

Trabajo de Fin de Grado

Construcción de puertas lógicas para el desarrollo de la
computación cuántica mediante la teoría de control óptimo

Autor

ADRIÁN GARCÍA CARRIZO

Dirigido por

ALBERTO CASTRO BARRIGÓN

Resumen

Hemos estudiado, computacionalmente, el diseño de puertas lógicas en circuitos de información cuántica, en concreto, puertas para sistemas de uno y dos qbits. En este documento comenzamos con una breve introducción a la computación cuántica, que nos permitirá entender el funcionamiento de las puertas lógicas cuánticas. Esencialmente, un sistema de N qbits se modela mediante un sistema cuántico de 2^N niveles; una puerta cuántica es una operación unitaria sobre este sistema. Esta operación se corresponde con la acción de una perturbación diseñada adecuadamente para que el operador evolución resultante coincida con la operación buscada. Tras definir la perturbación mediante una forma funcional dependiente de unos parámetros, hemos utilizado la teoría de control óptimo para buscar los parámetros que optimizan la forma del operador evolución. Finalmente, hemos estudiado cómo las formas funcionales elegidas, o el uso de unos u otros algoritmos de optimización afecta a la calidad de la puerta lógica obtenida, o al coste computacional.

Índice

1. Introducción a la computación cuántica	1
2. Objetivos	3
3. Construcción de puertas lógicas mediante QOCT	3
4. QOCT vs. oscilaciones de Rabi	7
4.1. Oscilaciones de Rabi en un sistema de dos estados	7
4.2. π -pulse vs QOCT	11
5. Elección de parámetros y métodos de optimización	14
5.1. Algoritmos de optimización	14
5.2. Amplitudes y tiempo de propagación	18
6. Resultados	18
6.1. Puerta Controlled-NOT	19
6.2. Puerta Controlled-Z	20
6.3. Puerta Controlled-Y	20
6.4. Puerta SWAP	21
6.5. Puerta $\sqrt{\text{SWAP}}$	22
6.6. Controlled-phase	23
7. Conclusiones	24
Bibliografía	25
Anexo 1: Centro de supercomputación de Aragón	26
Anexo 2: Justificación experimental	27

Índice de figuras

4.1. Solución numérica a las oscilaciones de Rabi para distintos valores de w	11
4.2. Transición del estado $ 0\rangle$ al $\frac{1}{\sqrt{2}}(0\rangle + 3\rangle)$ mediante pulsos- π	13
4.3. Transición del estado $ 0\rangle$ al $\frac{1}{\sqrt{2}}(0\rangle + 3\rangle)$ mediante QOCT.	15
5.1. Variación de J para los cuatro algoritmos estudiados.	17
6.1. Perturbación aplicada para generar la puerta CNOT	19
6.2. Perturbación aplicada para generar la puerta CONTROL-Z	20
6.3. Perturbación aplicada para generar la puerta CONTROL-Y	21
6.4. Perturbación aplicada para generar la puerta SWAP	22
6.5. Perturbación aplicada para generar la puerta $\sqrt{\text{SWAP}}$	23
6.6. Perturbación aplicada para generar la puerta CONTROL-Phase	24

Índice de tablas

4.1. coeficientes óptimos de la perturbación aplicada en la transición de $ \varphi(0)\rangle$ a $ \varphi(t)\rangle$.	14
5.1. Resultados del proceso de optimización deteniendo el programa para $J > 0.995$. . .	16
5.2. Resultados del proceso de optimización deteniendo el programa para $J > 0.9995$. .	16
5.3. Resultados del proceso de optimización deteniendo el programa para $J > 0.99999$. .	16
5.4. Valor obtenido de J para para distintas combinaciones de A y T . ω_i hace referencia a las tres frecuencias naturales del sistema pues $i = \{1, 2, 3\}$	19
6.1. Parámetros del pulso para generar una puerta CNOT	19
6.2. Parámetros del pulso para generar una puerta CONTROL-Z	20
6.3. Parámetros del pulso para generar una puerta CONTROL-Y	21
6.4. Parámetros del pulso para generar una puerta SWAP	22
6.5. Parámetros del pulso para generar una puerta $\sqrt{\text{SWAP}}$	23
6.6. Parámetros del pulso para generar una puerta CONTROL-Phase	24

1. Introducción a la computación cuántica

Comenzamos con una breve introducción a los conceptos de la teoría de información cuántica que nos han sido necesarios para la elaboración de este trabajo – una introducción completa al tema puede consultarse en la Ref. [1]. La computación clásica se basa en la manipulación de la unidad mínima de información, el bit, que solo puede tomar dos valores, por ejemplo, 0 o 1. La computación cuántica utiliza el principio de la superposición de estados para ir más allá, apareciendo el concepto de q-bit (*quantum bit*). Un q-bit puede entenderse como un sistema cuántico de dos estados, que denotaremos como $|0\rangle$ y $|1\rangle$. De esta forma, el estado más general posible para un q-bit será:

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle \quad \forall \alpha, \beta \in \mathbb{C} \quad t.q. \quad |\alpha|^2 + |\beta|^2 = 1 \quad (1.1)$$

Esta mayor libertad permite que, algoritmos cuyo tiempo de cálculo crece potencialmente con el número de bits de entrada, pasen a tener tiempos de cálculo que crecen linealmente. Un ejemplo es el algoritmo de Deutsch-Jozsa^[2].

La manipulación de la información en computación clásica se realiza mediante las llamadas *puertas lógicas*, que toman un número de bits de entrada, y producen unos bits de salida de acuerdo con reglas predefinidas. Análogamente, existen puertas lógicas cuánticas que toman el estado de uno o varios q-bits y los alteran de cierta forma. Las puertas más sencillas tienen un solo q-bit de entrada. Un primer ejemplo es la puerta Pauli-X (o *bit-flip*): Esta toma el estado del q-bit e intercambia el $|0\rangle$ por el $|1\rangle$ y viceversa. Podemos compilar el comportamiento de cualquier puerta lógica de un 1 q-bit como una matriz 2x2 unitaria definida sobre la base ortonormal $\{|0\rangle, |1\rangle\}$ tal que:

$$\begin{array}{l} |0\rangle \longrightarrow |1\rangle \\ |1\rangle \longrightarrow |0\rangle \end{array} \quad \text{---} \boxed{X} \text{---} \quad U_{\text{Pauli-X}} = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$$

Hemos incluido también el símbolo que suele utilizarse en la representación de circuitos lógicos. Viendo la representación matricial de esta puerta cobra sentido su nombre pues, más formalmente, aplica la matriz de Pauli σ_x al estado inicial. De la misma forma existe la puerta Pauli-Y y Pauli-Z.

A diferencia de las puertas lógicas clásicas, las cuánticas no solamente actúan sobre los estados base $|0\rangle$ o $|1\rangle$, sino que puede actuar sobre cualquier estado superposición. Así, por ejemplo, si introducimos como estado de entrada $|\varphi_1\rangle = |0\rangle - i|1\rangle$ tendremos que se obtendrá como estado final $U_{\text{Pauli-X}}|\varphi_1\rangle = |1\rangle - i|0\rangle$. Otra puerta muy utilizada es la puerta Hadamard, que devuelve superposición de estados:

$$\begin{array}{l} |0\rangle \longrightarrow \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \\ |1\rangle \longrightarrow \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle) \end{array} \quad \text{---} \boxed{H} \text{---} \quad U_{\text{Hadamard}} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$$

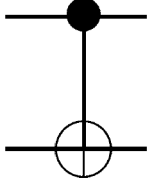
Si queremos trabajar con más q-bits y sus representaciones matriciales necesitamos definir clara-

mente una generalización de la base para un sistema de N niveles. En el caso de sistemas formados por un único q-bit la base es trivial, sin embargo, para un sistema de N q-bits primero definimos una base para cada uno de los q-bits y luego tomamos la base del sistema como el producto tensorial de todas estas bases^[1]. La base resultante se denomina base computacional e incluye todas las posibles combinaciones de estados de cada q-bit. Para simplificar la notación denominaremos al ket que representa cada estado de la base con el número que representa en base decimal. Así, denotaremos cada uno de estos estados como:

$$|a_N\rangle_N \otimes |a_{N-1}\rangle_{N-1} \otimes \cdots \otimes |a_1\rangle_1 = |a_N a_{N-1} \cdots a_1\rangle = |c\rangle, \quad (1.2)$$

donde $c = a_N 2^{N-1} + a_{N-1} 2^{N-2} + \cdots + a_1$. Así, la base (ortonormal) se expresará como $\{|0\rangle, |1\rangle, \dots, |2^N - 1\rangle\}$ y cualquier estado se podrá formar mediante una combinación lineal de los kets que la forman.

Como ejemplo de puertas de 2 qbits, una de las primeras con las que hemos trabajado es la CNOT (“Controlled” NOT): si el primer q-bit está en estado $|0\rangle$ no realiza ningún cambio en ningún q-bit, pero si está en $|1\rangle$, aplica σ_x al segundo q-bit. Así:

$$\begin{aligned} |0\rangle &= |00\rangle \longrightarrow |00\rangle = |0\rangle \\ |1\rangle &= |01\rangle \longrightarrow |01\rangle = |1\rangle \\ |2\rangle &= |10\rangle \longrightarrow |11\rangle = |3\rangle \\ |3\rangle &= |11\rangle \longrightarrow |10\rangle = |2\rangle \end{aligned}$$


$$U_{\text{CNOT}} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

El resto de puertas de dos qbits se construye utilizando reglas semejantes. Existen asimismo puertas para sistemas de tres o más qbits, teniendo en cuenta que el tamaño de la matriz crece exponencialmente con el número de q-bits: la dimensión es 2^N .

Formalmente, una puerta cuántica no es más que un operador unitario U . En las realizaciones físicas de los sistemas de computación cuántica, una puerta lógica cuántica no es más que un operador evolución temporal, $U(T, 0)$, que actúa durante un tiempo característico T produciendo la transformación deseada. Los sistemas cuánticos, sin embargo, están sujetos a perturbaciones externas incontrolables, ya que en la práctica no podemos aislarlos totalmente de su entorno. Ello implica que el comportamiento típicamente cuántico, *coherente*, se deteriora con el tiempo. Es necesario que los tiempos de decoherencia típicos del sistema sean lo más largos posibles, o alternativamente, que los tiempos de operación T sean cortos. Este trabajo está motivado precisamente en el intento de reducir estos tiempos, buscando maneras *óptimas* de diseñar las puertas cuánticas.

Este trabajo está motivado, igualmente, por el proyecto SUMO (Scaling Up quantum computation with MOlecular spins), brevemente descrito en el Anexo 2. En este proyecto se realiza un montaje experimental que trata de crear puertas cuánticas con moléculas magnéticas. Se enfrenta precisamente al problema de los cortos tiempos de decoherencia, y es por ello que una solución puede ser la aplicación de la QOCT para acelerar las puertas cuánticas.

2. Objetivos

Ahora que sabemos qué es una puerta lógica cuántica nos preguntamos cuál es la forma de modelar un sistema físico que implemente esta idea. La forma de inducir este comportamiento se basa en introducir una perturbación dependiente del tiempo en un Hamiltoniano de un sistema de 2^N niveles. Esta perturbación debe estar diseñada para que el correspondiente operador evolución (a un tiempo T lo más breve posible) sea igual (o equivalente) a la puerta lógica deseada. Matemáticamente, el problema consiste por lo tanto en: dado un Hamiltoniano dependiente de unos parámetros (que, en nuestro caso, definen la forma funcional de la perturbación), deben encontrarse aquellos parámetros que inducen en el sistema un comportamiento predefinido (en nuestro caso, la evolución de acuerdo con la puerta lógica buscada). Este planteamiento matemático encaja con el problema estudiado por la teoría de control óptimo cuántico (QOCT), que introduciremos más adelante. Por lo tanto, nuestros objetivos en este trabajo han sido:

1. Aprender los fundamentos teóricos de la teoría de información cuántica, y de la QOCT.
2. Aprender a ejecutar un código que implementa la QOCT para sistemas genéricos.
3. Ejecutar cálculos de optimización para diseñar puertas lógicas para sistemas de 2 q-bits modelo.
4. Estudiar la eficiencia de estos cálculos con respecto a los parámetros libres que han de decidirse a la hora de buscar la forma de la perturbación óptima (amplitud máxima, tiempo de aplicación de la perturbación, etc).
5. Estudiar la eficiencia de algunos de los algoritmos de optimización posibles.

