

2023年度 修士論文

超伝導量子コンピュータにおける
量子トリットを用いた多制御ビットゲートの開発
(Development of multi-controlled gates using
superconducting qutrits for quantum computers)

東京大学大学院
理学系研究科 物理学専攻
寺師研究室

尾嶋 陸斗
令和6年1月

概要

量子コンピュータは、従来のコンピュータではリソースや計算速度の観点で解くことが困難な問題に対し、有効であると期待されている。さらに近年はソフトウェアとハードウェアの開発が盛んに行われており、量子コンピュータは現実のものとなりつつある。しかし現在の量子コンピュータは計算に誤りが発生しやすく、量子ビット数や量子回路の深さ等に制約がある。これは NISQ (Noisy Intermediate-Scale Quantum) デバイスと呼ばれており、演算精度を上げるために、ノイズ特性を理解して抑制することが重要となる。

量子コンピュータにおける多制御ビットゲートは量子アルゴリズムに不可欠なゲートだが、制御ビット数が大きいほど、実装には多くの CNOT が必要であり、エラー補正がない場合、長時間演算によって精度低下の影響を受けやすい。また超伝導量子コンピュータである IBMQuantum では単一量子ビットゲートと比べて CNOT のエラー率は数十倍高く、主なエラー源となっているため、多制御ビットゲートの精度向上のためには、より少ない CNOT 数で実装することが重要である。

量子トリットとは、量子ビットの $|2\rangle$ を使った 3 準位系で演算を行う技術であり、使用する量子ビット数や量子回路の深さを抑えられるため、少ない CNOT 数で多制御ビットゲートを実装できる。しかし量子ビットと比べてノイズが増加するため、この技術を活かすためにはどれだけ量子トリットのノイズが抑制できるかが重要な課題となる。

本研究では、量子トリットを用いて、より少ない CNOT 数で多制御ビットゲートを実装するマイクロ波パルスシーケンスの設計を試みた。その際、量子トリットにおいて大きな位相エラーを起こすべり一位相、ZZ 相互作用、電荷分散という効果を抑制するような技術を考案し、量子トリットを用いた多制御ビットゲートのパルスシーケンスに具体的に組み込む方法を提案した。またこの提案した設計について、シミュレータと実機を用いて実証を行ない、各効果への対策設計が有効であることを確認した。

目 次

第 1 章 序論	3
1.1 背景	3
1.1.1 量子コンピュータ	3
1.1.2 多制御ビットゲート	3
1.2 目的	4
1.3 構成	4
第 2 章 量子計算	5
2.1 量子回路	5
2.1.1 量子情報単位	5
2.1.2 量子ゲート	5
2.1.3 測定	9
2.2 超伝導量子コンピュータ	11
2.2.1 トランズモン	11
2.2.2 ハミルトニアン	12
2.2.3 パルスによる量子ゲートの実装	13
2.2.4 Virtual Z Gate	14
2.2.5 CNOT	15
第 3 章 量子トリット	20
3.1 量子トリットの利点	20
3.2 量子トリット使用上の問題点	21
3.2.1 ベリー位相	21
3.2.2 ZZ 相互作用	21
3.2.3 電荷分散	24
第 4 章 量子トリット n 制御ビットゲートの設計	27
4.1 n 制御ビットゲート分解方法	27
4.1.1 使用するゲートの行列表現	28
4.1.2 ベリー位相への対策	30
4.2 ZZ 相互作用への設計	34
4.2.1 ZZ 相互作用を解消する原理	34
4.2.2 ゲートシーケンスへの組み込み方	36

4.3	電荷分散への設計	37
4.3.1	電荷分散を解消する手法	37
4.3.2	ゲートシーケンスへの組み込み方	39
第5章	シミュレーションと実機での実証	42
5.1	qudit-sim での評価	42
5.1.1	qudit-sim の説明	42
5.1.2	評価手法	43
5.1.3	実験	44
5.2	IBMQQuantum での評価	50
5.2.1	実験手法	50
5.2.2	実験	51
5.2.3	最終的な忠実度比較と考察	61
5.2.4	local Rz の原因考察	62
第6章	結論	68
6.1	まとめ	68
6.2	今後の展望	69
第7章	謝辞	70
付録A	付録	76
A.1	Qutrit echo cross-resonance CNOT の $ZX(\pi/3)\zeta X(-\pi/3)$ の計算	76
A.2	量子トリット-量子トリット間の ZZ 相互作用の解消計算	77
A.3	補正用位相の算出	78
A.4	dressed frame	80

第1章 序論

1.1 背景

1.1.1 量子コンピュータ

量子計算 [1] とは、重ね合わせ状態や量子もつれといった量子力学の現象を用いることで、従来の古典計算機では計算時間やリソースの観点で解くことが困難な問題に対して、有効であると期待されている計算手法である。実際にショアの素因数分解アルゴリズム [2] やグローバーの探索アルゴリズム [3] は、対象とする問題に対し古典アルゴリズムよりも少ない計算量で解けることが示されている。この量子計算を実現するため、近年は量子コンピュータのハードウェア開発が多く行われている。2011年に D-Wave が商用型の量子コンピュータを発表した [2][4] 他、2019年に Google が実機を用いて量子超越性を主張する発表 [5] を行った。IBM が開発した超伝導量子コンピュータ（以降 IBMQuantum の記す）はクラウド上で利用可能であり、現在もアップデートが多く行われている。

量子コンピュータは様々な物理系で開発がなされており、代表的なものとして超伝導回路式 [6] や、光式 [7]、イオントラップ式 [8] 等がある。しかし現在開発されているこれらの量子コンピュータは、Noisy Intermediate-Scale Quantum(以降 NISQ と記す)[9] と呼ばれており、量子制御が不完全であり外部からのノイズやエラーによって誤りが発生しやすい、数十から数百ビットと比較的中規模である、誤り訂正機能を持たない等の共通点を持つ。NISQ は上記の制約を有しているため、現在もアプリケーションとしての量子アルゴリズムやデバイスの精度向上の研究、ミドルウェア開発等が行われている。有名なミドルウェア開発として、CNOT エラー緩和 [10] や測定エラー緩和 [11]、トランスペイイルの最適化 [12] がある。

一般的に超伝導量子コンピュータでは、

1. ソフトウェアで実装したい量子回路を作成する
2. 量子回路を特定の実機で物理的に実装可能な形に書き換えるトランスペイイルを行う
3. 超伝導量子ビットとマイクロ波パルスにより、ハードウェアでゲートを実装する

といった流れで量子演算が行われる。2番目の工程のトランスペイイルでは、ソフトウェアで作成した量子回路を物理的に実装可能な基本ゲートに分解する必要があったり、物理的に離れている量子ビット間の操作を行うためにゲートを追加したりと手間がかかるため、最適化の切り口が多数存在する。

1.1.2 多制御ビットゲート

多制御ビットゲートは上で述べた重要なアルゴリズム [2][3] や特定の量子誤り訂正符号 [13] で使用するため、量子アルゴリズムに不可欠なゲートである。多制御ビットゲートはハードウェアで実装する際、作

業ビットや SWAP ゲートが必要になることに加え、制御ビットの数に比例して CNOT が多数必要になる [14][15]。特に IBMQuantum では、単一量子ビットゲートのエラー率が 10^{-4} であるのに対し、CNOT のエラー率は 10^{-2} であるため、CNOT は主な精度低下要因であり [16]、使用数を減らしたいというモチベーションがある。この課題へのアプローチとして、通常の量子ビットの状態 $|0\rangle|1\rangle$ に加えて状態 $|2\rangle$ を使用した量子トリットという技術を用いて、新しい基本ゲートへ分解することで、トランスペイル後の CNOT 数を大幅に削減する手法がある [17][18][19][13] [20][21][22]。

超伝導量子ビットであるトランズモンは、基底状態を $|0\rangle$ 、第一励起状態を $|1\rangle$ としているため、トランズモンの第二励起状態 $|2\rangle$ を使用することで量子トリットとして扱える。IBMQuantum では、最近 IBM が開発した qiskit.pulse[23] というモジュールにより、ユーザー側がマイクロ波パルスを制御して、第二励起状態を使用したり、独自に量子ゲートを作成したりすることが可能になっている [24][22][25][26]。実際にこの qiskit.pulse によって量子トリットによる基本ゲートを独自で作成し、制御ビットが 2 個の多制御ビットゲートである Toffoli を少ない CNOT 数で実装することで、精度が向上した先行研究もある [27]。この先行研究では、制御ビットが 3 個以上の多制御ビットゲートのパルスレベルでの実装手法については言及されていないため、任意の n 個の制御ビットを持つ量子トリットを使用した多制御ビットゲート（以降 n 制御ビットゲートと記す）をパルスで実装する方法については、新たに設計する必要がある。

1.2 目的

本研究では n 制御ビットゲートをより少ない CNOT 数でハードウェアに実装することを目的に、量子トリットを使用したパルスレベルの多制御ビットゲート実装手法の開発を行う。

トランスペイル後の CNOT 数が大幅に削減できる利点を生かすためには、量子トリットの基本ゲートを精度よく実装することが不可欠である。そのためには、以下の 2 点を満たす必要がある：

1. 量子トリットの基本ゲートを実現するマイクロ波パルスのシーケンスを確立する
2. 量子トリットで $|2\rangle$ を使用することで新しく生じるエラーへの対策をする

前者に関しては、Toffoli の先行研究 [27] に基本ゲートのパルスシーケンスが記されているため、 n 制御ビットゲートにも適用できる。後者に関しては、先行研究では $n = 2$ の Toffoli にのみ適用可能な形となっているため、 n 制御ビットゲートに拡張可能なエラー対策手法を新たに開発する必要がある。

したがって対策手法を開発し、その効果をシミュレータと IBMQuantum で検証することで、今後の発展の方向性を議論することを本研究の目的とする。

1.3 構成

2 章で量子計算における量子回路や超伝導量子コンピュータの基礎的な知識を説明し、3 章で多制御ビットゲートを量子トリットで実装することの利点と、ハードウェアで使用する際の問題を紹介する。その後、4 章で 3 章に挙げた問題を解消する手法の説明をし、5 章ではそれらの手法に対してシミュレータと実機を用いて実証した結果を述べる。最後に 6 章で本研究の結論と今後の展望を述べる。

第2章 量子計算

2.1 量子回路

2.1.1 量子情報単位

量子情報単位は、用いられる計算基底数によって呼び方が異なる。量子ビット (Qubit) は $|0\rangle$ と $|1\rangle$ の 2 つ、量子トリット (Qutrit) はこれらに $|2\rangle$ を加えた 3 つの計算基底を用いる。なお d 個の計算基底を用いる場合は 量子ディット (Qudit) と呼ぶ。実際には扱いやすさと、[2.2.1](#) 節で説明するようなエラーの存在から、量子ビットが使われる事が殆どである。量子ビットは、二つの正規直交基底を計算基底として扱い、これらの複素数の係数を用いた線型結合もしくは重ね合わせによる 2 次元複素ベクトル空間で表現される。

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle \quad (2.1.1)$$

ただし $|\alpha|^2 + |\beta|^2 = 1$ であり、これを正規化条件という。つまり量子ビットにおける量子状態は、図 2.1 にあるように 2 次元複素ベクトル空間における半径 1 のブロッホ球上の点で表される。なおブロッホ球の点を極座標で

$$|\psi\rangle = e^{i\gamma} \left(\cos \frac{\theta}{2} |0\rangle + e^{i\phi} \sin \frac{\theta}{2} |1\rangle \right) \quad (2.1.2)$$

というように表す事もできる。 $e^{i\gamma}$ は絶対位相と呼び、 γ の値によって量子状態は区別されないため、量子計算においては全く意味のないパラメータである。そのため絶対位相は省略される。 $e^{i\phi}$ は相対位相と呼び、 ϕ の値に応じて測定基底によっては異なった測定が得られたりするため、量子計算において意味のあるパラメータである。同様に量子トリットでは、

