented by Turing machines». *Journal of Statistical Physics* 22 (5). DOI: <https://doi.org/10.1007/BF01011339>.
- Bouwmeester, Dirk, Artur Ekert y Anton Zeilinger (2001). *the Physics of Quantum Information, Quantum Cryptography, Quantum Teleportation and Quantum Computation*. Springer-Verlag.
- Chuang, Isaac L., Neil Gershenfeld y Mark Kubinec (1998). «Experimental Implementation of Fast Quantum Searching». *Phys. Rev. Lett.* 80 (15), págs. 3408-3411. DOI: [10 . 1103 / PhysRevLett . 80 . 3408](https://link.aps.org/doi/10.1103/PhysRevLett.80.3408). URL: <https://link.aps.org/doi/10.1103/PhysRevLett.80.3408>.
- Diósi, Lajos (2007). *A Short Course in Quantum Information Theory*. Springer.
- Einstein, Albert, Boris Podolsky y Nathan Rosen (1935). «Can Quantum-Mechanical Description of Physical Reality Be Considered Complete?». *Physycal Review* 47, págs. 777-780.
- Ferrie, Christopher (2014). «Self-Guided Quantum Tomography». *Phys. Rev. Lett.* 113 (19), pág. 190404. DOI: [10 . 1103/PhysRevLett . 113 . 190404](https://link.aps.org/doi/10.1103/PhysRevLett.113.190404). URL: <https://link.aps.org/doi/10.1103/PhysRevLett.113.190404>.
- Feynman, Richard P. (1982). «Simulating physics with computers». *International Journal of Theoretical Physics* 21 (6). DOI: <https://doi.org/10.1007/BF02650179>.
- Feynman, Richard P. (1986). «Quantum mechanical computers». *Foundations of Physics* 16 (6). DOI: <https://doi.org/10.1007/BF01886518>.

- Gaikwad, Akshay y col. (2022). «Implementing efficient selective quantum process tomography of superconducting quantum gates on IBM quantum experience». *Scientific Reports* 12.1, pág. 3688. ISSN: 2045-2322. DOI: [10.1038/s41598-022-07721-3](https://doi.org/10.1038/s41598-022-07721-3). URL: <https://doi.org/10.1038/s41598-022-07721-3>.
- Galvis Florez, Cristian Andrey (2022). *QBronze installation session*. https://github.com/cagalvisf/QBronze_installation.
- Grover, Lov K. (1996). «A Fast Quantum Mechanical Algorithm for Database Search». *Proceedings of the Twenty-Eighth Annual ACM Symposium on Theory of Computing*. STOC '96. Philadelphia, Pennsylvania, USA: Association for Computing Machinery, 212–219. ISBN: 0897917855. DOI: [10.1145/237814.237866](https://doi.org/10.1145/237814.237866). URL: <https://doi.org/10.1145/237814.237866>.
- Gupta, Rishabh, Raphael D. Levine y Sabre Kais (2021). «Convergence of a Reconstructed Density Matrix to a Pure State Using the Maximal Entropy Approach». *The Journal of Physical Chemistry A* 125.34. PMID: 34410718, págs. 7588-7595. DOI: [10.1021/acs.jpca.1c05884](https://doi.org/10.1021/acs.jpca.1c05884). eprint: <https://doi.org/10.1021/acs.jpca.1c05884>. URL: <https://doi.org/10.1021/acs.jpca.1c05884>.
- Gupta, Rishabh y col. (2021). «Maximal Entropy Approach for Quantum State Tomography». *PRX Quantum* 2 (1), pág. 010318. DOI: [10.1103/PRXQuantum.2.010318](https://doi.org/10.1103/PRXQuantum.2.010318). URL: <https://link.aps.org/doi/10.1103/PRXQuantum.2.010318>.
- IBM (s.f.). *IBM Quantum Experience*. Accessed: 2022. URL: <https://quantum-computing.ibm.com/>.
- James, Daniel F. V. y col. (2001). «Measurement of qubits». *Phys. Rev. A* 64 (5), pág. 052312. DOI: [10.1103/PhysRevA.64.052312](https://doi.org/10.1103/PhysRevA.64.052312). URL: <https://link.aps.org/doi/10.1103/PhysRevA.64.052312>.