3. Construcción de puertas lógicas mediante QOCT

Partimos del Hamiltoniano H_0 de un sistema de N niveles que, sin pérdida de generalidad, consideraremos que es diagonal (de no serlo, se realiza un cambio de base a la base de autoestados). Introducimos entonces una perturbación dependiente del tiempo, y nos preguntamos cuál debe ser la forma de ésta tal que el operador evolución temporal coincida con el operador asociado a una puerta lógica para un tiempo T fijo. Así, sea la ecuación de Schrödinger:

$$i \frac{d}{dt} |c(t)\rangle = (H_0 + f(\{u_i\}, t)V) |c(t)\rangle, \quad (3.1)$$

donde $\{u_i\}$ son una serie de parámetros que controlan la forma de la perturbación, que denominaremos parámetros de control, y V un operador que conecta distintos niveles del sistema. Para simplificar la notación escribiremos $u = \{u_i\}$. Podemos tomar, por ejemplo:

$$f(u, t) = \sum_{i=1}^m u_i \sin(w_i t + u_{i+m}). \quad (3.2)$$

Siendo así m el número de frecuencias de las que se compone la perturbación. Normalmente, estas se corresponden con las frecuencias naturales del sistema o sus armónicos. El operador evolución temporal se define entonces como:

$$|c(t)\rangle = U(t, t') |c(t')\rangle . \quad (3.3)$$

Por simplicidad tomaremos el tiempo inicial de propagación en $t = 0$ tal que $U(t, 0) = U(t)$ y $U(0) = \mathbb{I}$. Así, como $|c(t)\rangle = U(t) |c(0)\rangle$ podemos escribir la ecuación (3.1) para $U(t)$ y su condición inicial como:

$$\begin{aligned} i \frac{d}{dt} U(t) &= (H_0 + f(u, t)V)U(t) \\ U(0) &= \mathbb{I} . \end{aligned} \quad (3.4)$$

Ahora bien, normalmente, en vez de usar la representación de Schrödinger utilizaremos la representación de interacción, que simplifica bastante la forma de esta ecuación. En esta representación atribuimos dependencia temporal tanto a los operadores como a los vectores de estado, de forma que:

$$\tilde{O}(t) = e^{itH_0} O e^{-itH_0} , \quad (3.5)$$

$$|\tilde{c}(t)\rangle = e^{itH_0} |c(t)\rangle . \quad (3.6)$$

$$|\tilde{c}(t)\rangle = \tilde{U}(t) |\tilde{c}(0)\rangle . \quad (3.7)$$

Así, la expresión (3.4) puede escribirse en la representación de interacción de la forma:

$$i \frac{d}{dt} \tilde{U}(t) = f(u, t) \tilde{V}(t) \tilde{U}(t) . \quad (3.8)$$

Como los parámetros de control son los que dictan la evolución del sistema podemos escribir $\tilde{U}(t)$ como una función de esos parámetros: $\tilde{U}(t) = \tilde{U}[u](t)$. De esta forma, nuestro objetivo será buscar aquellos parámetros que hagan que este operador evolución sea lo más parecido posible a la puerta lógica objetivo. La perturbación se aplica un tiempo T , que interesará que sea pequeño.

La búsqueda de el valor óptimo de estos parámetros se llevará a cabo mediante la teoría de control óptimo^[3]. Esta es la aplicación al mundo cuántico de la más general teoría de control óptimo, aplicable en principio a cualquier tipo de sistema y proceso^[4]. Su aplicación al problema que nos ocupa es como sigue: Comenzamos por definir una función objetivo (de ahora en adelante, por simplicidad, dejamos de utilizar la tilde \sim para denotar los objetos en la representación de interacción):

$$G(u) = J_1(U[u](T)) + J_2(u) . \quad (3.9)$$

La definición de las funciones J_1 y J_2 se hacen teniendo en cuenta el objetivo de la optimización: buscamos que la función G tome valores máximos cuando los parámetros implican la consecución del objetivo. Así, J_1 es una función que depende de la forma del operador evolución temporal en el tiempo característico T , y que deberá tomar valores máximos cuando este operador sea igual (o equivalente) a la puerta cuántica buscada. En ocasiones se utiliza una función opcional J_2 que

penaliza determinadas regiones del espacio de parámetros que, por la razón que sea (dificultad para su utilización experimental, etc.), es poco apropiada.

Dado que $G(u)$ tiene que tener un máximo para los parámetros que hagan que $U[u](T)$ equivalga a un operador asociado a una puerta lógica, tomamos J_1 como el módulo al cuadrado del producto de Fröbenius entre $U[u](T)$ y el operador objetivo. El producto de Fröbenius entre dos operadores A y B se define como:

$$A \cdot B = \frac{1}{d} \text{Tr}[A^\dagger B], \quad (3.10)$$

siendo d la dimensión de las matrices. Por lo tanto:

$$J_1 = |U[u](T) \cdot U_{\text{target}}|^2. \quad (3.11)$$

Para operadores unitarios, este producto no puede tomar valores mayores que uno, y valdrá uno cuando los operadores sean equivalentes, es decir, que se diferencien únicamente en un factor de fase global, e. g.: $U(T) = e^{i\theta} U_{\text{target}}$:

$$J_1 = |U(T) \cdot U_{\text{target}}|^2 = |e^{-i\theta} U_{\text{target}}^\dagger \cdot U_{\text{target}}|^2 = \frac{|e^{-i\theta}|^2}{d^2} \text{Tr}[I]^2 = 1 \quad (3.12)$$

El problema se reduce por lo tanto a la maximización de la función $G(u)$. Necesitamos un algoritmo de optimización, que en general necesitará de un procedimiento para calcular $G(u)$ y, en algunos casos, de un procedimiento para calcular el gradiente de $G(u)$. Para calcular $G(u)$ el programa, dado un valor fijo T , resuelve la ecuación (3.8) para los valores iniciales de u , obteniendo $U(T)$. Hecho esto, calcula J . Si el algoritmo necesita del cálculo de la derivada de $G(u)$ con respecto a distintos parámetros de control, la QOCT nos proporciona la siguiente expresión^{[3],[5]}:

$$\frac{\partial G(u)}{\partial u_m} = 2\text{Im} \left[\int_0^T dt \frac{\partial f(u, t)}{\partial u_m} B^\dagger[u](t) \cdot V(t) U[u](t) \right] + \frac{\partial J_2(u)}{\partial u_m}, \quad (3.13)$$

donde aparece un nuevo objeto, el llamado *coestado* $B[u](t)$, definido por las siguientes ecuaciones de movimiento:

$$\begin{aligned} i \frac{d}{dt} B[u](t) &= f(u, t) V^\dagger(t) B[u](t) \\ B[u](T) &= (U_{\text{target}}(T) \cdot U[u](T)) U_{\text{target}}(T). \end{aligned} \quad (3.14)$$

Estas ecuaciones tienen una gran similitud con las ecuaciones del movimiento para el sistema [expresión (3.4)]: se trata de una ecuación de Schrödinger en la representación de interacción con una condición inicial. Ahora bien, esta condición inicial, en lugar de estar referenciada a tiempo 0 lo está a tiempo T , es decir, el tiempo final, lo cual implica que tiene que integrarse numéricamente hacia atrás. Asimismo, es claro que el cálculo del gradiente implica un coste mayor que el cálculo del valor de la función.

El cómo se explora el espacio de parámetros depende del algoritmo de optimización. Se ha usado la librería *NLopt*^[6]. Los algoritmos pueden dividirse en general de dos maneras: primero, pueden

ser *globales* o *locales*. Segundo, pueden requerir el gradiente, o no hacerlo (*derivative free*).

Los algoritmos de optimización globales permiten encontrar el mínimo o máximo absoluto de una función dentro de un determinado dominio acotado de parámetros. La dificultad de este tipo de problemas crece exponencialmente con el número n de parámetros, y por lo general no son sencillos de resolver. Por otra parte, los algoritmos locales tienen como objetivo encontrar un único mínimo local. Pueden existir una gran cantidad de mínimos locales: cuál de todos encuentre dependerá tanto del funcionamiento del propio algoritmo como de los parámetros iniciales que se usen para comenzar la búsqueda. Este tipo de algoritmos no suele tener dificultades en ubicar un mínimo local, incluso para problemas de dimensión alta. Los algoritmos de optimización locales que garantizan el encontrar un mínimo local desde cualquier punto de partida del espacio de parámetros se suelen denominar globalmente convergentes, lo cual suele dar lugar a confusión.

La otra clasificación mencionada se basa en si el algoritmo requiere o no el uso de la derivada de la función a optimizar: los algoritmos locales, por ejemplo, suelen requerir el cálculo del gradiente (además del propio valor de la función), pues es un factor importante para acelerar la eficiencia de la optimización. Esto suele suponer problemas pues la función a optimizar no tiene por qué tener definido el gradiente en todo su dominio (no es nuestro caso) o su cálculo no tiene por qué ser obvio. En este último caso se recurre a técnicas de diferenciación automática, lo cual supone un aumento no despreciable en el coste computacional. Sin embargo, si la función no es derivable (o peor, discontinua) se usan los algoritmos *derivative-free* que solo requieren los valores de la función. Estos algoritmos suelen evaluar la función una gran cantidad de veces y, por ello, suelen ser usados para la optimización (global o local) de funciones con un número n de parámetros no muy grande.

En nuestro caso los algoritmos en los que nos hemos centrado han sido:

- MLSL-LDS(Multi-Level Single Linkage):^[7] MLSL es un algoritmo de comienzo múltiple. Funciona lanzando una secuencia de optimizaciones locales en una serie de puntos iniciales escogidos mediante secuencias de baja discrepancia para evitar caer varias veces en el mismo mínimo local. Existen garantías teóricas de encontrar todos los mínimos locales en un número finito de minimizaciones locales. Como algoritmo local puede usarse cualquiera de los incluidos en la librería, con o sin el uso de derivadas, siendo el usado por defecto el algoritmo BFGS (Broyden–Fletcher–Goldfarb–Shanno)^[8], que si que requiere el cálculo del gradiente.
- CRS-LM (Controlled Random Search with Local Mutation):^[9] El algoritmo CRS es un algoritmo de optimización global que no requiere el uso de la derivada de la función a optimizar. Se basa en métodos puramente heurísticos y comienza generando una serie de puntos distribuidos de forma uniforme sobre el espacio de búsqueda. Este conjunto se va “contrayendo” mediante la sustitución de los puntos que dan peor resultado con otros mejores mediante simplex, similar al algoritmo de Nelder y Mead^[10].
- ISRES (Improved Stochastic Ranking Evolution Strategy):^[11] Al igual que el anterior, es un algoritmo de optimización global (o al menos semi-global, pues se basa en métodos heurísticos para escapar de mínimos locales) que no requiere el uso de la derivada. Se basa en una combinación de una regla de mutación y una regla de actualización similar a la del algoritmo de Nelder-Mead.^[10]

- ESCH (Evolutionary Algorithm): Se trata de una modificación del algoritmo evolutivo para optimización global desarrollado por C.H de Silva Santos, véase [12].

4. QOCT vs. oscilaciones de Rabi

Antes de comenzar los cálculos de optimización previstos, hemos comprobado la funcionalidad del código utilizando un sistema de solución analítica exacta conocida: el sistema de dos niveles. En este sistema se demuestran las oscilaciones de Rabi y los llamados pulsos- π , que son el mecanismo habitual utilizado en sistemas de computación cuántica para inducir las transiciones de niveles. En sistemas con más de dos niveles, las transiciones mediante pulsos π , sin embargo, necesitan de un tiempo largo, ya que solo serán exactas cuando puedan desprejarse las transiciones a niveles no deseados, gracias a sus diferentes frecuencias de resonancia. Por ello, demostraremos también en esta sección cómo la QOCT permite acortar los tiempos de transición.