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle + \gamma|2\rangle \quad (2.1.3)$$

ただし $|\alpha|^2 + |\beta|^2 + |\gamma|^2 = 1$ である。極座標で表すと

$$|\psi\rangle = \sin \frac{\eta}{2} \cos \frac{\theta}{2} |0\rangle + e^{i\phi_1} \sin \frac{\eta}{2} \sin \frac{\theta}{2} |1\rangle + e^{i\phi_2} \cos \frac{\eta}{2} |2\rangle \quad (2.1.4)$$

となり、相対位相が 2 つになっている [28]。

2.1.2 量子ゲート

量子ゲートは、量子コンピュータで計算や情報処理を行うための基本的な操作の一つである。古典的なコンピュータにおける論理ゲートに相当するもので、量子ビットの状態を変更するために使用される。

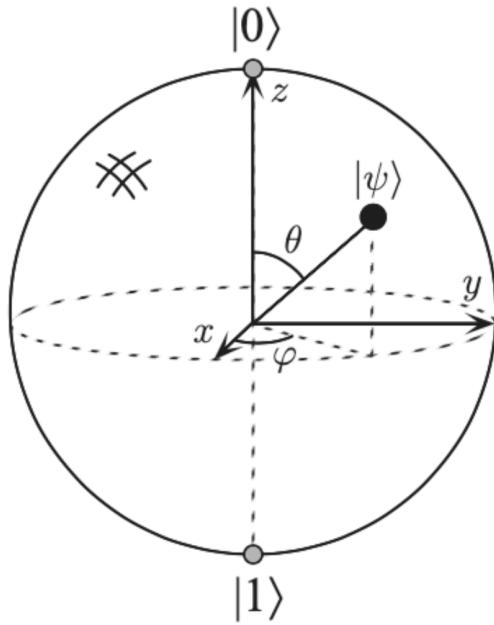


図 2.1.1: 量子ビットのブロッホ球による表現

量子ゲートはユニタリ変換であり、一般的に行列表現が便利であるため、まずは量子ビットゲートの行列表示を行う。量子ビットの各状態を

$$|0\rangle = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, |1\rangle = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \quad (2.1.5)$$

とベクトルで表現する。ここで量子ビットゲートの生成子はパウリ行列であり、

$$\begin{aligned} I &= |0\rangle\langle 0| + |1\rangle\langle 1| = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \\ \sigma_x &= |0\rangle\langle 1| + |1\rangle\langle 0| = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \\ \sigma_y &= -i|0\rangle\langle 1| + i|1\rangle\langle 0| = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix} \\ \sigma_z &= |0\rangle\langle 0| - |1\rangle\langle 1| = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \end{aligned} \quad (2.1.6)$$

である。図 2.1.1 の表されるブロッホ球上の x、y、z 軸周りの回転ゲートをそれぞれ R_x 、 R_y 、 R_z ゲートと呼び、パウリ行列を用いて

$$\begin{aligned}
Rx(\theta) &= e^{-i\sigma_x \frac{\theta}{2}} = \cos \frac{\theta}{2} I - i \sin \frac{\theta}{2} \sigma_x = \begin{bmatrix} \cos \frac{\theta}{2} & -i \sin \frac{\theta}{2} \\ -i \sin \frac{\theta}{2} & \cos \frac{\theta}{2} \end{bmatrix} \\
Ry(\theta) &= e^{-i\sigma_y \frac{\theta}{2}} = \cos \frac{\theta}{2} I - i \sin \frac{\theta}{2} \sigma_y = \begin{bmatrix} \cos \frac{\theta}{2} & -\sin \frac{\theta}{2} \\ \sin \frac{\theta}{2} & \cos \frac{\theta}{2} \end{bmatrix} \\
Rz(\theta) &= e^{-i\sigma_z \frac{\theta}{2}} = \cos \frac{\theta}{2} I - i \sin \frac{\theta}{2} \sigma_z = \begin{bmatrix} e^{-i\frac{\theta}{2}} & 0 \\ 0 & e^{i\frac{\theta}{2}} \end{bmatrix} \sim \begin{bmatrix} 1 & 0 \\ 0 & e^{i\theta} \end{bmatrix}
\end{aligned} \tag{2.1.7}$$

と定義される。 Rz ゲートの最後の式変形では絶対位相を除いた。以降、絶対位相を除くときは～を使用する。例えば $\theta = \pi$ である時に R_x ゲートは

$$Rx(\pi) = e^{-i\sigma_x \frac{\pi}{2}} = -i\sigma_x = \begin{bmatrix} 0 & -i \\ -i & 0 \end{bmatrix} \sim \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \tag{2.1.8}$$

であり、 X ゲートとなる。任意の一量子ビットゲートは、 Rx 、 Ry 、 Rz のうち 2 種類を選び、それらの組み合わせで表現することが可能である [1]。

また代表的な二量子ビットゲートである CNOT (CX) は、

$$CX = |0\rangle\langle 0| \otimes I + |1\rangle\langle 1| \otimes X = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \tag{2.1.9}$$

と表される。CNOT は制御ビットが $|0\rangle$ のときは標的ビットに作用せず、制御ビットが $|1\rangle$ のときは標的ビットに X を作用させるゲートである。任意の二量子ビットゲートは 2 つの CNOT と、複数の単一量子ビットゲートで表現することができる [1]。また CNOT は制御ビットが 1 つであったが、2 つ以上の制御ビットをもつ制御ゲートを n 制御ビットゲートと呼ぶ。 n 個の制御ビットがある n 制御ビットゲート ($C^n X$) は

$$C^n X = (I^{\otimes n} - |1\rangle^{\otimes n} \langle 1|^{\otimes n}) \otimes I + |1\rangle^{\otimes n} \langle 1|^{\otimes n} \otimes X \tag{2.1.10}$$

と表せる。この n 制御ビットゲートは、 $n = 2$ の Toffoli ゲートを多項式個数と、二量子ビットゲートのみで表現することが可能であることが知られている [29][30][31]。なお n 制御ビットゲートの分解方法は様々あり、分解の仕方によって補助として必要な作業ビット数や SWAP ゲート数といった実装コストが異なる。

次に量子トリットゲートについて紹介する。こちらもユニタリ変換であるため、行列表示で説明する。まず量子トリットの各状態を

$$|0\rangle = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, |1\rangle = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}, |2\rangle = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}, \tag{2.1.11}$$

とベクトルで表現する。量子トリットゲートの生成子はゲルマン行列が用いられ、その一部を紹介すると、

$$I = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \lambda_1 = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \lambda_3 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad (2.1.12)$$

$$\lambda_6 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}, \lambda_8 = \frac{1}{\sqrt{3}} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & -2 \end{bmatrix}$$

である。これらにより量子トリットゲートは生成され、例えば量子ビットゲートとして紹介した Rx に関する量子トリット表記では、

$$Rx_{01}(\theta) = e^{-i\lambda_1 \frac{\theta}{2}} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} + \cos \frac{\theta}{2} I - i \sin \frac{\theta}{2} \lambda_1 = \begin{bmatrix} \cos \frac{\theta}{2} & -i \sin \frac{\theta}{2} & 0 \\ -i \sin \frac{\theta}{2} & \cos \frac{\theta}{2} & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (2.1.13)$$

となる。 $|1\rangle - |2\rangle$ 間の x 軸回転ゲートである Rx_{12} も同様に

$$Rx_{12}(\theta) = e^{-i\lambda_6 \frac{\theta}{2}} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} + \cos \frac{\theta}{2} I - i \sin \frac{\theta}{2} \lambda_6 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \frac{\theta}{2} & -i \sin \frac{\theta}{2} \\ 0 & -i \sin \frac{\theta}{2} & \cos \frac{\theta}{2} \end{bmatrix} \quad (2.1.14)$$

と与えられる。例えば $\theta = \pi$ であるとき、 Rx_{01} と Rx_{12} は次のようになる：

$$Rx_{01}(\pi) = e^{-i\lambda_1 \frac{\pi}{2}} = \begin{bmatrix} 0 & -i & 0 \\ -i & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \equiv X_{01} \quad (2.1.15)$$

$$Rx_{12}(\pi) = e^{-i\lambda_6 \frac{\pi}{2}} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & -i \\ 0 & -i & 0 \end{bmatrix} \equiv X_{12} \quad (2.1.16)$$

量子トリットゲートは $|0\rangle$ 、 $|1\rangle$ 、 $|2\rangle$ の全ての計算基底状態に作用させる必要はなく、このうち 2 つの基底間に作用するゲートのみで任意の単一量子トリットゲートを構成することができる [32]。以降、どの二つの基底に作用する量子トリットゲートかわかりやすくするために、量子トリットゲートは U_{kl} と表記する。 U はユニタリ行列で k, l は作用する計算基底を指し、 $k, l = 0, 1, 2 (k \neq l)$ である。例えば $|0\rangle$ 、 $|1\rangle$ 、 $|2\rangle$ のうち 2 つの基底間にのみ作用する任意の単一量子トリットゲートは、 X_{12} と単一量子ビットゲートで実現可能である。実際に $|0\rangle - |1\rangle$ 間の任意の単一量子ビットゲート U_{01} を $|0\rangle - |2\rangle$ 間でしたい場合は、次のように X_{12} で挟めば実装可能である：

$$X_{12} U_{01} X_{12}^\dagger = U_{02} \quad (2.1.17)$$

また $|1\rangle - |2\rangle$ 間でしたい場合は

$$X_+ \equiv X_{01} X_{12} \quad (2.1.18)$$

で定義される X_+ を用いて次のように実装可能である:

$$X_+ U_{01} X_+^\dagger = U_{12} \quad (2.1.19)$$

任意の二量子トリットゲートは、二量子トリットゲートとしての CNOT と单一量子トリットゲートから作成可能であり [33]、任意の单一量子トリットゲート U_{kl} は、 U_{01} と X_{01} 、 X_{12} により作成できることは上で見た。したがって二量子トリットゲートとしての CNOT を構成できれば、任意の二量子トリットゲートが作成できる。二量子トリットゲートとしての CNOT の構成方法は、この後の 2.2.5 節で紹介する。

2.1.3 測定

量子状態トモグラフィ

量子状態トモグラフィとは、未知の量子状態がどのような状態であるかを同定する手法である [1][34]。この後に出でてくる量子プロセストモグラフィで使用するため、先にこちらを紹介する。まず一量子ビット系の量子状態 ρ を次のように表す:

$$\rho = \frac{1}{2} (I + r_x \sigma_x + r_y \sigma_y + r_z \sigma_z) \quad (2.1.20)$$

このとき係数 r_x 、 r_y 、 r_z が特定できれば、量子状態が同定できる。ここでパウリ演算子 σ_i は $\sigma_i \sigma_j (i \neq j)$ のトレースが 0 であることを利用すれば、 $r_i = \text{Tr}[\rho \sigma_i]$ で求まる。

量子実験において、ほとんどの測定は量子状態の Z 基底に対する占有率、すなわち r_z を特定する Z 基底での測定である。この場合、 r_x や r_y は測定によって直接特定できないが、Z 基底で測定する直前に $Rx(\pi/2)$ や $Ry(\pi/2)$ を作用させると、式 (2.1.20) は次のようになる:

$$Rx(\pi/2) \rho Rx(\pi/2)^\dagger = \frac{1}{2} (I + r_x \sigma_x + r_y \sigma_z + r_z \sigma_y) \quad (2.1.21)$$

$$Ry(\pi/2) \rho Ry(\pi/2)^\dagger = \frac{1}{2} (I + r_x \sigma_z + r_y \sigma_y + r_z \sigma_x) \quad (2.1.22)$$