- Jattana, Manpreet Singh y col. (2020). «General error mitigation for quantum circuits». *Quantum Information Processing* 19.11, pág. 414. ISSN: 1573-1332. DOI: [10.1007/s11128-020-02913-0](https://doi.org/10.1007/s11128-020-02913-0). URL: <https://doi.org/10.1007/s11128-020-02913-0>.
- Johnson, M. W. y col. (2011). «Quantum annealing with manufactured spins». *Nature* 473.7346, págs. 194-198. ISSN: 1476-4687. DOI: [10.1038/nature10012](https://doi.org/10.1038/nature10012). URL: <https://doi.org/10.1038/nature10012>.
- Johnstun, Scott y Jean-François Van Huele (2021). «Understanding and compensating for noise on IBM quantum computers». *American Journal of Physics* 89.10, págs. 935-942. DOI: [10.1119/10.0006204](https://doi.org/10.1119/10.0006204). eprint: <https://doi.org/10.1119/10.0006204>. URL: <https://doi.org/10.1119/10.0006204>.
- Jones, Nicola (2013). «Google and NASA snap up quantum computer». *Nature*. ISSN: 1476-4687. DOI: [10.1038/nature.2013.12999](https://doi.org/10.1038/nature.2013.12999). URL: <https://doi.org/10.1038/nature.2013.12999>.
- Kandala, Abhinav y col. (2017). «Hardware-efficient variational quantum eigensolver for small molecules and quantum magnets». *Nature* 549.7671, págs. 242-246. ISSN: 1476-4687. DOI: [10.1038/nature23879](https://doi.org/10.1038/nature23879). URL: <https://doi.org/10.1038/nature23879>.
- Kay, Steven M. (1993). *Fundamentals of Statistical Signal Processing: Estimation Theory*. Prentice Hall PTR.
- Kiktenko, Evgeniy O., Daria N. Kublikova y Aleksey K. Fedorov (2020). «Estimating the precision for quantum process tomography». *Optical Engineering* 59.6, págs. 1 -9. DOI: [10.1117/1.OE.59.6.061614](https://doi.org/10.1117/1.OE.59.6.061614). URL: <https://doi.org/10.1117/1.OE.59.6.061614>.
- Killoran, Nathan y col. (mar. de 2019). «Strawberry Fields: A Software Platform for Photonic Quantum Computing». *Quantum* 3, pág. 129. ISSN: 2521-327X. DOI: [10.22331/q-2019-03-11-129](https://doi.org/10.22331/q-2019-03-11-129). URL: <https://doi.org/10.22331/q-2019-03-11-129>.
- Lehmann, E.L. y George Casella (1998). *Theory of Point Estimation*. 2.^a ed. Springer Verlag.

- Lohani, Sanjaya y col. (2021). «On the Experimental Feasibility of Quantum State Reconstruction via Machine Learning». *IEEE Transactions on Quantum Engineering* 2, págs. 1-10. DOI: [10.1109/TQE.2021.3106958](https://doi.org/10.1109/TQE.2021.3106958).
- Matteo Paris, Jaroslav Rehacek (2004). *Quantum State Estimation*. Springer, Berlin, Heidelberg. DOI: <https://doi.org/10.1007/b98673>. eprint: <https://link.springer.com/book/10.1007/b98673>. URL: <https://link.springer.com/book/10.1007/b98673>.
- Nielsen, Michael A. e Isaac L. Chuang (2011). *Quantum Computation and Quantum Information: 10th Anniversary Edition*. 10th. USA: Cambridge University Press. ISBN: 1107002176.
- Perarnau-Llobet, Martí y Theodorus Maria Nieuwenhuizen (2017). «Simultaneous measurement of two noncommuting quantum variables: Solution of a dynamical model». *Phys. Rev. A* 95 (5), pág. 052129. DOI: [10.1103/PhysRevA.95.052129](https://doi.org/10.1103/PhysRevA.95.052129). URL: <https://link.aps.org/doi/10.1103/PhysRevA.95.052129>.
- Peres, Asher (1986). «When is a quantum measurement?» *American Journal of Physics* 54.8, págs. 688-692. DOI: [10.1119/1.14505](https://doi.org/10.1119/1.14505). eprint: <https://doi.org/10.1119/1.14505>. URL: <https://doi.org/10.1119/1.14505>.