El programa que se ha utilizado se denomina *quote* y se puede acceder a él a través del siguiente link: <https://gitlab.com/acbarrigon/quote>. Los ficheros de entrada necesarios para reproducir los cálculos de este trabajo pueden obtenerse del autor.

4.1. Oscilaciones de Rabi en un sistema de dos estados

La aplicación de QOCT requiere de sucesivas integraciones de la ecuación de Schrödinger para el sistema en cuestión y de sucesivas integraciones de la ecuación para el co-estado [Eq. (3.14)]. Por ello, antes de comenzar los cálculos de control óptimo, se va a comprobar si los cálculos de evolución de estados son correctos. Para ello se usará un caso conocido y con solución analítica de un sistema muy estudiado y de gran importancia en el campo de la computación cuántica: Las oscilaciones de Rabi en el sistema cuántico de dos niveles.

Consideremos el Hamiltoniano, H_0 de un sistema de dos niveles ($|1\rangle$ y $|2\rangle$) con energías $-\frac{\omega_0}{2}$ y $\frac{\omega_0}{2}$ respectivamente. En base $\{|1\rangle, |2\rangle\}$ la representación matricial de H_0 viene dada como

$$H_0 = \begin{pmatrix} -\omega_0/2 & 0 \\ 0 & \omega_0/2 \end{pmatrix} = -\frac{\omega_0}{2}\sigma_z \quad \text{con} \quad \sigma_z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}. \quad (4.1)$$

Introducimos una perturbación $\cos(\omega t + \phi)V$, donde V se define como

$$V = \begin{pmatrix} 0 & \mu_0 \\ \mu_0^* & 0 \end{pmatrix} = |\mu_0| \cos(\arg \mu_0) \sigma_x - |\mu_0| \sin(\arg \mu_0) \sigma_y \quad (4.2)$$

siendo

$$\sigma_x = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \quad \sigma_y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}. \quad (4.3)$$

Ahora bien, para simplificar los cálculos, vamos a tomar $\mu_0 \in \mathbb{R}$, de forma que el nuevo Hamiltoniano

pueda expresarse como:

$$H(t) = H_0 + \cos(\omega t + \phi)V = -\frac{\omega_0}{2}\sigma_z + \mu_0 \cos(\omega t + \phi)\sigma_x. \quad (4.4)$$

Como el objetivo es poder comparar si la propagación de estados calculada numéricamente es correcta, tenemos que encontrar el operador evolución temporal de forma analítica. Ahora bien, el Hamiltoniano depende explícitamente del tiempo por lo que no va a ser sencillo. Partimos de un estado cualquiera $C(t)$ que evoluciona de acuerdo con la ecuación de Schrödinger:

$$C(t) = \begin{pmatrix} C_1(t) \\ C_2(t) \end{pmatrix} \quad i\frac{d}{dt}C(t) = -\frac{\omega_0}{2}\sigma_z C(t) + \cos(\omega t + \phi)VC(t). \quad (4.5)$$

Dado que el Hamiltoniano sin perturbar, H_0 , es independiente del tiempo, y la perturbación incluida no, resulta más sencillo trabajar en la representación de interacción, ya definida en la expresión (3.7). Así, definiremos el estado $C(t)$ en la representación de interacción como:

$$c(t) = e^{iH_0 t}C(t), \quad (4.6)$$

por lo que la expresión (4.5) puede expresarse como:

$$\begin{aligned} i\frac{d}{dt}[c(t)e^{-iH_0 t}] &= H_0 c(t)e^{-iH_0 t} + \cos(\omega t + \phi)Vc(t)e^{-iH_0 t} \longrightarrow \\ &\longrightarrow H_0 c(t)e^{-iH_0 t} + ie^{-iH_0 t}\frac{d}{dt}c(t) = H_0 c(t)e^{-iH_0 t} + \cos(\omega t + \phi)Vc(t)e^{-iH_0 t} \longrightarrow. \\ &\longrightarrow ie^{-iH_0 t}\frac{d}{dt}c(t) = \cos(\omega t + \phi)Vc(t)e^{-iH_0 t} \end{aligned} \quad (4.7)$$

Dado que $[H_0, e^{eH_0 t}] = 0$ pero $[V, e^{eH_0 t}] \neq 0$, el resultado anterior se puede simplificar de la siguiente forma:

$$i\frac{d}{dt}c(t) = \cos(\omega t + \phi)\underbrace{e^{iH_0 t}Ve^{-iH_0 t}}_{V(t)}c(t), \quad (4.8)$$

donde $V(t)$ no es otra cosa que la forma de V en la representación de interacción. Así, hemos obtenido una nueva ecuación de Schrödinger, pero librándonos de la componente independiente del tiempo. Esto coincide con la expresión (3.8) comentada anteriormente.

Necesitamos la expresión matricial de $\hat{V}(t)$. Para ello debemos obtener primero la representación matricial de e^{itH_0} : como H_0 es diagonal, e^{itH_0} es trivialmente la matriz diagonal formada por las exponenciales de los elementos diagonales de H_0 , por lo que \hat{V} tomará el siguiente valor:

$$\begin{aligned} \hat{V} &= e^{iH_0 t}Ve^{-iH_0 t} = \begin{pmatrix} e^{-i\omega_0 t/2} & 0 \\ 0 & e^{i\omega_0 t/2} \end{pmatrix} \begin{pmatrix} 0 & \mu_0 \\ \mu_0 & 0 \end{pmatrix} \begin{pmatrix} e^{i\omega_0 t/2} & 0 \\ 0 & e^{-i\omega_0 t/2} \end{pmatrix} = \\ &= \begin{pmatrix} 0 & \mu_0 e^{-i\omega_0 t} \\ \mu_0 e^{i\omega_0 t} & 0 \end{pmatrix}, \end{aligned} \quad (4.9)$$

La nueva forma de la ecuación (4.8) es por lo tanto la siguiente:

$$i \frac{d}{dt} c(t) = \cos(\omega t + \phi) \begin{pmatrix} 0 & \mu_0 e^{-i\omega_0 t} \\ \mu_0 e^{i\omega_0 t} & 0 \end{pmatrix} c(t). \quad (4.10)$$

Si escribimos $\cos(\omega t + \phi) = \frac{1}{2} (e^{i(\omega t + \phi)} + e^{-i(\omega t + \phi)})$:

$$i \frac{d}{dt} c(t) = \frac{1}{2} \begin{pmatrix} 0 & \mu_0 [e^{i((\omega - \omega_0)t + \phi)} + e^{-i((\omega + \omega_0)t + \phi)}] \\ \mu_0 [e^{i((\omega + \omega_0)t + \phi)} + e^{-i((\omega - \omega_0)t + \phi)}] & 0 \end{pmatrix} c(t). \quad (4.11)$$

En este punto debemos aplicamos la *Rotating Wave Approximation*^[13] (RWA), que nos permite eliminar las oscilaciones de alta frecuencia: eliminamos los términos con $\omega + \omega_0$ y nos quedamos con los de $\omega - \omega_0$. De esta forma, la expresión anterior se reduce a:

$$i \frac{d}{dt} c(t) = \frac{1}{2} \begin{pmatrix} 0 & \mu_0 e^{i(\delta t + \phi)} \\ \mu_0 e^{-i(\delta t + \phi)} & 0 \end{pmatrix} c(t), \quad (4.12)$$

con $\delta = \omega - \omega_0$. Esta aproximación es válida cuando estamos cerca de la resonancia ($\omega = \omega_0$), y el tiempo total de acción de la perturbación es lo suficientemente largo como para descartar los términos de alta frecuencia.

Para eliminar la dependencia temporal realizamos un último cambio de representación:

$$\begin{pmatrix} d_1(t) \\ d_2(t) \end{pmatrix} = \begin{pmatrix} e^{-i\frac{\delta}{2}t} c_1(t) \\ e^{i\frac{\delta}{2}t} c_1(t) \end{pmatrix} \quad (4.13)$$

Así, la forma de la expresión (4.12) es

$$i \frac{d}{dt} d(t) = \frac{1}{2} \begin{pmatrix} \frac{\delta}{2} & \frac{\mu_0}{2} e^{i\phi} \\ \frac{\mu_0}{2} e^{-i\phi} & -\frac{\delta}{2} \end{pmatrix} d(t). \quad (4.14)$$

Si definimos \vec{n} como

$$\vec{n} = \left(\frac{1}{2} \mu_0 \cos \phi, -\frac{1}{2} \mu_0 \cos \phi, \frac{1}{2} \delta \right) \quad (4.15)$$

podemos reescribir esta expresión de la siguiente forma:

$$i \frac{d}{dt} d(t) = \vec{n} \cdot \vec{\sigma} d(t). \quad (4.16)$$

Al ser $\vec{n} \cdot \vec{\sigma}$ independiente del tiempo, podemos escribir directamente la solución como: $d(t) = \exp[-i\vec{n} \cdot \vec{\sigma} t] d(0)$. Para desarrollar la exponencial, como esta vez la matriz no es diagonal, utilizamos la siguiente identidad válida para las matrices de Pauli:

$$e^{ia(\vec{n} \cdot \vec{\sigma})t} = I \cos(at) + i(\vec{n} \cdot \vec{\sigma}) \sin(at) \quad \text{con } \vec{\sigma} = (\sigma_x, \sigma_y, \sigma_z), \quad (4.17)$$

y llegamos a:

$$d(t) = \begin{pmatrix} \cos\left(\frac{\mu t}{2}\right) - i\frac{\delta}{\mu} \sin\left(\frac{\mu t}{2}\right) & -i\frac{\mu_0}{\mu} e^{i\phi} \sin\left(\frac{\mu t}{2}\right) \\ -i\frac{\mu_0}{\mu} e^{-i\phi} \sin\left(\frac{\mu t}{2}\right) & \cos\left(\frac{\mu t}{2}\right) + i\frac{\delta}{\mu} \sin\left(\frac{\mu t}{2}\right) \end{pmatrix} \quad (4.18)$$

donde $\mu = 2|\vec{n}| = \sqrt{\delta^2 + \mu_0^2}$. Podemos ahora deshacer el cambio de variables anteriores [Eq. (4.14)], escribir la ecuación para $c(t)$, y obtener finalmente el correspondiente operador evolución:

$$U(t) = \begin{pmatrix} e^{i\frac{\delta}{2}t} \left[\cos\left(\frac{\mu t}{2}\right) - i\frac{\delta}{\mu} \sin\left(\frac{\mu t}{2}\right) \right] & e^{i\frac{\delta}{2}t} \left[-i\frac{\mu_0}{\mu} e^{i\phi} \sin\left(\frac{\mu t}{2}\right) \right] \\ e^{-i\frac{\delta}{2}t} \left[-i\frac{\mu_0}{\mu} e^{-i\phi} \sin\left(\frac{\mu t}{2}\right) \right] & e^{-i\frac{\delta}{2}t} \left[\cos\left(\frac{\mu t}{2}\right) + i\frac{\delta}{\mu} \sin\left(\frac{\mu t}{2}\right) \right] \end{pmatrix}. \quad (4.19)$$