したがって r_x と r_y も測定により特定することができる。

このように量子実験により多数回測定し、 (r_x, r_y, r_z) を特定して量子状態を同定することを量子状態トモグラフィと呼ぶ。

量子プロセストモグラフィ

量子プロセストモグラフィ (QPT) とは、量子過程を完全に同定するための手法であり [1][34]、実験的に実装された量子ゲートが理想的な量子ゲートにどれだけ近いかを定量的に評価できる。本研究では、最終的に量子過程の評価をするためにこの技術を用いているため、ここで紹介する。

ある量子状態 ρ に対し、未知の量子ゲートを作用させる事を考える。量子ゲートによる作用は次のように表現できる:

$$\mathcal{E}(\rho) = \sum_i A_i \rho A_i^\dagger \quad (2.1.23)$$

n 量子ビット系の場合、量子状態は $d = 2^n$ 次元のヒルベルト空間になるが、これに対する Kraus 演算子 A は、直交する d 個の演算子を用いて次のように表現できる:

$$A_i = \sum_m a_{im} E_m, m = 1, 2, \dots, d \quad (2.1.24)$$

この A_i を式 (2.1.23) に代入すると、次のように書き直せる:

$$\mathcal{E}(\rho) = \sum_{m,n} \chi_{mn} E_m \rho E_n^\dagger \quad (2.1.25)$$

ここで $\chi_{mn} = \sum_i a_{im} a_{in}^*$ である。

また実験において、 d^2 次元の量子ビット系のうち、線形独立な状態 $\{\rho_j\}$ のうちの一つ ρ_j に量子ゲートを作用させ、量子状態トモグラフィによって作用後の量子状態を同定する。この結果は次のように表現できる:

$$\mathcal{E}(\rho_j) = \sum_k c_{jk} \rho_k \quad (2.1.26)$$

ここで $\{\rho_j\}$ は線形独立なので、式 (2.1.25) の右辺は

$$E_m \rho_j E_n^\dagger = \sum_k \beta_{mnjk} \rho_k \quad (2.1.27)$$

という展開も可能であり、 $\mathcal{E}(\rho_j)$ に対し、式 (2.1.25) と (2.1.26) は等しいため、係数比較により次の結果が成立する:

$$\sum_k c_{jk} \rho_k = \sum_{m,n} \sum_k \chi_{mn} \beta_{mnjk} \rho_k \Leftrightarrow c_{jk} = \sum_{m,n} \chi_{mn} \beta_{mnjk} \quad (2.1.28)$$

$$\Leftrightarrow \chi_{mn} = \sum_{j,k} \beta_{mnjk}^{-1} c_{jk}. \quad (2.1.29)$$

したがって χ_{mn} は、 β_{mnjk} と c_{jk} が求まれば明らかになる。 β_{mnjk} は式 (2.1.27) から計算によって算出でき、 c_{jk} は式 (2.1.26) から量子状態トモグラフィ実験によって算出できる。つまり $\{\rho_j\}$ に対し、量子状態トモグラフィ実験を行えば量子過程を同定できる。

欠点は、量子ビット数が増えると量子状態トモグラフィ実験数と扱う行列サイズが指数関数的に増大するため、大規模な系への実装は現実的ではないことである。例えば 4 量子ビット系の場合、量子状態は $d = 2^4$ 次元ヒルベルト空間であり、 ρ_j は $d^2 = 256$ 個ある。この 256 個の ρ_j を量子状態トモグラフィで特定するため、各量子ビットに X, Y, Z 基底で測定するとなると、4 量子ビット系では $3^4 = 81$ 通りの測定が必要となる。つまり 4 量子ビット系の量子ゲートを同定するためには、一度の量子プロセストモグラフィに $256 \times 81 = 20736$ 個の量子回路の実装が必要になる。エラーの見積もり等で複数回量子プロセストモグラフィを実行する場合、その回数倍だけ量子回路の実装が必要になってしまう。

忠実度

量子状態トモグラフィで量子状態が同定できたとき、理想的な量子状態に対してどの程度近いかを表す指標となるのがこの忠実度である [1][34]。一般に量子状態 ρ と σ に対し、忠実度 F は次のように定義される:

$$F = \text{Tr}[\sqrt{\sqrt{\rho} \sigma \sqrt{\rho}}] \quad (2.1.30)$$

2つの状態が直交しているときは0、一致しているときは1となり、直感的にわかりやすい指標となっている。

2.2 超伝導量子コンピュータ

この章では、本研究で扱う超伝導量子コンピュータに関する基本的な知識を説明する。[2.2.1](#) 節では超伝導量子ビットであるトランズモンについて、[2.2.2](#) 節以降では超伝導量子コンピュータにおけるマイクロ波パルスによるハミルトニアンと量子ゲート実装手法について、[6][35][34] を参考に説明する。またこれ以降の本文では $\hbar \equiv 1$ としてハミルトニアンを記述する。

2.2.1 トランズモン

まずインダクタとキャパシタのみから構成される LC 回路について考える。 L をインダクタンス、 C をキャパシタンス、 Q を電荷、 Φ を流束とする。LC 回路の古典力学によるハミルトニアンを量子化すると、

$$\hat{H}_{\text{LC}} = \frac{\hat{Q}^2}{2C} + \frac{\hat{\Phi}^2}{2L} \quad (2.2.1)$$

となる。ここで e を電荷素量、 h をプランク定数とし、 $\Phi_0 = \frac{h}{2e}$ を用いて、換算電荷 $\hat{n} = \frac{\hat{Q}}{2e}$ 、換算流束 $\hat{\phi} = \frac{2\pi\hat{\Phi}}{\Phi_0}$ を定義し、

$$\hat{H}_{\text{LC}} = 4E_c\hat{n}^2 + \frac{1}{2}E_L\hat{\phi}^2 \quad (2.2.2)$$

と表せる。これは一次元量子調和振動子のハミルトニアンの形であるため、生成消滅演算子 \hat{a}, \hat{a}^\dagger と交換関係 $[\hat{\phi}, \hat{n}] = i$ を用いて

$$\hat{H}_{\text{LC}} = \hbar\omega \left(\hat{a}^\dagger \hat{a} + \frac{1}{2} \right) \quad (2.2.3)$$

と表せる。ここで $\omega = \frac{1}{\sqrt{LC}}$ である。これにより、エネルギー準位 n は

$$E_n = \hbar\omega \left(n + \frac{1}{2} \right) \quad (2.2.4)$$

となる。このエネルギー準位では、任意の n に対して $E_{n+1} - E_n = \hbar\omega$ と等間隔になるため、ある $E_n \leftrightarrow E_{n+1}$ のエネルギー遷移を選んで励起しようとしても、全てのエネルギー準位に作用してしまう。

これを防ぎ、狙ったエネルギー遷移の励起を可能にするため、トランズモンではインダクタの代わりにジョセフソン接合を導入し、ハミルトニアンに非線形効果を入れる。ジョセフソン接合とキャパシタによるハミルトニアンは、ジョセフソンエネルギー $E_J = \frac{I_0\Phi_0}{2\pi}$ を定義すると

$$\hat{H}_{\text{tr}} = 4E_c\hat{n}^2 - E_J \cos \hat{\phi} \quad (2.2.5)$$

と表せる。ここで $\cos \hat{\phi}$ をテーラー展開して、6次以上の項を無視し、生成消滅演算子で置き換えると、

$$\hat{H}_{\text{tr}} = \hbar \left(\omega - \frac{\delta}{2} \right) \hat{b}^\dagger \hat{b} + \hbar \frac{\delta}{2} \hat{b}^\dagger \hat{b} \hat{b}^\dagger \hat{b} + \frac{1}{2} \hbar\omega \quad (2.2.6)$$

となる。ここで、 $\delta = -E_c < 0$ である。よって、トランズモンのエネルギー準位は、

$$E_n = \hbar \left(\omega - \frac{\delta}{2} \right) n + \hbar \frac{\delta}{2} n^2 + \frac{1}{2} \hbar \omega \quad (2.2.7)$$

となる。このとき、

$$E_{n+1} - E_n = \hbar(\omega + n\delta) \quad (2.2.8)$$

となっており、ジョセフソン接合の非線形効果によりエネルギー準位の間隔が n に依存しているため、トランズモンでは特定の $E_n \leftrightarrow E_{n+1}$ のエネルギー遷移を選択して励起させることができる。この E_n はトランズモンの $|n\rangle$ に対応しており、 $E_n \leftrightarrow E_{n+1}$ のエネルギー遷移は、 $|k\rangle \leftrightarrow |k+1\rangle$ の状態操作に対応する。この δ は非調和性と呼ばれており、これによりトランズモンでは特定の準位間の遷移を選択的に励起できるようになるため、重要な役割を担っている。

トランズモンは上記のように構成されていることから、量子ディット (Qudit) という技術が存在する。これは通常の量子ビット (qubit) として状態 $|0\rangle$ 、 $|1\rangle$ を使用するのではなく、トランズモンの $|0\rangle$ から $|d-1\rangle$ までの d 個の状態を使用して量子計算を行うというものである。 n 個のトランズモン系では基底数が d^n と量子ビットよりも大きくなり、より複雑な情報をエンコードし、計算効率を向上させることができるために、少ないゲート数で等価な量子計算が可能となる。しかし高準位ほど自然放出や電荷分散 (charge dispersion) による脱位相エラーが大きくなることに加えて、考慮すべきエネルギー準位が多くなるため、遷移周波数が近づくことによって不必要的エネルギー遷移の励起が生じやすい [36]。このことから高準位を用いることは、ゲート数の減少という利点とエラー増幅という欠点のトレードオフがあり、実験的には制限なく高準位を用いることは現実的ではない [36]。

本研究では、量子トリット ($d = 3$) を用いて実験を行っている。量子トリットを選択した理由については 3 章に記す。

2.2.2 ハミルトニアン

任意のトランズモン間で結合がある n Qudit 系で、パルスをドライブした場合のハミルトニアン H は、Qudit ハミルトニアン H_0 、相互作用ハミルトニアン H_{int} 、ドライブハミルトニアン H_d により、次のように与えられる [35]:

$$H_{\text{stat}} \equiv H_0 + H_{\text{int}} + H_d \quad (2.2.9)$$

このハミルトニアンを $H_0 + H_{\text{int}}$ と H_d に分けて説明する。

静的ハミルトニアン

式 (2.2.9)において、パルスをドライブしない場合のハミルトニアンは $H_0 + H_{\text{int}} (\equiv H_{\text{stat}})$ であり、これを静的ハミルトニアンと定義する。 H_0 と H_{int} はそれぞれ次のように与えられる [35]:

$$\begin{aligned} H_0 &\equiv \sum_{j=1}^n \left[\omega_j b_j^\dagger b_j + \frac{\delta_j}{2} b_j^\dagger b_j \left(b_j^\dagger b_j - 1 \right) \right] = \sum_{j=1}^n \left[\left(\omega_j - \frac{\delta_j}{2} \right) n_j + \frac{\delta_j}{2} n_j^2 \right] \\ H_{\text{int}} &\equiv \sum_{j < k} J_{jk} \left(b_j^\dagger b_k + b_j b_k^\dagger \right) \end{aligned} \quad (2.2.10)$$

ここで $b_k = \sum \sqrt{n+1} |n\rangle \langle n+1|$ である。また、添字 j は j 番目の Qudit であることを指し、 ω_j は Qudit の量子ビット周波数、 δ_j は非調和性、 J_{jk} は結合定数 (coupling constant) と呼ばれ、 j 番目と k 番目の Qudit の結合強度を表す量である。本研究で使用する IBMQuantum のパラメータの典型的な大きさは、 $\omega_j \sim 5[GHz]$ 、 $\delta_j \sim -0.35[GHz]$ 、 $J_{jk} \sim 2[MHz]$ である。

n Qudit 系はパルスの有無に関わらず、常に H_{stat} によって時間発展が起こるため、長時間の量子演算であるほど、この H_{stat} をアイデンティティ I に近づけることが重要である。