- Rezakhani, A. T. (2004). «Characterization of two-qubit perfect entanglers». *Phys. Rev. A* 70 (5), pág. 052313. DOI: [10.1103/PhysRevA.70.052313](https://doi.org/10.1103/PhysRevA.70.052313). URL: <https://link.aps.org/doi/10.1103/PhysRevA.70.052313>.
- Saavedra, Daniel y K. M. Fonseca-Romero (2019). «Complete and incomplete state estimation via the simultaneous unsharp measurement of two incompatible qubit operators». *Phys. Rev. A* 99 (4), pág. 042130. DOI: [10.1103/PhysRevA.99.042130](https://doi.org/10.1103/PhysRevA.99.042130). URL: <https://link.aps.org/doi/10.1103/PhysRevA.99.042130>.
- Sakurai, J. John y Jim J. Napolitano (2014). *Modern Quantum Mechanics*. Pearson Education Limited.
- Shor, P.W. (1994). «Algorithms for quantum computation: discrete logarithms and factoring». *Proceedings 35th Annual Symposium on Foundations of Computer Science*, págs. 124-134. DOI: [10.1109/SFCS.1994.365700](https://doi.org/10.1109/SFCS.1994.365700).
- Teo, Yong Siah (2015). *Introduction to Quantum-State Estimation*. WORLD SCIENTIFIC. DOI: [10.1142/9617](https://doi.org/10.1142/9617). eprint: <https://www.worldscientific.com/doi/pdf/10.1142/9617>. URL: <https://www.worldscientific.com/doi/abs/10.1142/9617>.
- Tilma, Todd y E C G Sudarshan (2002). «Generalized Euler angle parametrization for $SU(N)$ ». *Journal of Physics A: Mathematical and General* 35.48, págs. 10467-10501. DOI: [10.1088/0305-4470/35/48/316](https://doi.org/10.1088/0305-4470/35/48/316).
- Turing, A. M. (1938). «On Computable Numbers, with an Application to the Entscheidungsproblem. A Correction». *Proceedings of the London Mathematical Society* s2-43.1, págs. 544-546. DOI: <https://doi.org/10.1112/plms/s2-43.6.544>.
- Unruh, W. G. (1995). «Maintaining coherence in quantum computers». *Phys. Rev. A* 51 (2), págs. 992-997. DOI: [10.1103/PhysRevA.51.992](https://doi.org/10.1103/PhysRevA.51.992). URL: <https://link.aps.org/doi/10.1103/PhysRevA.51.992>.
- Virtanen, Pauli y col. (2020). «SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python». *Nature Methods* 17, págs. 261-272. DOI: [10.1038/s41592-019-0686-2](https://doi.org/10.1038/s41592-019-0686-2).
- Watrous, John (2018). *The Theory of Quantum Information*. Cambridge University Press.
- Wilde, Mark M (2013). *Quantum Information Theory*. Cambridge University Press.
- Wootters, William K. (1998). «Entanglement of Formation of an Arbitrary State of Two Qubits». *Phys. Rev. Lett.* 80 (10), págs. 2245-2248. DOI: [10.1103/PhysRevLett.80.2245](https://doi.org/10.1103/PhysRevLett.80.2245). URL: <https://link.aps.org/doi/10.1103/PhysRevLett.80.2245>.

- Zanardi, Paolo, Christof Zalka y Lara Faoro (2000). «Entangling power of quantum evolutions». *Phys. Rev. A* 62 (3), pág. 030301. DOI: [10.1103/PhysRevA.62.030301](https://doi.org/10.1103/PhysRevA.62.030301). URL: <https://link.aps.org/doi/10.1103/PhysRevA.62.030301>.
- Řeháček, Jaroslav, Yong Siah Teo y Zdeněk Hradil (2015). «Determining which quantum measurement performs better for state estimation». *Phys. Rev. A* 92 (1), pág. 012108. DOI: [10.1103/PhysRevA.92.012108](https://doi.org/10.1103/PhysRevA.92.012108). URL: <https://link.aps.org/doi/10.1103/PhysRevA.92.012108>.
- Řeháček, Jaroslav y col. (2007). «Diluted maximum-likelihood algorithm for quantum tomography». *Phys. Rev. A* 75 (4), pág. 042108. DOI: [10.1103/PhysRevA.75.042108](https://doi.org/10.1103/PhysRevA.75.042108). URL: <https://link.aps.org/doi/10.1103/PhysRevA.75.042108>.