Para comparar los cálculos realizados con los resultados numéricos, que expondremos más adelante, calculamos la probabilidad de encontrar el sistema en el estado $|2\rangle$, asumiendo que el sistema parte del estado $|1\rangle$. Así, si $|\varphi(0)\rangle = |1\rangle$, tendremos que $|\varphi(t)\rangle$ vendrá dado por:

$$\begin{aligned} |\varphi(t)\rangle &= U(t) |\varphi(0)\rangle = \\ &= \begin{pmatrix} e^{i\frac{\delta}{2}t} \left[\cos\left(\frac{\mu t}{2}\right) - i\frac{\delta}{\mu} \sin\left(\frac{\mu t}{2}\right) \right] & e^{i\frac{\delta}{2}t} \left[-i\frac{\mu_0}{\mu} e^{i\phi} \sin\left(\frac{\mu t}{2}\right) \right] \\ e^{-i\frac{\delta}{2}t} \left[-i\frac{\mu_0}{\mu} e^{-i\phi} \sin\left(\frac{\mu t}{2}\right) \right] & e^{-i\frac{\delta}{2}t} \left[\cos\left(\frac{\mu t}{2}\right) + i\frac{\delta}{\mu} \sin\left(\frac{\mu t}{2}\right) \right] \end{pmatrix} \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \\ &= e^{i\frac{\delta}{2}t} \left[\cos\left(\frac{\mu t}{2}\right) - i\frac{\delta}{\mu} \sin\left(\frac{\mu t}{2}\right) \right] |1\rangle + e^{-i\frac{\delta}{2}t} \left[-i\frac{\mu_0}{\mu} e^{-i\phi} \sin\left(\frac{\mu t}{2}\right) \right] |2\rangle, \end{aligned} \quad (4.20)$$

por lo que la probabilidad de encontrar $|\varphi(t)\rangle$ en el estado $|2\rangle$ es:

$$P_{1 \rightarrow 2}(t) = \left(\frac{\mu_0}{\mu} \right)^2 \sin^2 \left(\frac{\mu t}{2} \right) \quad \text{Con } \mu = \sqrt{(\omega - \omega_0)^2 + \mu_0^2}. \quad (4.21)$$

Esta expresión se denomina fórmula de Rabi^[13] y nos dice que la ocupación del estado $|2\rangle$ oscila con el tiempo, siendo la ocupación máxima $(\mu_0/\mu)^2$. La situación más interesante se da cuando esta ocupación vale 1, lo cual solo puede ocurrir si $\delta = 0$, es decir, $\omega = \omega_0$. Esto implica que la frecuencia de oscilación de la perturbación coincide con la frecuencia característica del sistema (diferencia de energía entre los dos niveles). A esto se le llama condición de resonancia.

De esta forma, periódicamente, la ocupación se invertirá con respecto a la inicial. Si en esta situación detenemos la perturbación habremos conseguido que el sistema pase del estado $|1\rangle$ al $|2\rangle$, lo cual se da cuando $t = t_\pi = \frac{\pi}{\mu_0}$. La perturbación resultante en este caso se denomina π -*pulse*.

Si damos valores numéricos tal que $\omega_0 = 1$ y $\mu_0 = 0.1$ podemos lanzar distintas simulaciones para distintos valores de ω y ver si coincide con los resultados teóricos. Así, si tomamos $\omega = \{0.75, 1.0, 1.1\}$ deberíamos obtener oscilaciones de la ocupación en $|2\rangle$ con frecuencia $\frac{\mu}{2} = \{0.1346, 0.050, 0.0707\}$ y valor máximo de 0.138, 1 y 0.5 respectivamente, debiendo darse la inversión total en el caso de la resonancia en 10π . El resultado obtenido se muestra en la [figura 4.1](#). Vemos que, efectivamente, el cálculo numérico se corresponde con la solución analítica.

El fenómeno de resonancia junto con el concepto de π -*pulse* puede relacionarse con otro

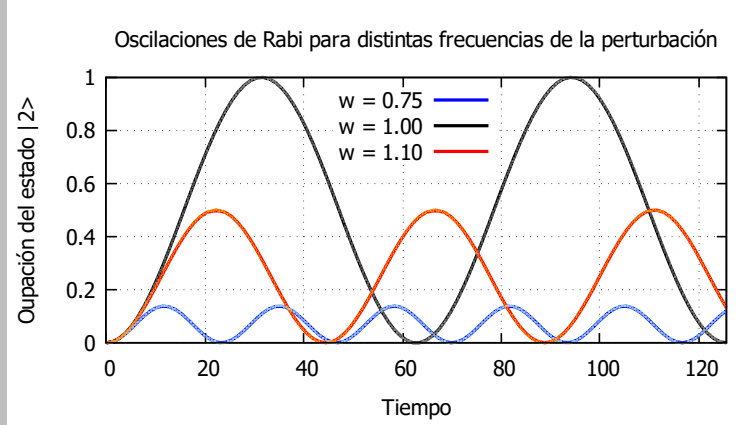


Figura 4.1: Solución numérica a las oscilaciones de Rabi para distintos valores de w .

concepto visto anteriormente: la puerta *bit-flip*, pues la aplicación de esta puerta a un q-bit invertía el estado de este. Si calculamos el operador evolución temporal de nuestro sistema perturbado en tiempo t_π obtendríamos que $|U(t_\pi) \cdot U_{bit-flip}|^2 = 1$. Veámoslo: si aplicamos a la expresión 4.19 la condición de resonancia ($\delta = 0 \rightarrow \mu = \mu_0$) obtenemos que el operador evolución temporal viene dado como:

$$U(t) = \begin{pmatrix} \cos\left(\frac{\mu t}{2}\right) & -ie^{i\phi} \sin\left(\frac{\mu t}{2}\right) \\ -ie^{-i\phi} \sin\left(\frac{\mu t}{2}\right) & \cos\left(\frac{\mu t}{2}\right) \end{pmatrix} \quad (4.22)$$

Como hemos visto antes, el pulso- π se define para $t_\pi = \frac{\pi}{\mu_0}$ por lo que en este caso la anterior ecuación se reduce a:

$$U(t_\pi) = \begin{pmatrix} 0 & -ie^{i\phi} \\ -ie^{-i\phi} & 0 \end{pmatrix} \quad (4.23)$$

Escogiendo $\phi = 0$ obtenemos que $U(t_\pi) = -iU_{bit-flip}$. Ambos operadores son equivalentes, pues únicamente se diferencian en un factor de fase. Concluimos por lo tanto que el pulso- π en un sistema de dos niveles implementa la puerta *bit-flip*.

4.2. π -pulse vs QOCT

En el apartado anterior hemos visto que los pulsos- π permiten manipular fácilmente los estados cuánticos de un sistema de dos estados. Vamos ahora a ver cómo generalizar esta idea a sistemas más grandes, por ejemplo un sistema de cuatro estados ($|0\rangle$, $|1\rangle$, $|2\rangle$, $|3\rangle$). Utilizando únicamente pulsos- π , trataremos por ejemplo de llevar el estado $|\varphi(0)\rangle = |0\rangle$ al estado $|\varphi(t_{final})\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |3\rangle)$. El Hamiltoniano que utilizaremos es:

$$H_0 = \begin{pmatrix} -2 & 0 & 0 & 0 \\ 0 & -1.2 & 0 & 0 \\ 0 & 0 & -0.5 & 0 \\ 0 & 0 & 0 & 0.5 \end{pmatrix}, \quad (4.24)$$

y está escogido para que las frecuencias características sean distintas entre sí:

$$\omega_1 = E_2 - E_1 = 0.8 \quad (4.25)$$

$$\omega_2 = E_3 - E_2 = 0.7 \quad (4.26)$$

$$\omega_3 = E_4 - E_3 = 1.0 \quad (4.27)$$

La forma de la perturbación que hemos usado es:

$$V = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \quad (4.28)$$

Es decir, asumimos que los estados solo se acoplan con sus niveles vecinos. Además, la magnitud del acoplo es igual para todos.

En el apartado anterior se ha probado que es posible usar pulsos- π para modificar de manera controlada el estado cuántico del sistema. Como solo había dos niveles de energía no existía el riesgo de que el estado final se contaminase con otros estados del sistema. Sin embargo, en este caso tenemos cuatro niveles de energía distintos y queremos que el estado final sea una superposición de únicamente dos de ellos. Para encontrar la perturbación adecuada de forma analítica usaremos el fenómeno de la resonancia: si la longitud de los pulsos- π es lo suficientemente grande (esto equivale a que la amplitud de la perturbación sea pequeña) solo se verán afectados los dos niveles con la separación de energía adecuada (ω_i). Esta idea es la misma en la que se fundamenta la RWA.

De esta forma, podemos alterar la ocupación de estados contiguos usando la aproximación a dos niveles sin miedo de que el estado final se vea contaminado por niveles de energía no deseados. Así, por ejemplo, para inducir la transición deseada, podemos emplear el siguiente protocolo: Si inicialmente el sistema está en el estado $|0\rangle$, podemos disminuir la ocupación de este estado mediante la aplicación de un pulso de frecuencia ω_1 y debemos aplicarlo hasta que la ocupación del estado $|0\rangle$ sea la deseada, en nuestro caso 0.5. A partir de los resultados mostrados en el apartado anterior sabemos que para una perturbación sinusoidal de frecuencia ω y amplitud λ se producirá la inversión total si $\omega = \omega_1$ y aplicamos el pulso un tiempo $t_\pi = \frac{\pi}{\lambda}$ por lo que, si queremos que la ocupación sea del 50 % para cada estado, habrá que aplicarlo la mitad del tiempo. A esto suele denominarse pulso- $\frac{\pi}{2}$ y denotaremos el tiempo de aplicación como $t_{\frac{\pi}{2}} = \frac{\pi}{2\lambda}$.

Tras aplicar este pulso tendremos que $|\varphi(t_{\frac{\pi}{2}})\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$. El siguiente paso consiste en intercambiar el estado $|1\rangle$ por el $|2\rangle$. Esto es sencillo de hacer aplicando un pulso- π de frecuencia ω_2 un tiempo t_π . Lo mismo para pasar del $|2\rangle$ al $|3\rangle$.

En resumen, la perturbación que hay que aplicar a H_0 será, en base únicamente al concepto de pulso- π :

$$W(t) = \begin{cases} \lambda \sin(\omega_1 t) V, & \text{si } t < t_{\frac{\pi}{2}} \\ \lambda \sin(\omega_2 t) V, & \text{si } t_{\frac{\pi}{2}} < t < 3t_{\frac{\pi}{2}} \\ \lambda \sin(\omega_3 t) V, & \text{si } 3t_{\frac{\pi}{2}} < t < 5t_{\frac{\pi}{2}} \end{cases} \quad (4.29)$$

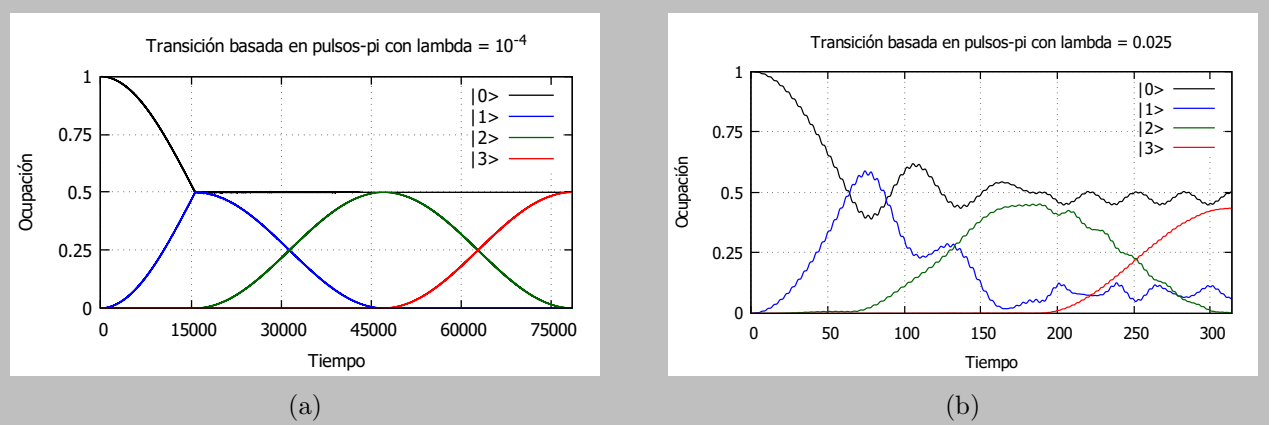


Figura 4.2: Ocupación de cada nivel en una transición del estado $|\varphi(0)\rangle = |0\rangle$ al $\frac{1}{\sqrt{2}}(|0\rangle + |3\rangle)$ mediante pulsos- π para (a) $\lambda = 10^{-4}$ y para (b) $\lambda = 0.025$

En un primer cálculo, hemos tomado la amplitud $\lambda = 10^{-4}$. Representamos la ocupación en función del tiempo en la [figura 4.2a](#). Vemos que, efectivamente, el estado final es el deseado y, en concreto, las ocupaciones de los estado $|0\rangle$ y $|3\rangle$ son, respectivamente, 0.50005 y 0.49995. Ahora bien, como hemos comentado, interesa que las puertas lógicas sean lo más rápidas posibles pues los estados cuántico no permanecen eternamente. Si queremos acelerar las puertas pero utilizando este tipo de protocolos basados en pulsos- π , debemos aumentar el valor de λ , que es el que regula el tiempo de aplicación de la perturbación. Si tomamos $\lambda = 0.025$, por ejemplo, tal que $t_{final} = 10\pi$, obtenemos la gráfica mostrada en la [figura 4.2b](#). El resultado obtenido es claramente peor: se observan oscilaciones en la ocupación debido a la contaminación del resto de estados^[14]. La ocupación final de los estados $|0\rangle$ y $|3\rangle$ es 0.504 y 0.434 respectivamente.