ドライブハミルトニアン

パルスをドライブした際のハミルトニアンは、次のように与えられる [35]:

$$H_d = \sum_{jk} \alpha_{jk} \frac{\Omega_j}{2} \left(r_j(t) e^{-i(\nu_j t - \rho_{jk})} + \text{c. c.} \right) (b_k^\dagger + b_k) \quad (2.2.11)$$

ここで Ω_j はパルスの振幅、 ν_j はパルスのドライブ周波数、 $r_j(t)$ はパルスの形を表すパラメータ、 α_{jk} と ρ_{jk} はそれぞれ結合で生じた信号干渉による減衰と位相シフトである。本研究でしばしばパルス振幅を $0 < \alpha < 1$ で表記するが、これは $\sim 0.8[GHz]$ 程度で規格化した値であり、単位は [a.u.] である。

2.2.3 パルスによる量子ゲートの実装

回転系と有効ハミルトニアン

ドライブハミルトニアンの役割を明らかにするため、式 (2.2.9) のハミルトニアンを回転系に変換する。 $U_q(t) \equiv e^{iH_0 t}$ を使用して、変換は次のようにされる [6][35]:

$$\begin{aligned} \tilde{H} &\equiv U_q H U_q^\dagger + i \dot{U}_q U_q^\dagger \\ &= U_q (H_{\text{int}} + H_d) U_q^\dagger = \tilde{H}_{\text{int}} + \tilde{H}_d \end{aligned} \quad (2.2.12)$$

以降の本文ではハミルトニアンの上のチルダは、回転系での有効ハミルトニアンであることを指す。この有効ハミルトニアンによる時間発展が、量子ゲートの物理的な正体である。

量子ゲートの実装

簡易化のため、単体の量子ビットがパルスでドライブされている状況を考えると、そのハミルトニアンは式 (2.2.9) から次のように与えられる：

$$H = \frac{\omega_0}{2} \sigma_z + \frac{\Omega_0}{2} \left(b_0^\dagger e^{-i\nu_0 t} + b_0 e^{i\nu_0 t} + b_0^\dagger e^{i\nu_0 t} + b_0 e^{-i\nu_0 t} \right) \quad (2.2.13)$$

$$\simeq \frac{\omega_0}{2} \sigma_z + \frac{\Omega_0}{2} \left(b_0^\dagger e^{-i\nu_0 t} + b_0 e^{i\nu_0 t} \right) \quad (2.2.14)$$

ドライブ周波数 ν_0 は量子ビット周波数 ω_0 に近い値であると仮定して、最後の式変形は $\omega_0 + \nu_0 \gg \omega_0 - \nu_0$ による回転波近似 [34] を利用した。

ここでユニタリ行列 $U \equiv \exp [i(\nu_0/2)\sigma_z t]$ によって回転系への変換を考える。 ν_0 を ω_0 に近い値にして、再度回転波近似 [34] を用いると、式 (2.2.13) の有効ハミルトニアンは式 (2.2.12) に則り、パルスの位相 ϕ を用いて、次のようになる：

$$\tilde{H} = -\frac{(\omega_0 - \nu_0)}{2}\sigma_z + \Omega_0 (\cos \phi \sigma_x + \sin \phi \sigma_y) \quad (2.2.15)$$

この \tilde{H} には Rx 、 Ry の生成子である σ_x 、 σ_y が含まれている。したがって、この \tilde{H} を用いれば、2.1 節で紹介した $Rx(\theta)$ 、 $Ry(\theta)$ の実装が可能となる。例えばパルスのドライブ周波数を $\nu_0 = \omega_0$ に設定し、位相を $\phi = 0$ でドライブすれば、式 (2.2.15) は

$$\tilde{H}^{(x)} = \Omega_0 \sigma_x \quad (2.2.16)$$

となり、この $\tilde{H}^{(x)}$ で $\tau = \theta/2\Omega_0$ だけ時間発展させたとき、その量子状態操作はシュレディンガー方程式による時間発展演算子から、次のように与えられる。

$$\exp(-i\tilde{H}^{(x)}\tau) = e^{-i\sigma_x \frac{\theta}{2}} \quad (2.2.17)$$

これは式 (2.1.7) の $Rx(\theta)$ に他ならない。同様に $\phi = \pi/2$ にすれば、 $Ry(\theta)$ も実装できることがわかる。このように超伝導量子コンピュータでは、パルスの振幅、位相、周波数、ドライブ時間というパラメータを設定することで、量子ゲートを実装している [16][23][22]。

またドライブ周波数が $\nu_0 = \omega_0 + \delta$ と δ だけ誤差がある場合、式 (2.2.16) は

$$\tilde{H}^{(x)} = -\frac{1}{2}\delta\sigma_z + \Omega_0 \sigma_x \quad (2.2.18)$$

となり、図 2.1.1 のブロックホ球の z 軸回転成分が残っており、これは Rx 、 Ry に対して位相エラーという形で表れ、ゲートの忠実度を下げる主な要因となる [34]。 Rz については、2.2.4 節で説明する Virtual Z Gate という手法を用いてエラーなしで実装できることが知られているので [6]、式 (2.2.15) の生成子 σ_z を使用して物理的に実装することはしない。

2.2.2 節のハミルトニアンの形は d 準位系に対応しているため、トランズモンを量子トリットとして使用する場合も、今の量子ビットの説明と同様に $|0\rangle - |1\rangle$ 間や $|1\rangle - |2\rangle$ 間の遷移周波数でパルスをドライブすることで、 Rx_{01} や Rx_{12} といった量子ゲートを実装できる。

ここまで、簡易化のために単体の量子ビット系で考えて H_{int} による効果はないとして計算を行ったが、実際は常に回転系においても \tilde{H}_{int} は存在し、量子状態は \tilde{H}_{int} によるエラーの影響を受け続けている。量子トリットにおけるこの \tilde{H}_{int} エラーは本研究で解決すべき課題であり、3 章で詳しく説明することにする。

2.2.4 Virtual Z Gate

ここでは超伝導量子コンピュータでよく用いられている Virtual Z Gate という技術の説明をする [37]。2.2.3 節で紹介した式 (2.2.15) は、 $\nu_0 = \omega_0$ かつ $\phi = \phi_0$ でドライブした場合、

$$\tilde{H}' = \Omega_0 (\cos \phi_0 \sigma_x + \sin \phi_0 \sigma_y) \quad (2.2.19)$$

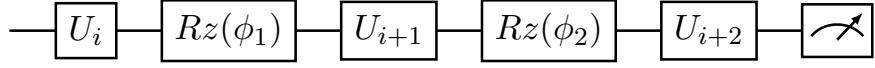
となり、 $\tau = \theta/2\Omega_0$ だけ時間発展させると、その量子ゲートは次のようになる：

$$e^{-i\tilde{H}'\tau} = e^{-i\frac{\theta}{2}(\cos\phi_0\sigma_x + \sin\phi_0\sigma_y)} \quad (2.2.20)$$

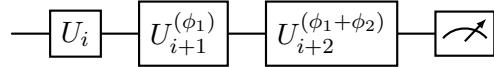
$$= Rz(-\phi_0)Rx(\theta)Rz(\phi_0) \quad (2.2.21)$$

したがってパルスの位相 ϕ は、そのパルスによる量子ゲートの前後に $Rz(\pm\phi)$ と等価な量子操作を可能にする。

この性質を利用すれば、任意の量子ゲート U に対し、次のようなゲートシーケンス



を実装したい場合、後続のパルスの位相を



と変更することで実現できる。実際に上のようなパルスの位相を変更したゲートシーケンスの計算式は、次のようになる：

$$U_{i+2}^{(\phi_1+\phi_2)}U_{i+1}^{(\phi_1)}U = Rz(-\phi_1 - \phi_2)U_{i+2}Rz(\phi_1 + \phi_2)Rz(-\phi_1)U_{i+1}Rz(\phi_1)U_i \quad (2.2.22)$$

$$= Rz(-\phi_1 - \phi_2)U_{i+2}Rz(\phi_2)U_{i+1}Rz(\phi_1)U_i \quad (2.2.23)$$

$$\sim U_{i+2}Rz(\phi_2)U_{i+1}Rz(\phi_1)U_i \quad (2.2.24)$$

最初の式変形で式 (2.2.20) の結果を利用し、最後の式変形は Rz は Z 基底の測定に影響を与えないという性質を利用した。したがって式 (2.2.22) の計算結果から、Z 基底での測定において 2 つのゲートシーケンスは等価であるとわかる。

このように、パルスの位相を用いて仮想的に実装される Rz を Virtual Z Gate[37][34][6] と呼び、全くエラーがない Rz を 0 秒で実装できることから、超伝導量子コンピュータでは Rz の主流となっている。本研究では、この技術を量子トリットの Rz_{01} 、 Rz_{12} で用いる。用途の詳細は 4.1.2 で説明する。

2.2.5 CNOT

量子トリットが絡む CNOT は 2 種類存在する。一つは制御ビットが量子ビットであり、標的ビットのみ量子トリットである場合の CNOT (以降 $CNOT_{bit}$ と記す)、もう一つは制御ビットと標的ビット共に量子トリットとして扱う場合の CNOT (以降 $CNOT_{trit}$ と記す) である。制御ビットの入力として、 $|2\rangle$ を含むか否かで CNOT の実装方法を区別する。 $CNOT_{trit}$ は $CNOT_{bit}$ の拡張版であるが、精度が $CNOT_{bit}$ よりも低いので、 $CNOT_{bit}$ で実装できるところはこちらを使いたいのである。

2.2.3 節で、

- パラメータを指定したパルスによって、ドライブハミルトニアン H_d を生成する
- 回転系ヘユニタリ変換して、有効ハミルトニアン \tilde{H} を導出する
- 時間発展させる分の時間を指定して、量子ゲートを作成する

という手順を説明した。この節では、上記の手順に則って CNOT を作成する方法を簡潔に説明する。

CNOT_{bit} の実装方法

まず CNOT_{bit} の作成に必要な交差共鳴 (Cross resonance) ゲートについて紹介する。交差共鳴ゲート (以降 CR_{bit} と記す) とは、標的ビットの量子ビット周波数で制御ビットにパルスをドライブすることで生成される量子ゲートである。制御ビットと標的ビットが結合している 2 量子ビット系において、制御ビットと標的ビットの量子ビット周波数を ω_1 と ω_2 とし、 $\Delta_{12} \equiv \omega_1 - \omega_2$ と表示したとき、回転系へ変換した後に生成される CR_{bit} の有効ハミルトニアンは、次のように与えられる [6][22][38]:

$$\tilde{H}_{CR}^{\pm} = \Omega_1 (\sigma_z \otimes I \mp \mu_{IX} I \otimes \sigma_x \pm \mu_{ZX} \sigma_z \otimes \sigma_x) \quad (2.2.25)$$

ここで、 \pm はパルスの振幅 Ω_1 の符号である。 μ_{IX} と μ_{ZX} はそれぞれ次のように定義した:

$$\mu_{IX} \equiv \frac{J_{12}}{\Delta_{12}} \frac{\delta_1}{(\delta_1 + \Delta_{12})}, \quad \mu_{ZX} \equiv \frac{J_{12}}{(\delta_1 + \Delta_{12})} \quad (2.2.26)$$

ここで、 J_{12} は式 (2.2.10) で定義した結合定数である。式 (2.2.25) の $\sigma_z \otimes \sigma_x$ はエンタングル項であり、この生成子によるゲートは次のように算出できる:

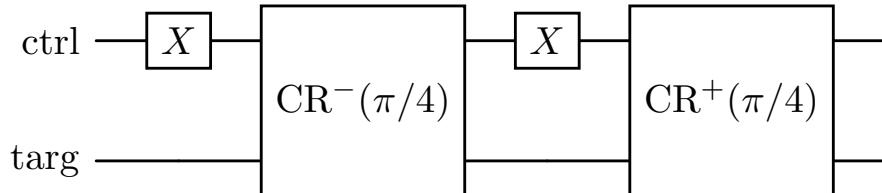
$$ZX(\theta) = e^{-\frac{i}{2}\theta(\sigma_z \otimes \sigma_x)} \\ = \begin{bmatrix} \cos \frac{\theta}{2} & -i \sin \frac{\theta}{2} & 0 & 0 \\ -i \sin \frac{\theta}{2} & \cos \frac{\theta}{2} & 0 & 0 \\ 0 & 0 & \cos \frac{\theta}{2} & i \sin \frac{\theta}{2} \\ 0 & 0 & i \sin \frac{\theta}{2} & \cos \frac{\theta}{2} \end{bmatrix} \quad (2.2.27)$$

したがって、ZX(θ) は制御ビットが $|0\rangle$ のとき標的ビットに $Rx(\theta)$ を作用させ、制御ビットが $|1\rangle$ のとき標的ビットに $Rx(-\theta)$ を作用させるようなゲートである。

この ZX 回転成分が $\theta = \pm\pi/4$ となるようにドライブした交差共鳴ゲートを CR $^{\pm}(\pi/4)$ と表示したとき、式 (2.2.25) の \tilde{H}_{CR}^{\pm} の生成子から与えられる $(\sigma_z \otimes I)$ 回転角 ϕ_0 と $(I \otimes \sigma_x)$ 回転角 ϕ_1 を用いて次のように表せる:

$$\text{CR}^{\pm}(\pi/4) = [Rz(\phi_0) \otimes Rx(\mp\phi_1)][ZX(\pm\pi/4)] \quad (2.2.28)$$

この中の ZX($\pm\pi/4$) を利用するため、次のようなゲートシーケンスを考える:



シーケンスは次のようなゲートになる:

$$\begin{aligned} & \text{CR}^+(\pi/4)(X \otimes I)\text{CR}^-(\pi/4)(X \otimes I) \\ &= ([Rz(\phi_0) \otimes Rx(-\phi_1)][ZX(\pi/4)])(X \otimes I)([Rz(\phi_0) \otimes Rx(\phi_1)][ZX(-\pi/4)])(X \otimes I) \\ &= ([Rz(\phi_0) \otimes Rx(-\phi_1)][ZX(\pi/4)])([XRz(\phi_0)X \otimes Rx(\phi_1)][(X \otimes I)ZX(-\pi/4)(X \otimes I)]) \quad (2.2.29) \\ &= ([Rz(\phi_0) \otimes Rx(-\phi_1)][ZX(\pi/4)])([Rz(-\phi_0) \otimes Rx(\phi_1)][ZX(\pi/4)]) \\ &= [I \otimes I][ZX(\pi/2)] \end{aligned}$$

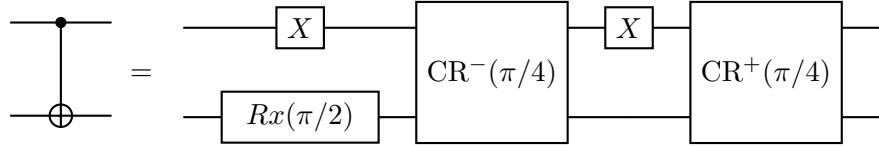


図 2.2.1: echo CNOT のゲートシーケンス

したがって、式 (2.2.27) の結果を考慮すると $ZX(\pi/2)$ は制御ビットが $|0\rangle$ のとき標的ビットに $Rx(\pi/2)$ を作用させ、制御ビットが $|1\rangle$ のとき標的ビットに $Rx(-\pi/2)$ を作用させるようなゲートである。この $ZX(\pi/2)$ に加えて、標的ビットに $Rx(-\pi/2)$ を作用させると、制御ビットが $|0\rangle$ のとき標的ビットに作用せず、制御ビットが $|1\rangle$ のとき標的ビットに $Rx(-\pi) = X$ を作用させるようなゲート、つまり $CNOT_{bit}$ となる。

この図 2.2.1 のゲートシーケンスは echo CNOT[39] と呼ばれており、式 (2.2.28) の ϕ_0 、 ϕ_1 がどのような値でも式 (2.2.29) の計算結果が保証されるため、安定して $CNOT_{bit}$ を実装できるという利点がある。式 (2.2.28) に対し、 Rz と Rx を使って $[Rz(\phi_0) \otimes Rx(\mp\phi_1)]$ をキャンセルして $ZX(\pm\pi/4)$ を残す Direct CNOT[40][41][42] という手法では、echo CNOT と比較して短時間で $CNOT_{bit}$ を実装できるが、 ϕ_0 、 ϕ_1 は時間が経つと変化してしまうため、定期的な較正が必要になる。

またこの $CNOT_{bit}$ は標的ビットが量子トリットの場合も使用可能であり、その場合は標的トリットに式 (2.1.13) の $Rx_{01}(\theta = \pi)$ で与えられる X_{01} を作用させるようなゲートとなる。

CNOT_{trit} の実装方法

$CNOT_{trit}$ も交差共鳴ゲート (CR_{trit}) を使用して作成する。このゲートは制御ビットと標的ビットが共に量子トリットであるため、結合している 2 量子トリット系を考えていく。ここで

1. 制御トリットに対し、標的トリットの量子ビット周波数かつ特定の振幅でパルス A をドライブ
2. パルス A によって標的トリットに生じた Rx_{01} 成分をキャンセルするように、標的トリットに対してカウンターパルス B(Rx_{01}) をドライブ

のような 2 個のパルスを同時にドライブする状況を考える。このパルス A とパルス B のを同時にドライブしたときの有効ハミルトニアンは、

$$z = \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad \zeta = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & -1 \end{bmatrix}, \quad x = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad (2.2.30)$$

を用いて、次のように与えられる [27]:

$$\tilde{H}_{CR}^{\pm} = \nu_{zI} \frac{zI}{2} \pm \nu_{zx} \frac{zx}{2} + \nu_{\zeta I} \frac{\zeta I}{2} \pm \nu_{\zeta x} \frac{\zeta x}{2} \quad (2.2.31)$$

kl は $k \otimes l$ 等を省略して記したもので、 ν_{kl} は有効ハミルトニアンにおける生成子 kl の係数を表しており、これらの係数はパルスの振幅に依存している。 $CNOT_{bit}$ のときと同様に、生成子 zx による $ZX(\theta)$ と生成子 ζx による $\zeta X(\theta)$ を行列表示すると、 (3×3) の零行列 0 と Rx_{01} を用いて、次のように表せる：

$$ZX(\theta) = e^{-\frac{i}{2}\theta(z \otimes x)} \quad (2.2.32)$$

$$= \begin{bmatrix} Rx_{01}(\theta) & 0 & 0 \\ 0 & Rx_{01}(-\theta) & 0 \\ 0 & 0 & I \end{bmatrix} \quad (2.2.33)$$

$$\zeta X(\phi) = e^{-\frac{i}{2}\phi(\zeta \otimes x)} \quad (2.2.34)$$

$$= \begin{bmatrix} I & 0 & 0 \\ 0 & Rx_{01}(\phi) & 0 \\ 0 & 0 & Rx_{01}(-\phi) \end{bmatrix} \quad (2.2.35)$$

(2.2.36)

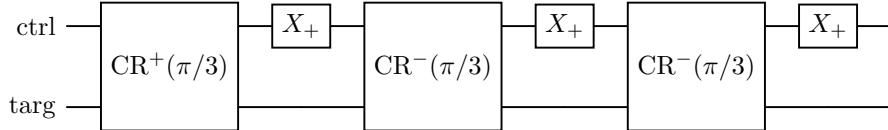
ここで式 (2.2.31) の $\nu_{zx} = -\nu_{\zeta x}$ となるパルスの振幅を指定すると、生成子 zx と ζx による回転角が $\theta = -\phi$ という関係になるため、 \tilde{H}_{CR}^{\pm} 中の生成子 zx と ζx によるゲート作用は、次のように与えられる:

$$ZX(\theta) \zeta X(-\theta) = \begin{bmatrix} Rx_{01}(\theta) & 0 & 0 \\ 0 & Rx_{01}(-2\theta) & 0 \\ 0 & 0 & Rx_{01}(\theta) \end{bmatrix} \quad (2.2.37)$$

この $ZX(\theta) \zeta X(-\theta)$ の $\theta = \pm\pi/3$ となるようにドライブした交差共鳴ゲートを $CR_{trit}^{\pm}(\pi/3)$ と表示したとき、 $CR_{trit}^{\pm}(\pi/3)$ は式 (2.2.31) の \tilde{H}_{CR}^{\pm} の生成子から与えられる $(\sigma_z \otimes I)$ 回転角 ϕ_0 と $(\zeta \otimes I)$ 回転角 ϕ_1 を用いて次のように表せる:

$$CR^{\pm}(\pi/3) = [Rz(\phi_0)R\zeta(\phi_1) \otimes I][ZX(\pm\pi/3)][\zeta X(\mp\pi/3)] \quad (2.2.38)$$

この $ZX(\pi/3) \zeta X(-\pi/3)$ を利用するため、次のようなゲートシーケンスを考える:



ここで X_+ とは、次のように定義している:

$$X_+ \equiv X_{01}X_{12} \quad (2.2.39)$$

このシーケンスは次のようなゲートとなる:

$$(X_+ \otimes I)CR^-(\pi/3)(X_+ \otimes I)CR^-(\pi/3)(X_+ \otimes I)CR^+(\pi/3) = ZX(\pi/3)\zeta X(-\pi/3) \quad (2.2.40)$$

式 (2.2.38) のゲートを左辺に代入して計算を進めると右辺になるが、計算が複雑なので途中式は付録 A.1 に記す。式 (2.2.37) の結果を考慮すると、この $ZX(\pi/3)\zeta X(-\pi/3)$ に追加して、標的トリットに $Rx_{01}(-\pi/3)$ を作用させると、制御トリットが $|0\rangle$ または $|2\rangle$ のとき標的トリットに作用せず、制御トリットが $|1\rangle$ のとき標的トリットに X_{01} を作用させるゲート、つまり $CNOT_{trit}$ になるとわかる。

この図 2.2.2 ゲートシーケンスは Qutrit echo cross-resonance CNOT として、[27] で開発されている。こちらも $CNOT_{bit}$ 同様、式 (2.2.31) の zI 、 ζI 成分が ν_{zI} 、 $\nu_{\zeta I}$ に依らず自動的にキャンセルできるため、時間が経過しても安定して実装できる。

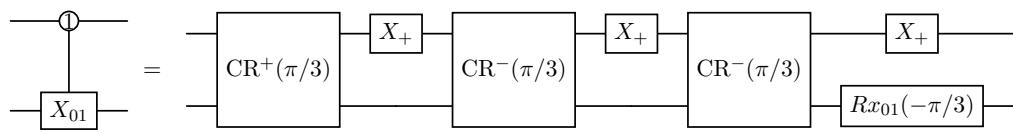


図 2.2.2: Qutrit echo cross-resonence CNOT のゲートシーケンス

第3章 量子トリット

本研究では、量子トリットを使用した n 制御ビットゲートの実装手法を開発することが目的である。量子トリットとは、トランズモンのエネルギー準位の基底状態、第一励起状態、第二励起状態を $|0\rangle$ 、 $|1\rangle$ 、 $|2\rangle$ に対応付けた 3 準位系としてトランズモンを使用したものである。2 章で説明した通り、原理的にはトランズモンは d 準位系で使用できる。しかしあるエネルギー準位が高いほど、自然放出による脱励起の影響が大きくなるため、長時間演算ではゲート忠実度が下がる。また使用する準位を増やすほど、考慮すべきエネルギー準位が増えてしまい、不必要的エネルギー遷移が励起されやすくなるため、エラーが増加する。実際に 3 準位系の量子トリットですら、3.2 節に記した 3 つの問題が生じたことから、 $4 \leq d$ 準位系の使用はゲート忠実度を大幅に下げると考えられる。量子トリットであれば、 n 制御ビットゲートである Toffoli を IBMQuantum 上で実装した先行研究 [27] が存在し、実際に忠実度が改善していたため、実用性が確認できている。したがって本研究では、実用性を考えて量子トリットを使用する。

この章では、 n 制御ビットゲートを実装する上で、量子トリットを使用する利点を 3.1 節で説明し、使用上の問題点について 3.2 節で説明する。

3.1 量子トリットの利点

n 制御ビットゲートを量子トリットで実装することで、量子ビットの際と比較して、大幅な実装コスト削減が可能になる。トランズモンが直線状に連結するトポロジーを想定する場合、 n 制御ビットゲートを実装するために、量子ビットでは最低でも $6n - 6$ 個の CNOT と 1 個の CU を必要とする [14][15]。しかし量子トリットでは、4 章で具体的に示している図 4.1.2 のようなツリー型の分解方法を用いると [43][13]、 $2n - 2$ 個の CNOT と 1 個の CU で実装が可能となる [18][19]。これにより、ゲートシーケンスの depth は 3 分の 1 になる。さらに量子ビットの実装では、 n 制御ビットゲートに使用する $n + 1$ 個の量子ビットに加えて $n - 2$ 個の作業ビットが必要になるが、量子トリットは $|2\rangle$ が作業ビットの役割を担うため、作業ビットが不要になる。これらを考慮すると、ゲート数と depth の観点で見れば、量子トリットによる n 制御ビットゲートの実装は、量子ビットでの実装に対し、完全な上位互換と言える。

このように量子トリットを用いて n 制御ビットゲートを効率よく量子コンピュータで実装する場合、量子トリットゲートを高忠実度で実装することが重要となる。本研究で必要となる量子トリットゲートは、量子トリット CNOT と X_{12} であるが、これらの量子ゲートは Toffoli の先行研究 [27] によって、qiskit.pulse というモジュールを使用したパルスレベルでの実装方法が確立されている。この先行研究の結果は、量子ビットで実装した Toffoli より高い忠実度を得ているため、本研究で必要な量子トリットゲートは、量子ビットでの実装精度を上回るのに十分な忠実度で実装が可能であると言える。

3.2 量子トリット使用上の問題点

量子トリットを使用して、 n 制御ビットゲートを実装するためには、Toffoli の先行研究 [27] で既に発見されていた 2 つのエラーに加えて、本研究で新たに発覚したエラーを抑制する必要がある。この節では、超伝導量子コンピュータ上で量子トリットを使用して n 制御ビットゲートを実装する際に、忠実度を下げる主な要因となる 3 つのエラーを説明する。

3.2.1 ベリー位相

ベリー位相とは、量子力学において量子系がパラメータ空間を閉じたループを描くときに生じる位相の変化を指す。これは物理系が位相的に変化する際に生じる非古典的な効果であり、古典的な位相とは異なり、系がパラメータを変える間に蓄積される位相シフトである [44]。このベリー位相は量子ゲートを実装する際に表れる [45]。具体的に X_{01} を行列表示することで、ベリー位相の説明をする。

式 (2.2.16) で算出した通り、トランズモンの $|0\rangle-|1\rangle$ 間の周波数に合わせてパルスをドライブすると、有効ハミルトニアンによる時間発展は次のように与えられた:

$$\tilde{H}_d^{(x)} = \Omega_0 \lambda_1 \quad (3.2.1)$$

λ_1 は、2.1 節で定義した量子トリット空間の生成子である。この $\tilde{H}_d^{(x)}$ で、 $\Omega_0 \tau = \frac{\pi}{2}$ となる時間 τ だけパルスをドライブしたとき、量子ゲートの行列表現は次のように与えられる:

$$\exp\left(-i\tilde{H}_d^{(x)}\frac{\tau}{2}\right) = \exp\left(-i\frac{\pi}{2}\lambda_1\right) = -i \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & i \end{bmatrix} \sim \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & i \end{bmatrix} \quad (3.2.2)$$

最後の変形では、絶対位相 $-i$ を無視した。この最右辺は、 $|0\rangle-|1\rangle$ 間の状態を反転する X_{01} となっている。量子ビットゲートである X は相対位相がなかったが、量子トリット空間では式 (3.2.2) の通り、 $|2\rangle$ に相対位相 i がかかっている。量子ゲートにおけるベリー位相とはこの相対位相のことを指し、量子ビットゲートでは考慮する必要がなかったが、量子トリットゲートでは考慮することになる。このベリー位相は、量子トリットを用いた n 制御ビットゲートの分解方法を提案している参考文献 [43][18][19] の中では考慮されていない。また先行研究 [27] の中でも、任意の n をとる n 制御ビットゲートに対し、ベリー位相を解消する汎用的なゲートシーケンスの組み方は明記されていない。特に 4.1.1 節で見るよう、本研究で必要な量子トリットゲートには全てベリー位相がついているため、任意の数の制御ビットを持つ n 制御ビットゲートに対し、ベリー位相が出力に影響しない汎用的なゲートシーケンスを新たに開発する必要がある。

3.2.2 ZZ 相互作用

ZZ 相互作用とは、2 個のトランズモン間が結合していることで生じる相互作用であり [46]、トランズモン間で位相変化を発生させる。これにより 2 個のトランズモンがエンタングル状態になり、相関が生じる。ZZ 相互作用は、特に量子エラー訂正や量子ゲート操作の設計において注意深く取り扱われる重要な要素である。

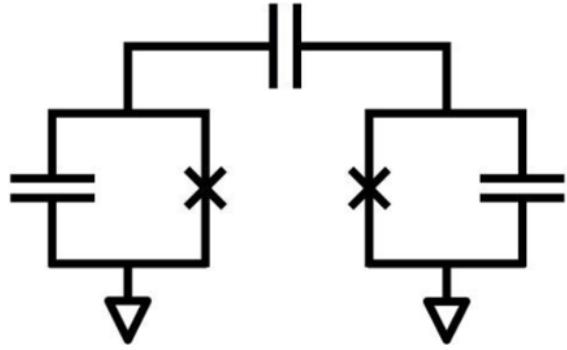


図 3.2.1: 2 トランズモン系の回路図。トランズモン間はコンデンサーで結合される。これにより CR 等のエンタングル操作を可能にするが、ZZ 相互作用も生じる。

ZZ 相互作用の発生原理を簡単に紹介する。2 個のトランズモン (q_1, q_2) が図 3.2.1 のように結合している系を考えたとき、系のハミルトニアンは 2.2.2 節で定義した H_0, H_{int} を用いて、次のように与えられる：

$$H_{\text{stat}} = H_0 + H_{\text{int}} \quad (3.2.3)$$

$$= \sum_{j=1,2} \omega_j b_j^\dagger b_j + J_{12}(b_1^\dagger b_2 + b_1 b_2^\dagger) \quad (3.2.4)$$

ここで $E_i^{(2)} \equiv H_{\text{stat}} |0\rangle \otimes |i\rangle$ を用いて、回転系へ変換する前の座標系における、 q_1 が $|0\rangle$ のときの q_2 の $|0\rangle - |1\rangle$ 遷移周波数 $E_1^{(2)} - E_0^{(2)}$ を算出してみる。式 (3.2.3) から

$$E_1^{(2)} - E_0^{(2)} = \omega_2 + J_{12} \quad (3.2.5)$$

となる。一方、 q_1 が $|1\rangle$ の際の遷移周波数も同様に算出すると、

$$E_1^{(2)} - E_0^{(2)} = \omega_2 - J_{12} \quad (3.2.6)$$

となる。したがって H_{int} によって、特定の量子ビットの遷移周波数が他の量子ビットの状態に依存しているとわかる。パルスのドライブ周波数は $\omega_2 \pm J_{12}$ のどちらかで固定する必要があるため、 q_1 の状態に依存して非共鳴効果が生じてしまう。この H_{stat} を 2.2.2 節で定義した U_q によって回転系へユニタリ変換すると、ある係数 Δ を用いて次のような形になる：

$$U_q H_{\text{stat}} U_q^\dagger = \tilde{H}_{\text{int}} = \Delta \sigma_z^{(1)} \otimes \sigma_z^{(2)} \quad (3.2.7)$$

ここで $\sigma_z^{(i)}$ は i 番目の量子ビットに作用するパウリ Z 行列であることを指す。回転系において、先ほど H_{int} による非共鳴効果はこの $\sigma_z^{(1)} \otimes \sigma_z^{(2)}$ 効果として記述され、このエンタングル効果を ZZ 相互作用と呼ぶ。

量子トリットの場合は、更に複雑な ZZ 相互作用が生じる。図 3.2.1 のように結合している 2 つのトランズモン (q_1, q_2) を量子トリットとして考えたとき、系のハミルトニアンは 2.2.2 節で定義した H_0, H_{int}

を用いて、次のように与えられる:

$$H_{\text{stat}} = H_0 + H_{\text{int}} \quad (3.2.8)$$

$$= \sum_{j=1,2} \left[\omega_j b_j^\dagger b_j + \frac{\delta_j}{2} b_j^\dagger b_j (b_j^\dagger b_j - 1) \right] + J_{12}(b_1^\dagger b_2 + b_1 b_2^\dagger) \quad (3.2.9)$$

ここで q_1 が $|0\rangle$ の際の q_2 の $|0\rangle - |1\rangle$ 遷移周波数 $E_1^{(2)} - E_0^{(2)}$ を算出すると、式 (3.2.8) から

$$E_1^{(2)} - E_0^{(2)} = \omega_2 + J_{12} \quad (3.2.10)$$

となり、 q_1 が $|1\rangle$ では

$$E_1^{(2)} - E_0^{(2)} = \omega_2 - J_{12} \quad (3.2.11)$$

となり、 q_1 が $|2\rangle$ では

$$E_1^{(2)} - E_0^{(2)} = \omega_2 \quad (3.2.12)$$

となる。またこの関係に加えて、 $|1\rangle - |2\rangle$ 遷移周波数 $E_2^{(2)} - E_1^{(2)}$ も算出してみる。式 (3.2.8) から q_1 が $|0\rangle$ では、

$$E_2^{(2)} - E_1^{(2)} = \omega_2 + \frac{\delta_2}{2} \quad (3.2.13)$$

となり、 q_1 が $|1\rangle$ では

$$E_2^{(2)} - E_1^{(2)} = \omega_2 + \frac{\delta_2}{2} + \sqrt{2}J_{12} \quad (3.2.14)$$

となり、 q_1 が $|2\rangle$ では

$$E_2^{(2)} - E_1^{(2)} = \omega_2 + \frac{\delta_2}{2} - \sqrt{2}J_{12} \quad (3.2.15)$$

となる。よって計算結果を見ると、 $|0\rangle - |1\rangle$ 遷移周波数と比較して、 $|1\rangle - |2\rangle$ 遷移周波数は ZZ 相互作用による非共鳴効果が $\sqrt{2}$ 倍だけ大きい [47] ので、量子トリットを使用すると、より大きな影響を持つ ZZ 相互作用が生じる。

また式 (2.1.12) のゲルマン行列を用いて、量子トリット空間での ZZ 相互作用の生成子を記述すると、量子ビット空間で見えていた $\lambda_3 \otimes \lambda_3$ に加えて、 $\lambda_3 \otimes \lambda_8$ 、 $\lambda_8 \otimes \lambda_3$ 、 $\lambda_8 \otimes \lambda_8$ がある。したがって、量子トリットの ZZ 相互作用は量子ビットのときと比べて種類も増える。