Para poder obtener resultados comparables a los de la [figura 4.2a](#) pero con tiempos de transición similares a los mostrados en la [figura 4.2b](#), debemos utilizar pulsos más complejos que los pulsos- π . Para encontrar tales pulsos, hemos utilizado la teoría de control óptimo desarrollada anteriormente. Utilizaremos el mismo tipo de perturbación que en la expresión (4.29); sin embargo, en lugar de aplicar cada pulso uno detrás de otro y con la misma amplitud, se permitirá que varíe tanto la amplitud (entre 0.0 y 0.1) como la fase (entre $-\pi$ y π) y se aplicarán simultáneamente. Además, no solo se usarán pulsos de frecuencia ω_i (siendo $i = \{1, 2, 3\}$), sino también frecuencias proporcionales a estas, en concreto, $2\omega_i$ y $0.5\omega_i$. La perturbación resultante deja de ser, obviamente, un pulso- π . Su forma genérica es:

$$f(\{u_i\}, t) = \sum_{m=1}^3 [u_m \sin(\omega_m t + u_{m+9}) + u_{m+3} \sin(2\omega_m t + u_{m+12}) + u_{m+6} \sin(0.5\omega_m t + u_{m+15})]. \quad (4.30)$$

Frecuencia	Amplitud $\cdot 10^{-3}$	Fase $\cdot 10^{-2}$
ω_1	71.18	0.039
ω_2	83.66	-314.16
ω_3	50.34	313.95
$2\omega_1$	100.00	314.16
$2\omega_2$	100.00	312.62
$2\omega_3$	100.00	314.16
$0.5\omega_1$	0.00	-117.97
$0.5\omega_2$	2.87	91.13
$0.5\omega_3$	78.66	0.066

Tabla 4.1: coeficientes óptimos de la perturbación aplicada en la transición de $|\varphi(0)\rangle$ a $|\varphi(t)\rangle$

La tarea del algoritmo de optimización basado en la QOCT es, por lo tanto, encontrar los parámetros $\{u_i\}$ óptimos que inducen la transición deseada. Estos se incluyen en la [tabla 4.1](#), ordenados para cada uno de los pulsos y separando los parámetros de amplitud y fase.

Como curiosidad, la frecuencia $0.5\omega_1$ presenta amplitud nula: realmente, la perturbación final se compone de la suma de ocho funciones sinusoidales y no nueve. Si ahora hacemos lo mismo que en los dos casos anteriores y representamos la evolución de la ocupación de cada uno de los estados con el tiempo obtenemos la gráfica mostrada en la [figura 4.3](#). Si bien es cierto que la transición entre estados es más compleja (pues la propia perturbación también lo es), debemos centrarnos en el objetivo: el estado final objetivo, sin contaminación de estados indeseados. Las ocupaciones de los estados $|0\rangle$ y $|3\rangle$ son 0.496 y 0.503: aunque no son tan buenos como en el caso de la [figura 4.2a](#), son mejores que los de la [figura 4.2b](#), pues la ocupación de los otros dos estados es relativamente baja ($7.003 \cdot 10^{-4}$ para el estado $|1\rangle$ y $8.875 \cdot 10^{-5}$ para el $|2\rangle$).

Concluimos por lo tanto que esta metodología basada en la QOCT permite encontrar pulsos complejos capaces de acelerar los tiempos de transición, con respecto a los tiempos necesarios mediante los protocolos basados en pulsos- π .

5. Elección de parámetros y métodos de optimización

Ahora que se ha confirmado que la teoría de control óptimo es el camino más adecuado a seguir debemos fijar una serie de parámetros que pueden afectar a los resultados obtenidos y al tiempo de computación para llegar a ellos. Estos son: (1) el algoritmo de optimización a usar, (2) los rangos en los que el programa puede modificar los parámetros de control, y (3) el tiempo total de aplicación de la perturbación. Los siguientes apartados se centran en cada uno de estos aspectos.

5.1. Algoritmos de optimización

NLopt es una biblioteca que incluye diversos algoritmos de optimización, tanto globales como locales. Nos hemos centrado en los algoritmos de optimización globales. El objetivo de esta sección es determinar cuál de todos ellos es el más eficiente, pues este será el que usaremos para el resto de

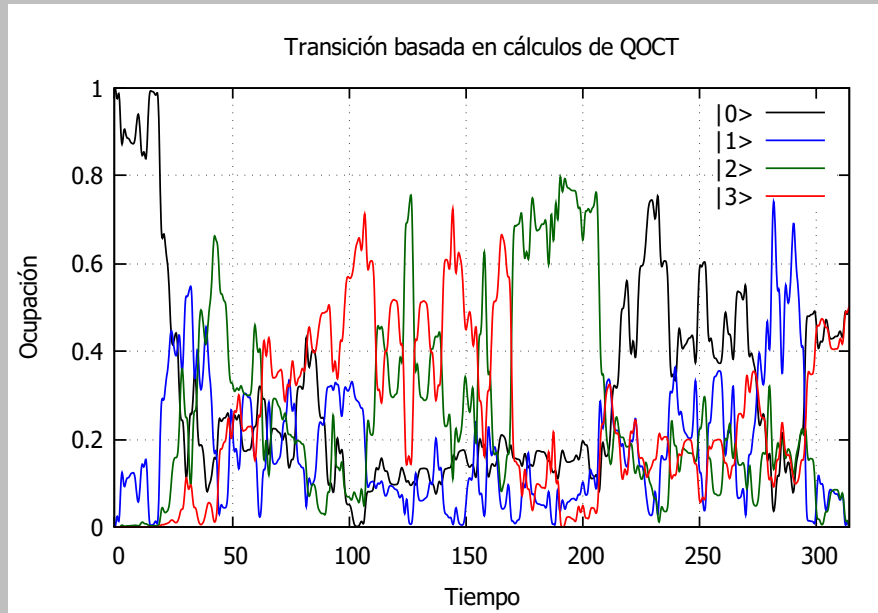


Figura 4.3: Ocupación de cada nivel en una transición del estado $|0\rangle$ al $\frac{1}{\sqrt{2}}(|0\rangle + |3\rangle)$ mediante una perturbación obtenida con QOCT.

las simulaciones.

Tras realizar distintas pruebas para un mismo problema, pero utilizando distintos algoritmos, se ha obtenido que los más eficientes son los que se mencionaron en el apartado introductorio: MLSDL, CRS-LM, ISRES y ESCH. Para poder comparar cada uno de estos algoritmos de forma cómoda NLOpt incluye una función que permite detener la búsqueda de parámetros cuando el valor de la función a optimizar llega a un determinado valor. Así, en base al cálculo de J [Eq. (3.9)], hacemos que el algoritmo se detenga para tres valores distintos de éste (recordando que el óptimo es cuando $J = 1$). En este caso, el problema a resolver es la generación de una puerta Control-phase, con $m = 6$ y $T = 100\pi$.

- Si detenemos el algoritmo de optimización cuando $J > 0.995$ obtenemos los resultados que se muestran en la [tabla 5.1](#). En este caso solo dos algoritmos han llegado al límite impuesto: ESCH y MLSL, siendo el primero el que menos tiempo (e iteraciones) ha requerido. Mientras tanto ISRES no ha conseguido alcanzar un valor de J superior a 0.995 y por ello se ha detenido a las 10000 iteraciones, otra de las condiciones de parada. Por último CRS tampoco ha llegado al valor de 0.995 y se ha detenido antes incluso de llegar al límite de iteraciones. Esto se debe a que otro criterio de parada se basa en el estudio de la variación de J entre iteraciones de forma que, si esta es menor que un determinado valor, el proceso de optimización se detiene pues detecta que se ha quedado atrapado en un mínimo (local o no).
- Si detenemos la optimización cuando $J > 0.9995$ el resultado, por lo que respecta a criterios

Algoritmo	Iteraciones	Tiempo [s]	J
CRS-LM	8027	390	0,974891
MLSL-LDS	3416	759	0,995946
ISRES	10000	475	0,964021
ESCH	2922	138	0,995453

Tabla 5.1: Resultados del proceso de optimización deteniendo el programa para $J > 0.995$

Algoritmo	Iteraciones	Tiempo [s]	J
CRS-LM	4540	220	0,928494
MLSL-LDS	4769	1044	0,999601
ISRES	10000	484	0,953946
ESCH	1995	96	0,999505

Tabla 5.2: Resultados del proceso de optimización deteniendo el programa para $J > 0.9995$

de parada y orden de tiempo de cálculo es similar al caso anterior. ESCH y MLSL son los únicos en alcanzar la condición de J siendo los otros dos algoritmos incapaces de lograrlo. Para estos últimos los resultados son incluso peores que en el caso anterior: CRS ha finalizado la optimización, de nuevo, en un mínimo local (los valores numéricos se muestran en la [tabla 5.2](#)).

- Por último, si el programa se detiene para $J > 0.99999$ ninguno de los algoritmos llega al objetivo (posiblemente porque el número de frecuencias sea demasiado pequeño y ni siquiera sea posible). Ahora bien, el que mejor resultado ha obtenido es MLSL siendo también el que más tiempo ha requerido. Por tercera vez, el único algoritmo que ha terminado de forma prematura es CRS.

Tras analizar estos resultados vemos que, dependiendo de cómo de buena queremos que sea nuestra puerta lógica, nos interesa usar un algoritmo u otro: para valores de J bajos, el que mejor resultado ha dado es ESCH; en cambio, para valores altos de J , el mejor resultado ha sido obtenido por el algoritmo MLSL. Cabe destacar, además, que tras lanzar distintas simulaciones se ha observado que, para valores de J muy cercanos a 1 (los que nos interesa obtener) el único algoritmo que ha mostrado una reproducibilidad de los resultados obtenidos en distintas simulaciones es MLSL.

Algoritmo	Iteraciones	Tiempo [s]	J
CRS-LM	5724	275	0,979969
MLSL-LDS	10000	2248	0,999962
ISRES	10000	483	0,963453
ESCH	10000	496	0,999769

Tabla 5.3: Resultados del proceso de optimización deteniendo el programa para $J > 0.99999$

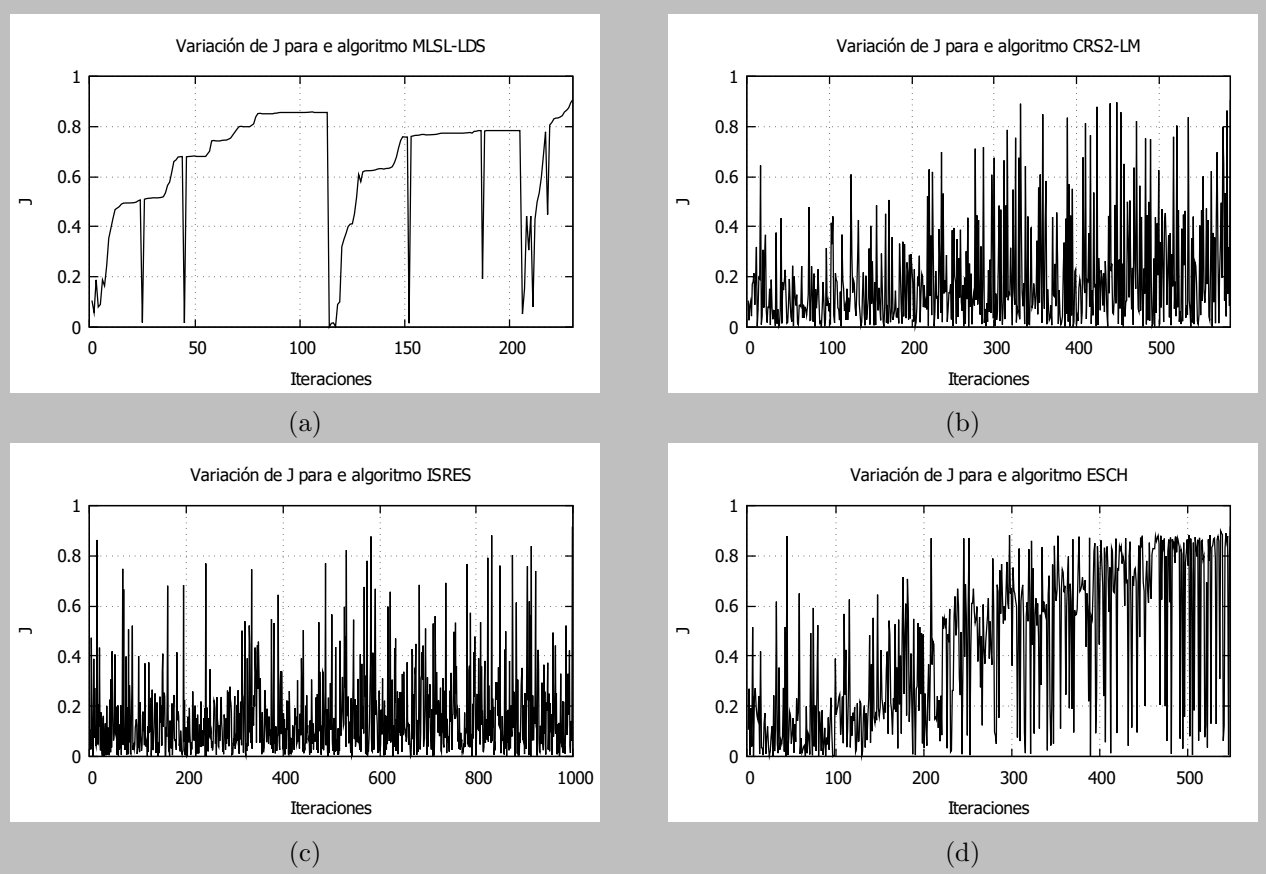


Figura 5.1: Variación de J con el número de iteraciones para los cuatro algoritmos estudiados. En este caso el programa se detiene cuando se llega a un valor de $J > 0.9$. (a) muestra como varía J para el algoritmo MLSL, (b) para CRS, (c) para ISRES y (d) para ESCH.

Otros algoritmos, como ESCH o CRS, presentan distintos resultados, a pesar de reiniciarse con los mismo parámetros iniciales. Ello implica que tienen mayor tendencia a quedarse atrapados en mínimos locales. Concluimos por lo tanto que conviene escoger MLSL para realizar las simulaciones finales pues, incluso para parámetros iniciales aleatorios suele llegar a resultados similares.

En la [figura 5.1](#) mostramos la variación de J con el número de iteraciones. Uno de los fenómenos más destacables que se observan son las distintas optimizaciones locales que realiza el algoritmo MLSL antes de llegar al valor de J requerido.

Antes de finalizar el estudio de estos algoritmos cabe comentar el hecho de por qué MLSL es el que más tiempo de cálculo requiere. Tal como se dijo en el apartado introductorio éste es el único algoritmo que se basa en el uso del gradiente por lo que en cada iteración se calcula el resultado de la expresión (3.13), lo que requiere una propagación del coestado desde tiempo T hasta tiempo 0. Esto supone doblar el número de propagaciones que normalmente se requieren en otros algoritmos para calcular la forma del pulso, lo cual se refleja en un aumento del tiempo de cálculo.

5.2. Amplitudes y tiempo de propagación

Existen determinados parámetros que vamos a fijar en las optimizaciones: El número de frecuencias incluidas (m), el tiempo de aplicación de la perturbación (T) y el rango de variación de las amplitudes (que irá desde 0 hasta A). También tenemos el rango para las fases pero este va siempre de π a $-\pi$ y no tendría sentido restringirlo. En este apartado vamos a determinar unos valores razonables para el valor de T y de A , pues la búsqueda del mejor valor de m se realizará de forma independiente para cada una de las puertas lógicas que modelizaremos.

Para ello se han lanzado distintas simulaciones que tienen como objetivo encontrar la perturbación más adecuada para generar la puerta CNOT, recordando siempre que la perturbación ideal es aquella que requiere del menor número de frecuencias distintas, de la menor amplitud posible (ya que experimentalmente es más sencillo trabajar con pulsos simples y amplitudes bajas), y del mínimo tiempo de propagación (esto es así porque los estados cuánticos no se mantienen eternamente y, si el pulso es demasiado largo, puede que el estado en si desaparezca antes incluso de que el pulso termine de aplicarse). Para asegurarnos de que la bondad de cada combinación de A y T no depende de m se han realizado 12 simulaciones distintas para cada combinación de A y T . Los posibles valores de m y las frecuencias incluidas se dividen en tres tipos: los que solo incluyen una única frecuencia natural del sistema (obviamente existen 3 posibilidades para este tipo de perturbación), los que incluyen las tres frecuencias naturales y proporcionales a estas (de las que se han probado 4 posibilidades) y las que incluyen las mismas tres frecuencias naturales y proporcionales solo a ω_3 , pues es la que mejor resultado ha dado en el caso de una sola frecuencia para la puerta CNOT. De este último tipo se han probado 5 posibilidades.

Así, primero se ha fijado el valor de T a 100π y se han lanzado varias simulaciones disminuyendo cada vez más el valor máximo de A (0.5, 0.1 y 0.01). Analizando los resultados de la [tabla 5.4](#) se deduce que, si incluimos una única frecuencia, es mejor el caso de $A = 0.5$. Sin embargo, si añadimos más (para acercarnos a 1), los resultados son bastante similares con el caso de $A = 0.1$. Dado que para $A = 0.01$ el resultado es claramente peor, el mejor valor de A en este caso es 0.1 que, de momento, presenta el menor valor de A con el mayor valor de J .

Para determinar si el valor de T es también el más adecuado se lanza un nuevo conjunto de simulaciones fijando A en 0.1 y variando el tiempo de aplicación del pulso. En concreto, los valores que se han probado para T son 10π , 20π y 1000π . La única combinación que se acerca a los resultados del caso anterior ($A = 0.1$, $T = 100\pi$) es la última, y solo es así porque el tiempo de aplicación de la perturbación es absurdamente mayor, lo cual supone un tiempo de cálculo bastante más largo. Podríamos aumentar la amplitud para mejorar los resultados, pero como tampoco queremos que sea muy grande nos quedamos con la combinación inicial. Destacar que los resultados en los que $J > 1$ son producto del error numérico, que es del orden de 10^{-5} pues, por la definición del producto de Fröbenius, J nunca puede ser mayor que uno.

6. Resultados

Una vez establecido el procedimiento (algoritmo de optimización, límites para las amplitudes, fases, frecuencias, etc.) fijamos el rango en la optimización de la amplitud en $A = 0.1$ y tomamos un

frecuencias	$T = 100\pi$			$A = 0, 1$		
	$A = 0, 5$	$A = 0, 1$	$A = 0, 01$	$T = 10\pi$	$T = 20\pi$	$T = 1000\pi$
ω_1	0,352788	0,335134	0,250000	0,250000	0,518830	0,483822
ω_2	0,701796	0,315768	0,250000	0,250000	0,356150	0,406484
ω_3	0,846953	0,633009	0,510536	0,578666	0,551055	0,904985
ω_i	0,997450	0,989045	0,510620	0,587258	0,728764	0,995046
$\omega_i, 2\omega_i$	0,999380	0,998755	0,518394	0,654451	0,874045	0,996637
$\omega_i, 2\omega_i, 0.5\omega_i$	0,999974	1,000003	0,550214	0,980311	0,988027	1,000019
$\omega_i, 2\omega_i, 0.5\omega_i, 3\omega_i$	0,999945	0,999827	0,550183	0,980241	0,992616	0,999245
$\omega_i, 2\omega_3$	0,998456	0,998868	0,512167	0,604162	0,776458	0,997579
$\omega_i, 2\omega_3, 0.5\omega_3$	0,999997	0,999874	0,512208	0,705163	0,836853	0,997319
$\omega_i, 2\omega_3, 0.5\omega_3, 3\omega_3$	1,000009	0,998794	0,512800	0,722267	0,836897	0,997441
$\omega_i, 2\omega_3, 0.5\omega_3, 3\omega_3, 0, 25\omega_3$	0,999599	0,999878	0,526544	0,870850	0,962048	0,997645
$\omega_i, 2\omega_3, 0.5\omega_3, 3\omega_3, 0, 25\omega_3, 4\omega_3$	0,999982	0,999883	0,526436	0,872760	0,964550	0,998112

Tabla 5.4: Valor obtenido de J para para distintas combinaciones de A y T . ω_i hace referencia a las tres frecuencias naturales del sistema pues $i = \{1, 2, 3\}$

tiempo total de propagación de $T = 100\pi$. Usando el algoritmo MLSL, lanzamos varios procesos de optimización con distintas combinaciones de frecuencias (las usadas en la [tabla 5.4](#)) para buscar los parámetros óptimos que modelizan cada una de las puertas. Igual que en el caso anterior, la forma matricial del hamiltoniano no perturbado está dada por la Eq. (4.24), mientras que la perturbación V está dada por la Eq. (4.28). Se incluyen a continuación los resultados obtenidos para cada una de las puertas analizadas.

6.1. Puerta Controlled-NOT

En el apartado de introducción a la computación cuántica ya se ha explicado el funcionamiento de la puerta CNOT, por lo que pasamos directamente a la exposición de los resultados. La perturbación óptima se da para $m = 7$, apareciendo en la [tabla 6.1](#) la combinación de frecuencias resultante, que da lugar a un valor de J esencialmente igual a uno. El hecho de que J sea levemente mayor que uno se debe, como hemos dicho antes, a error numérico. Destacar que, de las tres frecuencias naturales del sistema, es ω_3 la de mayor amplitud, pues esta se asocia con la diferencia de energía entre los estados $|2\rangle$ y $|3\rangle$, que son los únicos que se ven afectados al aplicar la puerta CNOT. La [figura 6.1](#) incluye la representación del pulso en su tiempo de aplicación.