実際、量子トリット特有の λ_8 を含んだ ZZ 相互作用が、 $\lambda_3 \otimes \lambda_3$ と比較して、大きく忠実度を下げていることが本研究中に判明した。図 3.2.2 は、ibmq-cairo の (q_7, q_{10}) の結合の ZZ 相互作用を 5.1 節で説明する qudit-sim というシミュレータによって算出したときの各生成子の係数である：

この結果は、

$$\tilde{H}_{\text{int}} = \nu_{33}(\lambda_3 \otimes \lambda_3) + \nu_{38}(\lambda_3 \otimes \lambda_8) + \nu_{83}(\lambda_8 \otimes \lambda_3) + \nu_{88}(\lambda_8 \otimes \lambda_8) \quad (3.2.16)$$

と表示したときの係数を縦軸としている。

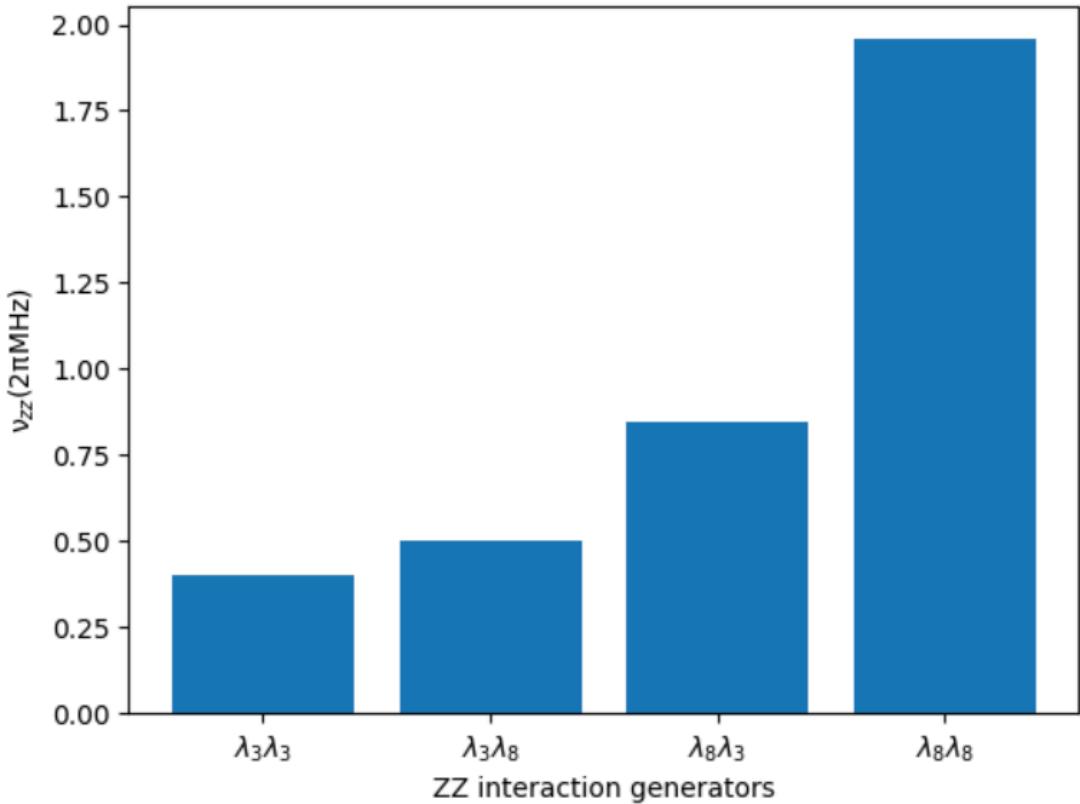


図 3.2.2: 量子トリット空間の \tilde{H}_{int} に含まれる ZZ 相互作用の係数を生成子ごとに表示した図。量子ビットでも見えていた $\lambda_3 \otimes \lambda_3$ と比べて、 $\lambda_3 \otimes \lambda_8$ 、 $\lambda_8 \otimes \lambda_3$ 、 $\lambda_8 \otimes \lambda_8$ は影響が大きく、特に $\lambda_8 \otimes \lambda_8$ は $\lambda_3 \otimes \lambda_3$ の 5 倍程度大きい。

2.2.2 節で述べた通り、 H_{stat} は H_d とは異なり、パルスをドライブしていない時間も常に時間発展を起こすため、 H_{stat} に含まれているエラーが小さくとも長時間の演算であるほど大きなエラーを生む。本研究で対象としている n 制御ビットゲートは、他の量子ゲートと比べて長時間の演算になるため、ZZ 相互作用への対策は不可欠である。特に量子トリットの使用により、新たに生じる $\lambda_3 \otimes \lambda_8$ 、 $\lambda_8 \otimes \lambda_3$ 、 $\lambda_8 \otimes \lambda_8$ への対策は先行研究 [27] に明記されておらず、前例がないため、本研究で対策設計を新規で開発する必要がある。

3.2.3 電荷分散

電荷分散とは、エネルギー準位が環境のオフセット電荷やゲート電圧に対してどれだけ変動するかを示すものである。言い換えると、トランズモンのエネルギー準位が外部の電荷や電圧の変化に対してどれだけ敏感かを表している [48]。

電荷分散が小さいほど、トランズモンの各エネルギー準位に対応する周波数は、外部の電荷ノイズに対して安定している。一方で、電荷分散が大きい場合、外部の電荷変動がトランズモンのエネルギー準位に大きな変化を引き起こし、トランズモンやゲートの性能に影響を与える可能性が高まる。そのため、電荷分散の小ささは、トランズモンが外部の影響に対して剛健であることを意味し、安定した動作を実現するために重要である。

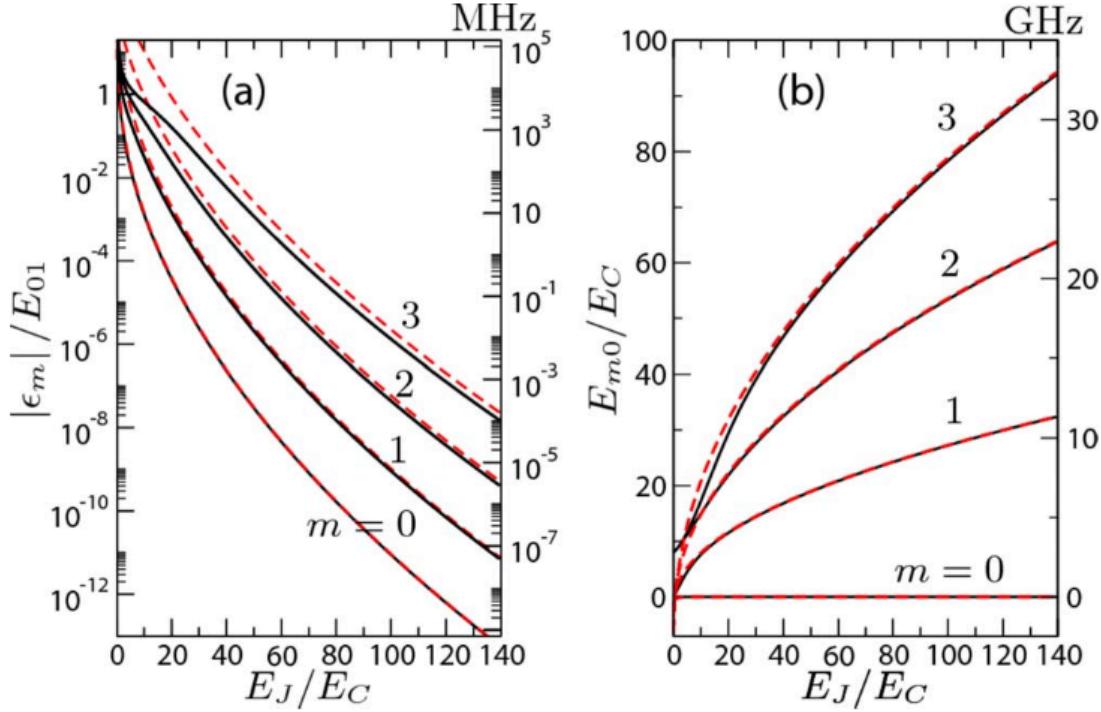


図 3.2.3: E_J/E_c に対する電荷分散(左)とエネルギー準位(右)の関係。黒実線は Mathieu 特性値というものを使用した正確な値、赤点線は漸近展開を利用した値を表している。 m はエネルギー準位数を示し、 ϵ_m は m 状態の電荷分散である。左図は E_J/E_c が大きいほど ϵ_m が小さく、右図は E_J/E_c が大きいほど $\delta = E_{m+1,m} - E_{m,m-1}$ が小さくなることを示す。[48]

図 3.2.3 の左図は、電荷分散 ϵ_m (縦軸) がコンデンサーのエネルギー E_c とジョセフソン接合のエネルギー E_J の比率 E_J/E_c (横軸) に依存していることを表しており、 E_J/E_c が大きいほど、電荷分散が小さくなる。よって、安定した動作を保証するためには E_J/E_c は大きくしたい。

しかし非調和性 δ も E_J/E_c に依存している。図 3.2.3 の右図は、縦軸 E_{m0} 、横軸 E_J/E_c であり、 $E_J/E_c = 20$ と $E_J/E_c = 140$ における $\delta = E_{m+1,m} - E_{m,m-1}$ を比較すると、 E_J/E_c が大きくなるほど δ は小さくなっていることがわかる。2.1 節で説明した通り、トランズモンの非調和性が大きいほど、量子状態操作の際に不要なエネルギー遷移を励起する可能性が低くなり、演算精度が向上する。 E_J/E_c が小さいほど、非調和性は大きくなるので、演算精度を向上させるためには E_J/E_c を小さくしたい。

これらの関係を考慮して、量子ビットとして使用するために、トランズモンは非調和性を保ったまま、 $|1\rangle$ の電荷分散 ϵ_1 が無視できるほど小さくなる E_J/E_c の値で設計される。

しかし量子トリットとしてトランズモンを使用する際は、 ϵ_2 は ϵ_1 の数十倍大きいことが左図 3.2.3 から読み取れるため、 $|2\rangle$ の電荷分散 ϵ_2 を主要なエラーとして考慮する必要がある。

この電荷分散の対策をしない場合、 X_{12} のようなゲート較正の際にエネルギー準位 E_2 を用いるゲートの忠実度が下がってしまう。具体的に電荷分散で X_{12} にエラーが生じる仕組みを説明する。図 3.2.4 のように、 X_{12} はドライブ周波数を $|1\rangle - |2\rangle$ 間の遷移周波数で較正する。この較正のための実験の段階で、 $|1\rangle - |2\rangle$ 間の遷移周波数は左 3.2.4 の左図のように定まったとする。しかしこの較正した X_{12} を他の実験で使用するときには、図 3.2.4 の右図のようにエネルギー準位 E_2 が電荷分散で既に変動している

ので、非共鳴効果を含んだゲートとなってしまっている。この非共鳴効果は後で見る式(4.3.1)のように X_{12} に位相エラーが付随するという形になり、ゲート忠実度を大きく下げてしまう。