Frecuencia	Amplitud $\cdot 10^{-3}$	Fase
ω_1	20.94	0.54
ω_2	20.07	-1.81
ω_3	92.37	0.54
$2\omega_1$	100.00	2.08
$2\omega_2$	60.63	-1.97
$0.5\omega_1$	53.84	2.82
$0.5\omega_3$	91.32	-0.84

Tabla 6.1: Parámetros del pulso para generar una puerta CNOT

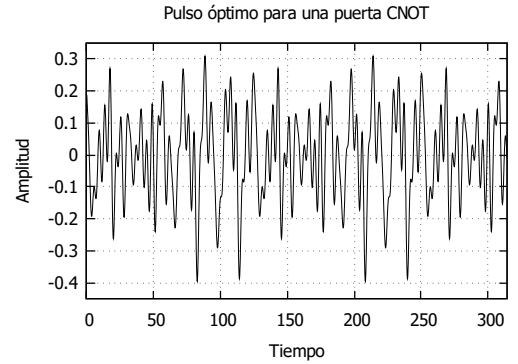
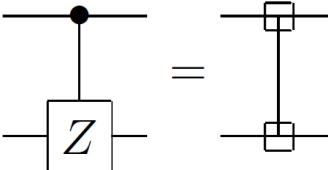


Figura 6.1: Perturbación aplicada para generar la puerta CNOT

6.2. Puerta Controlled-Z

La puerta Control-Z está definida por:

$$\begin{aligned}
 |0\rangle &= |00\rangle \longrightarrow |00\rangle = |0\rangle \\
 |1\rangle &= |01\rangle \longrightarrow |01\rangle = |1\rangle \\
 |2\rangle &= |10\rangle \longrightarrow |10\rangle = |2\rangle \\
 |3\rangle &= |11\rangle \longrightarrow -|11\rangle = -|3\rangle
 \end{aligned}$$



$$U_{\text{Control-Z}} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \end{pmatrix}$$

Así, si el primer q-bit está en el estado $|0\rangle$, el estado en el que se encuentra el segundo q-bit no se ve alterado. Sin embargo, si el primer q-bit está en el estado $|1\rangle$ se aplica σ_z al segundo q-bit, por ello a esta puerta se le denomina Control-Z. Vemos que la representación matricial es diagonal y que, por tanto, la ocupación de cada nivel tras la aplicación de esta puerta no cambiará, pues esta es el módulo al cuadrado del coeficiente asociado a cada nivel en el estado cuántico.

Los valores $\{u_i\}$ obtenidos se incluyen en la [tabla 6.2](#), que dan lugar a la puerta deseada con $J = 1.000007$. Como vemos, en este caso $m = 5$, menor que para la puerta CNOT, debido, posiblemente, a que al no tener que modificar la ocupación como tal sea más sencillo generarla. Además, las amplitudes de cada una de las frecuencias también son menores que en el caso de la CNOT. La gráfica que muestra el pulso resultante se incluye en la [figura 6.2](#).

Frecuencia	Amplitud $\cdot 10^{-3}$	Fase
ω_1	14.46	0.59
ω_2	60.69	-2.07
ω_3	0.127	2.57
$2\omega_2$	31.59	-2.92
$3\omega_2$	78.33	-1.25

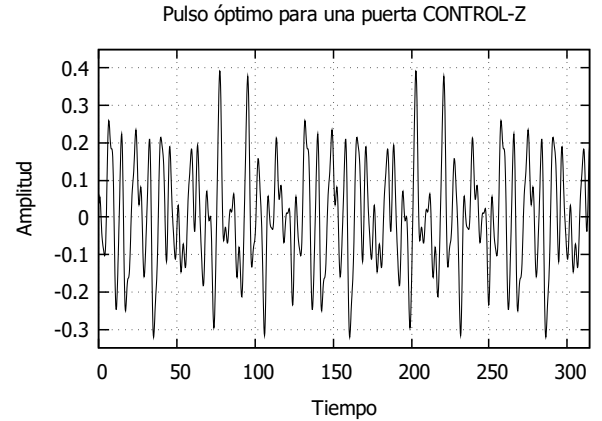


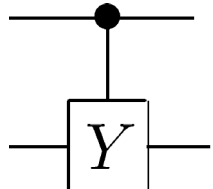
Tabla 6.2: Parámetros del pulso para generar una puerta CONTROL-Z

Figura 6.2: Perturbación aplicada para generar la puerta CONTROL-Z

6.3. Puerta Controlled-Y

La puerta Control-Y se caracteriza de la siguiente forma:

$$\begin{aligned}
 |0\rangle &= |00\rangle \longrightarrow |00\rangle = |0\rangle \\
 |1\rangle &= |01\rangle \longrightarrow |01\rangle = |1\rangle \\
 |2\rangle &= |10\rangle \longrightarrow i|11\rangle = i|3\rangle \\
 |3\rangle &= |11\rangle \longrightarrow -i|10\rangle = -i|2\rangle
 \end{aligned}$$



$$U_{\text{Control-Y}} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & i \\ 0 & 0 & -i & 0 \end{pmatrix}$$

Básicamente se basa en el mismo principio que la puerta Control-Z siendo la principal diferencia el hecho de que si el primer q-bit se encuentra en el estado $|1\rangle$ se aplica la matriz de Pauli σ_y en lugar de σ_z . Esto hace que la representación matricial deje de ser diagonal y se requiera una perturbación con m mayor, en concreto $m = 7$. Los parámetros obtenidos aparecen en la [tabla 6.3](#) y el valor de J resultante de estos es $J = 1.000002$. Al igual que ocurría con la puerta CNOT ES ω_3 la frecuencia natural de mayor amplitud, pues está asociada a los estados cuya ocupación se ve modificada. El pulso resultante se muestra en la [figura 6.3](#).

Frecuencia	Amplitud $\cdot 10^{-3}$	Fase
ω_1	21.98	-0.36
ω_2	19.89	-2.67
ω_3	93.77	-1.93
$2\omega_1$	100.00	1.43
$0.5\omega_1$	86.79	3.14
$0.5\omega_3$	68.89	-1.16

Tabla 6.3: Parámetros del pulso para generar una puerta CONTROL-Y

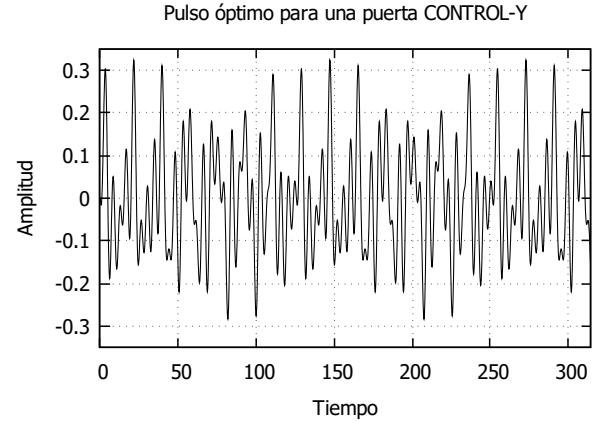
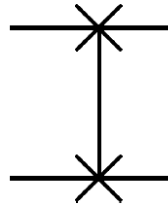


Figura 6.3: Perturbación aplicada para generar la puerta CONTROL-Y

6.4. Puerta SWAP

La puerta SWAP se caracteriza de la siguiente forma:

$$\begin{aligned}
 |0\rangle &= |00\rangle \longrightarrow |00\rangle = |0\rangle \\
 |1\rangle &= |01\rangle \longrightarrow |10\rangle = |2\rangle \\
 |2\rangle &= |10\rangle \longrightarrow |01\rangle = |1\rangle \\
 |3\rangle &= |11\rangle \longrightarrow |11\rangle = |3\rangle
 \end{aligned}$$



$$U_{\text{SWAP}} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

La utilidad de esta puerta se basa en el hecho de que permuta los estados cuánticos de los dos q-bits de entrada. En estados puros de dos q-bits esto se traduce en que los estados $|0\rangle$ y $|3\rangle$ quedan inalterados mientras que el $|1\rangle$ y el $|2\rangle$ se cambian entre si. La perturbación necesaria en este caso es de $m = 9$ (la mayor hasta ahora) y, a diferencia de las puertas anteriores, las tres amplitud de las componentes asociadas a las frecuencias naturales del sistema son bastante grandes, es decir, no existe una que claramente predomine sobre las otras dos. Estos parámetros, así como la representación del pulso obtenido, se incluyen en la [tabla 6.4](#) y [gráfica 6.4](#) respectivamente. El valor obtenido para J es en este caso 1.00003.

Frecuencia	Amplitud $\cdot 10^{-3}$	Fase
ω_1	69.13	0.43
ω_2	78.10	0.89
ω_3	99.96	1.47
$2\omega_1$	100.00	-0.83
$2\omega_2$	0.76	-1.41
$2\omega_3$	100.00	-2.58
$0.5\omega_1$	74.41	2.27
$0.5\omega_2$	0.02	2.08
$0.5\omega_3$	63.24	2.29

Tabla 6.4: Parámetros del pulso para generar una puerta SWAP

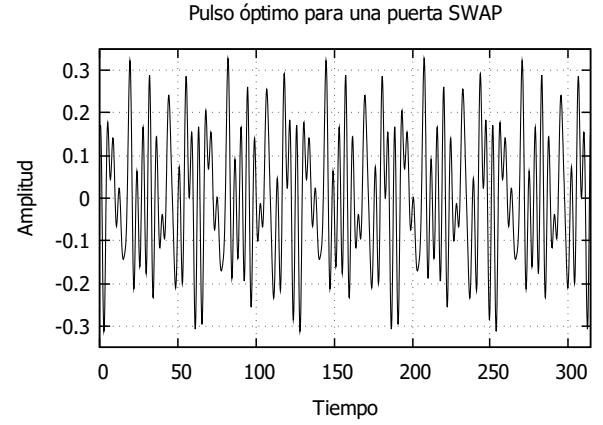
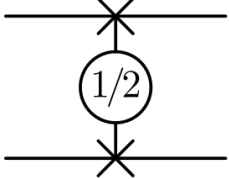


Figura 6.4: Perturbación aplicada para generar la puerta SWAP

6.5. Puerta $\sqrt{\text{SWAP}}$

Una variante de la puerta SWAP es la puerta $\sqrt{\text{SWAP}}$. Su nombre se debe a que la aplicación sucesiva de dos puertas de este tipo equivalen a la aplicación de una puerta SWAP, es decir: $\sqrt{\text{SWAP}}\sqrt{\text{SWAP}} = \text{SWAP}$. La expresión matricial será, por tanto, la siguiente:

$$\begin{aligned}
 |0\rangle &\longrightarrow |0\rangle \\
 |1\rangle &\longrightarrow \frac{1}{2}[(1+i)|1\rangle + (1-i)|2\rangle] \\
 |2\rangle &\longrightarrow \frac{1}{2}[(1-i)|1\rangle + (1+i)|2\rangle] \\
 |3\rangle &\longrightarrow |3\rangle
 \end{aligned}$$



$$U_{\sqrt{\text{SWAP}}} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \frac{1}{2}(1+i) & \frac{1}{2}(1-i) & 0 \\ 0 & \frac{1}{2}(1-i) & \frac{1}{2}(1+i) & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Teniendo en cuenta que el pulso de la puerta SWAP es, hasta ahora, el que más frecuencias ha necesitado que incluyamos, es esperable que la puerta $\sqrt{\text{SWAP}}$ no sea una excepción y efectivamente no lo es, pues se ha requerido un pulso de $m = 14$ (cuyos parámetros se incluyen en la [tabla 6.5](#)) para que $J = 1.00009$. Igual que ocurría antes no existe ninguna frecuencia natural que predomine sobre las demás. La representación gráfica del pulso con el tiempo se incluye en la [tabla 6.5](#).

Frecuencia	Amplitud $\cdot 10^{-3}$	Fase
ω_1	77.28	-0.79
ω_2	67.60	2.47
ω_3	66.03	-1.70
$2\omega_2$	100.00	-1.70
$2\omega_3$	100.00	-2.46
$0.5\omega_1$	0.007	1.40
$0.5\omega_2$	99.83	2.18
$0.5\omega_3$	78.60	-0.94
$3\omega_1$	100.00	-1.23
$3\omega_2$	87.54	0.40
$3\omega_3$	100.00	-1.23
$0.25\omega_1$	75.42	1.92
$0.25\omega_2$	81.96	-0.16
$0.25\omega_3$	72.17	3.14

Tabla 6.5: Parámetros del pulso para generar una puerta $\sqrt{\text{SWAP}}$

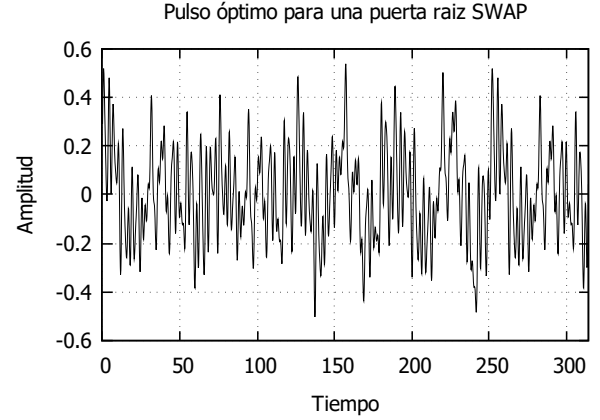
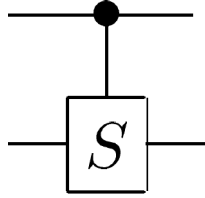


Figura 6.5: Perturbación aplicada para generar la puerta $\sqrt{\text{SWAP}}$

6.6. Controlled-phase

La última puerta que se va a modelar es la Control-Phase, su funcionamiento se muestra a continuación:

$$\begin{aligned}
|0\rangle &= |00\rangle \longrightarrow |00\rangle = |0\rangle \\
|1\rangle &= |01\rangle \longrightarrow |01\rangle = |1\rangle \\
|2\rangle &= |10\rangle \longrightarrow |10\rangle = |2\rangle \\
|3\rangle &= |11\rangle \longrightarrow i|11\rangle = i|3\rangle
\end{aligned}$$



$$\text{Control-}\varphi = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & i \end{pmatrix}$$

Al igual que todas las puertas de Control solo aplica un operador determinado al segundo q-bit si el primero esta en el estado $|1\rangle$. En este caso aplica el operador R_ϕ con $\phi = \frac{\pi}{2}$ definido como

$$R_\phi = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\phi} \end{pmatrix}. \quad (6.1)$$

Si $\phi = \frac{\pi}{4}$ se denomina puerta $\frac{\pi}{4}$ mientras que si $\phi = \pi$ se corresponde con la puerta Control-Z. Al igual que ocurría con la puerta Control-Z la probabilidad de medir cada estado no cambia al aplicarla. En su lugar equivale a “rotar” el segundo q-bit. Los parámetros resultantes de la optimización se incluyen en la [tabla 6.5](#) mientras que la representación del pulso aparece en la [gráfica 6.5](#). El valor de J obtenido con este pulso es 1.00001.

Frecuencia	Amplitud $\cdot 10^{-3}$	Fase
ω_1	16.37	0.96
ω_2	86.21	-1.49
$2\omega_1$	74.31	-2.98
$2\omega_2$	78.26	-0.85
$2\omega_3$	100.00	0.13

Tabla 6.6: Parámetros del pulso para generar una puerta CONTROL-Phase

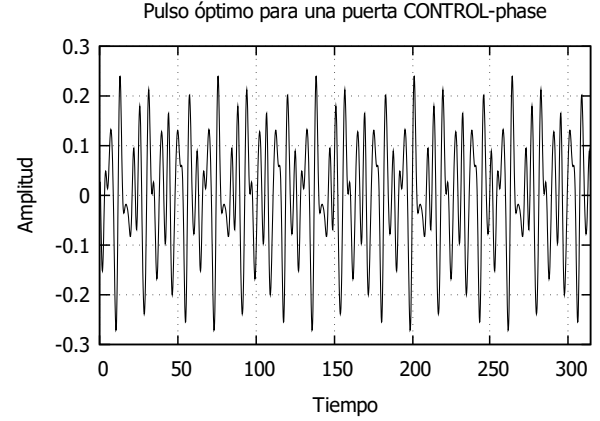


Figura 6.6: Perturbación aplicada para generar la puerta CONTROL-Phase

7. Conclusiones

A lo largo de este documento se ha conseguido mostrar que la teoría de control óptimo permite, efectivamente, calcular la forma de las perturbaciones que debemos introducir en un sistema de 4 niveles para que este se comporte como una puerta lógica. Hemos establecido unas formas funcionales determinadas para los pulsos. En este trabajo se han establecido de forma arbitraria, pero cabe la posibilidad de hacerlo de manera que sean compatibles con las posibilidades experimentales: en determinados casos los pulsos resultantes pueden ser bastante complejos y no realizables en la práctica. Si es posible aumentar el tiempo de aplicación de la perturbación o la amplitud máxima de ésta, puede merecer la pena a cambio de simplificar el pulso.

Asimismo, hemos analizado la eficiencia de varios algoritmos de optimización global, que forman parte del procedimiento de la teoría de control óptimo cuántico. Hemos concluido que, para el tipo de problemas estudiados, el algoritmo MLSL [7] ha resultado ser el más adecuado.

Hemos visto también que, aumentando el tiempo de aplicación de la perturbación, es más fácil obtener pulsos que realicen la puerta cuántica requerida con un número menor de frecuencias. Si aumentamos el valor máximo de la amplitud a la hora de buscar los parámetros óptimos sucede lo mismo pero, de nuevo, la posibilidad de aumentar la amplitud debe ajustarse a la capacidad experimental.

En este caso solo se han buscado los pulsos necesarios para generar puertas de 2 q-bits. Sin embargo, el mismo proceso se puede seguir para las de 1 q-bit (como se ha mostrado en el caso de las oscilaciones de Rabi en un sistema de dos niveles) y para las de 3 o más. El único factor a tener en cuenta es que aumentar el número de q-bits supondría aumentar o la complejidad del pulso o, de nuevo, el tiempo de aplicación de este. Así, independientemente del número de q-bits con los que trabajemos, siempre tenemos que cuidar el equilibrio entre la complejidad del pulso, el coste computacional que tiene calcularlo, el tiempo máximo de propagación y el tiempo de coherencia del estado cuántico.

Bibliografía

- [1] Michael A.Nielsen e Isaac L.Chuang. *Quantum computation and quantum information*. 2000.
- [2] David Collins, KW Kim y WC Holton. “Deutsch-Jozsa algorithm as a test of quantum computation”. En: *Physical Review A* 58.3 (1998), R1633.
- [3] Constantin Brif, Raj Chakrabarti y Herschel Rabitz. “Control of quantum phenomena: past present and future”. En: *New Journal of Physics* 12.7 (2010), pág. 075008.
- [4] Donald E. Kirk. *Optimal Control Theory. An Introduction*. New York: Dover Publications, Inc., 1998.
- [5] A. Castro. “Optimal Control Theory for Electronic Structure Methods”. En: *Handbook of Materials Modeling* (2018).
- [6] Jelmer Ypma, Hans W.Borchers y Dirk Eddelbuettel. “R interface to NLOpt”. En: *Journal of unreproducible Results* (2018).
- [7] Sergei Kucherenko y Yury Sytsko. “Application of deterministic low-discrepancy sequences in global optimization”. En: *Computational Optimization and Applications*, vol. 30, p. 297-318 (2005).
- [8] J. Nocedal. “Updating quasi-Newton matrices with limited storage”. En: *Math. Comput.* 35, 773-782 (1980).
- [9] P. Kao y M. Ali. “Some variants of the controlled random search algorithm for global optimization”. En: *J.Optim.Theory Appl.* 130(2) 253-264 (2006).
- [10] J.A. Nelde y R. Mead. “A simplex method for function minimization”. En: *The Computer Journal* 7, p.308-313 (1965).
- [11] Thomas P.Runarsson y Xin Yao. “Stochastic ranking for constrained evolutionary optimization”. En: *IEEE Trans. Evolutionary Computation*, vol. 4 (no. 3), pp. 284-294 (2000).
- [12] C.H. da Silva Santos y H.E. Hernandez-Figueroa. “Designing Novel Photonic Devices by Bio-Inspired Computing”. En: *IEEE Photonics Technology Letters* 22 (15), pp. 1177-1179 (2010).
- [13] C. Cohen-Tannoudji, B. Diu y F. Laloë. *Quantum mechanics*. 1973.
- [14] S Zhdanovich, JW Hepburn y V Milner. “Strong-field effects in Rabi oscillations between a single state and a superposition of states”. En: *Physical Review A* 84.5 (2011), pág. 053428.
- [15] MD Jenkins y col. “Coherent manipulation of three-qubit states in a molecular single-ion magnet”. En: *Physical Review B* 95.6 (2017), pág. 064423.
- [16] MD Jenkins y col. “A scalable architecture for quantum computation with molecular nanomagnets”. En: *Dalton Transactions* 45.42 (2016), págs. 16682-16693.

Anexo 1: Centro de supercomputación de Aragón

Un ordenador doméstico requiere demasiado tiempo de computación para realizar los cálculos de optimización. Por ello se ha utilizado el cluster cierto, alojado en Caesaragusta, un superordenador situado en el BIFI (Instituto de Biocomputación y Física de sistemas complejos).

Ceasaraugusta cuenta con 3072 núcleos y 25 TFLOPs y, además, es el nodo en Aragón de la Red Española de Supercomputación (RES), consistente en una interconexión de 12 superordenadores que tiene como objetivo ofrecer recursos de computación de alto rendimiento a la comunidad científica. Se coordina a través del centro de supercomputación de Barcelona.

A su vez, también forma parte, junto con otras dos máquinas (JANUS I y JANUS II) y varios núcleos de computación voluntaria, del centro de supercomputación de Aragón (CESAR).

Generalmente, los cálculos de optimización han tardado entre 1h y 12h. Cada cálculo individual se ha ejecutado de manera serial, aunque dado que necesitábamos múltiples resultados para pulsos iniciales aleatorios, debíamos ejecutar series de cálculos concurrentes en las colas paralelas.

Anexo 2: Justificación experimental

La motivación para la realización de este TFG se basa en el proyecto SUMO (Scaling Up quantum computation with MOlecular spins), liderado por Fernando Luis, del Instituto de Ciencias de Materiales de Aragón. Este proyecto tiene como objetivo el desarrollo de un procesador magneto-cuántico de forma que sea fácilmente escalable. Como sistema cuántico se utilizan los estados de espín del Gd^{+3} , que, al tener una configuración $4f^7$, tiene un momento angular orbital L nulo pero el mayor espín posible ($S = 7/2$). Esto implica 8 posibles estados distintos, es decir, un qudit con $d = 8$ o, de forma equivalente, un sistema de tres q-bits. La ventaja de utilizar los niveles de energía de moléculas individuales se basa en que se minimizan los efectos de interacción magnética dipolo-dipolo, lo cual suele ser una fuente de decoherencia.

La arquitectura del “chip” se basa en una molécula magnética (que incluye el gadolinio) acoplada a un circuito superconductor. En la superficie del procesador existen tres líneas superconductoras coplanares paralelas, donde la central actúa como un resonador superconductor. Sobre esta se sitúan, en lugares específicos, las moléculas mencionadas anteriormente. Además, existe un conjunto de líneas auxiliares superconductoras perpendiculares a las iniciales. Este chip se fabrica mediante depósito de películas delgadas (150-300 nm) de material superconductor, generalmente Nb o Al , sobre un sustrato y finalmente, mediante litografía óptica, se genera el circuito.

El gap de energía entre los estados lógicos puede ser modificado mediante un campo magnético \vec{B} , generado por la línea superconductora superior (la inferior actúa como tierra), y por los campos locales \vec{b}_i generados por las líneas auxiliares perpendiculares. Para que el sistema se acople a las componentes \vec{b}_i del campo, las moléculas se sitúan en el centro del resonador pues, en el modo fundamental, es donde se sitúa el máximo (siendo que los nodos se encuentran en los bordes del resonador). Estas componentes también pueden inducir las transiciones entre los estados pues los campos, además de tener una componente DC, pueden tener una componente oscilante, con frecuencias típicas entre 1 y 10 GHz. Dado que la frecuencia característica del resonador es de unos pocos GHz es fácil ajustar el diseño al rango de las transiciones en los niveles moleculares.

Todo el sistema debe estar a baja temperatura, no solo para que aparezcan las propiedades superconductoras, sino también para inicializar el sistema a su estado fundamental.

Para más información véase [15] y [16].