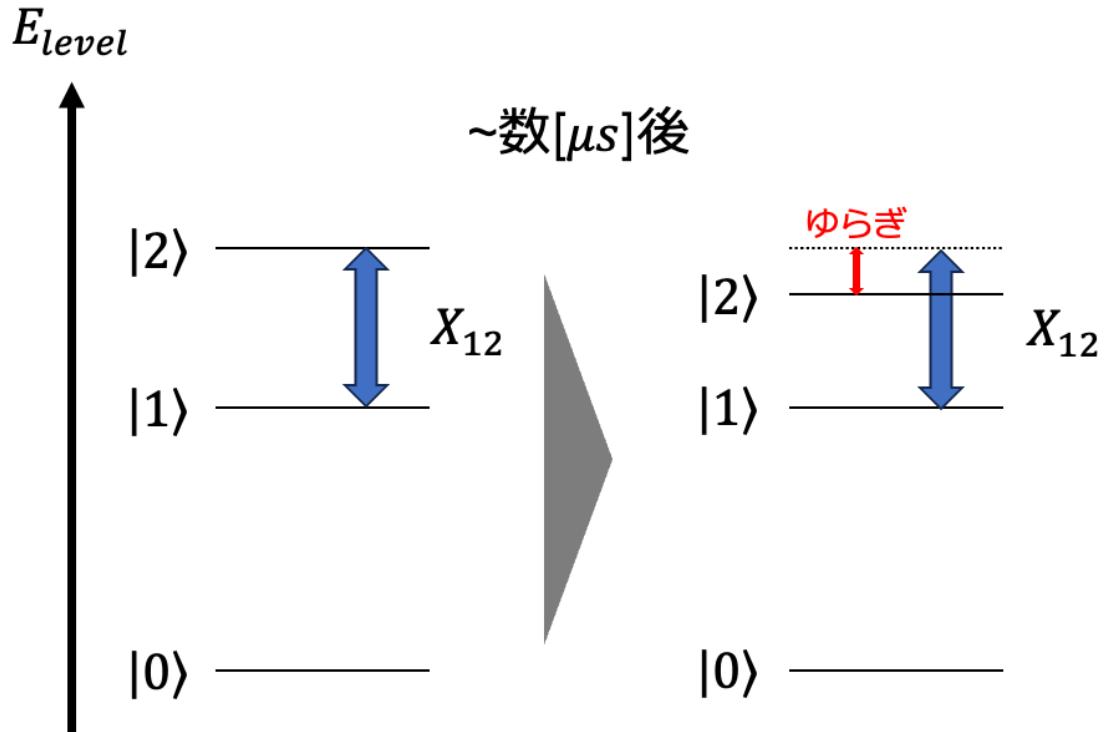


図 3.2.4: 電荷分散により X_{12} の較正に誤差が出る様子。数 [μs] で $|2\rangle$ が揺らいでいるため、ある時刻で X_{12} のために $|1\rangle - |2\rangle$ 遷移周波数を較正しても、その X_{12} を使用するまでにエラーを持つゲートとなってしまう。

よって、量子トリットを使用して忠実度の高いゲートを実装するためには、電荷分散 ϵ_2 への対策が不可欠となる。先行研究 [27] では、Toffoli 用の電荷分散対策設計を考えられていたが、 n 制御ビットゲートに拡張できる設計ではない。したがって Toffoli で使われている技術を参考に、 n 制御ビットゲート用に新しく設計し直す必要がある。

第4章 量子トリット n 制御ビットゲートの設計

本章では、IBMQQuantum 上で量子トリットを使用して、 n 制御ビットゲートを実装するためのゲート設計を紹介する。[4.1](#) 節では n 制御ビットゲートを量子トリット上で分解する方法とベリー位相を解消する設計、[4.2](#) 節では量子トリットを使用することで生じる ZZ 相互作用による位相エラーを解消する原理と設計、[4.3](#) 節では電荷分散によって生じる位相エラーを解消する原理と設計を紹介する。

4.1 n 制御ビットゲート分解方法

n 制御ビットゲートは、量子トリットを用いて余分な作業ビットや SWAP ゲートを使用することなく図 [4.1.2](#) のように分解できることが参考文献 [18] に載っている。しかし量子トリットを用いた Toffoli ゲートの分解方法は参考文献 [13] で図 [4.1.1](#) のように複数あることが記されている。したがってこの分

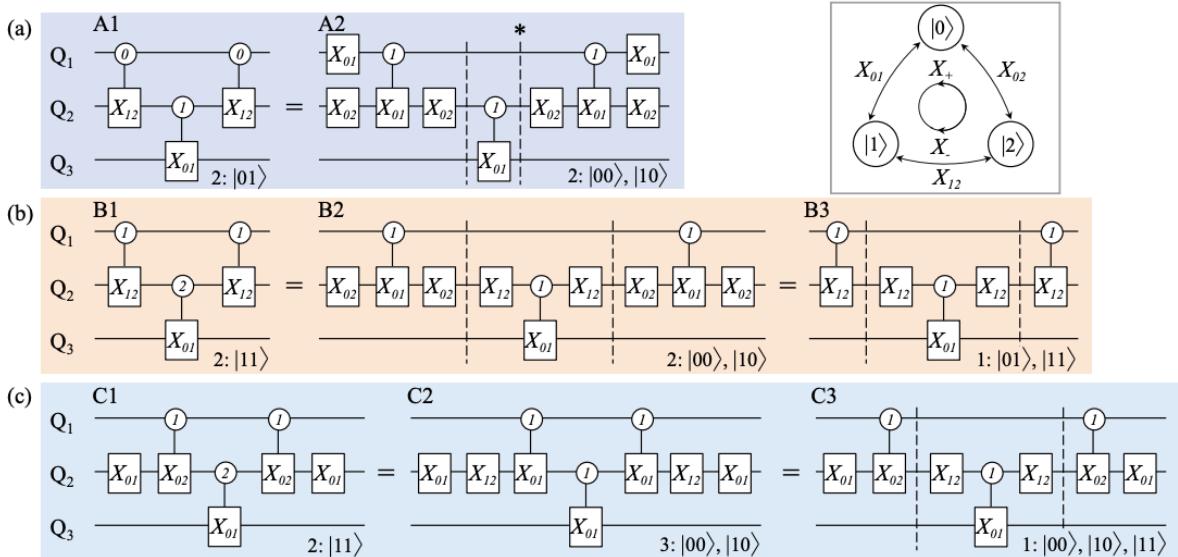


図 4.1.1: 量子トリットを使用した Toffoli の分解方法。ここで記されている X_{01} や X_{12} 、 X_+ 等は [2.1.2](#) 節で紹介したものと同じ定義であり、各ゲートによる状態の変え方が右上で紹介されている。[13]

解方法を参考にすれば、Toffoli の拡張版である n 制御ビットゲートも分解方法は複数考えられる。その中で先行研究 [27] と同じように図 [4.1.1](#) の C2 を参考に n 制御ビットゲートの分解方法を考えると、先行研究によって、シーケンスで使用する全てのゲートの作成方法が確立されているため、本研究では n 制御ビットゲートを C2 を参考にして図 [4.1.2](#) のように分解した。この分解方法を用いると、制御ビットが n 個の場合、CNOT は $2n - 1$ 個で n 制御ビットゲートが実装できる。量子ビットで分解した際には CNOT を $6n - 5$ 個使用することを考えると、CNOT の使用数を大幅に削減できていることがわかる。

C2以外の分解方法を参考にした n 制御ビットゲートの方が良い精度となる場合も考えられるが、新たにゲートの実装方法の開発から行う必要があったため、本研究では触れずに今後の課題とした。

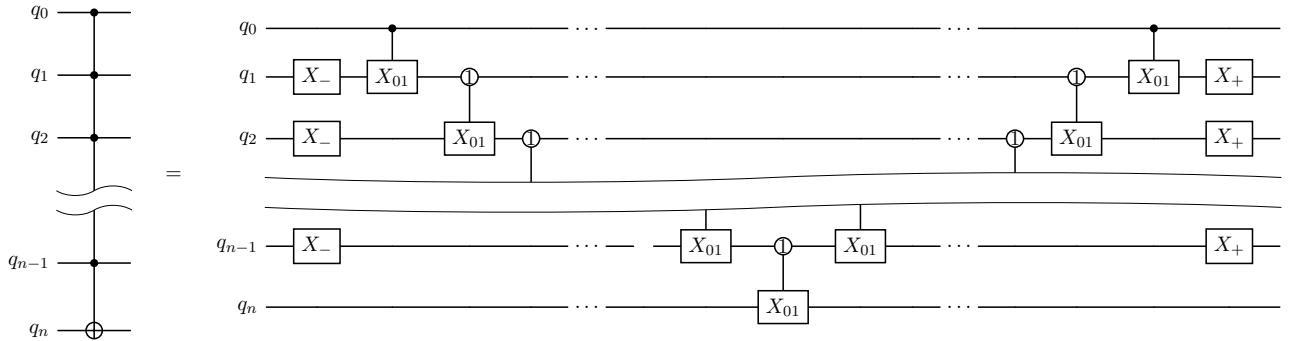


図 4.1.2: 図 4.1.1 の C2 を参考にした量子トリット上での n 制御ビットゲートの分解方法。 $X_+ \equiv X_{01}X_{12}$ 、 $X_- \equiv X_{12}X_{01}$ であるため、C2 と同様の分解方法になっている。

図 4.1.2 の分解方法はベリー位相が対策されておらず、位相エラーを含んだ n 制御ビットゲートになっている。したがって、これから n 制御ビットゲートのベリー位相を対策する手法を具体的に説明していく。まず 4.1.1 節で n 制御ビットゲートのシーケンスで使用する量子トリットゲートを行列表現で紹介し、ベリー位相があることを確認する。そして 4.1.2 節で各ゲートのベリー位相が n 制御ビットゲートの機能に影響を与えないように、図 4.1.2 のシーケンスを設計し直す。

以下、「 i, j, \dots 番目のトランズモン q に作用する量子ゲート」という意味で (q_i, q_j, \dots) という記法を用いる。

4.1.1 使用するゲートの行列表現

单一量子トリットゲート

n 制御ビットゲートで用いる単一量子トリットゲートは、 X_+ 、 X_- である。これらは、2.1.2 節で定義したように、ゲルマン行列を用いて表される：

$$X_{01} \equiv \exp\left(-i\frac{\pi}{2}\lambda_1\right) = \begin{bmatrix} 0 & -i & 0 \\ -i & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (4.1.1)$$

$$X_{12} \equiv \exp\left(-i\frac{\pi}{2}\lambda_6\right) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & -i \\ 0 & -i & 0 \end{bmatrix} \quad (4.1.2)$$

というゲートを用いて、それぞれ次のように定義される:

$$X_+ \equiv X_{01}X_{12} = \begin{bmatrix} 0 & 0 & -1 \\ -i & 0 & 0 \\ 0 & -i & 0 \end{bmatrix} \sim \begin{bmatrix} 0 & 0 & -i \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \quad (4.1.3)$$

$$X_- \equiv X_{12}X_{01} = \begin{bmatrix} 0 & -i & 0 \\ 0 & 0 & -i \\ -1 & 0 & 0 \end{bmatrix} \sim \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -i & 0 & 0 \end{bmatrix} \quad (4.1.4)$$

このゲートは図 4.1.1 の右上にも紹介されているように、 X_+ は状態を

$$|0\rangle \rightarrow |1\rangle \rightarrow |2\rangle \rightarrow |0\rangle \rightarrow \dots$$

と変化させ、 X_- は状態を

$$|0\rangle \rightarrow |2\rangle \rightarrow |1\rangle \rightarrow |0\rangle \rightarrow \dots$$

と変化させるゲートである。式 (4.1.3) の行列表現を見ると、ベリ一位相があることが確認できる。

CNOT_{bit}

n 制御ビットゲートのシーケンスにある (q_0, q_1) CNOT は制御ビット q_0 を量子ビットとして使用しているので、2.2.5 節で紹介した echo CNOT を使用する [39]。 q_1 は量子トリットとして使用するので、行列表現は(量子ビット dim2 \otimes 量子トリット dim3) の (6×6) 行列となる。CNOT_{bit} の行列表現は図 2.2.1 のゲートシーケンスを行列計算することで、次のように表される:

$$\text{CNOT}_{\text{bit}} = \text{CR}^+(\pi/4) (X \otimes I) \text{CR}^-(\pi/4) (X \otimes I) (I \otimes Rx_{01}(\pi/2)) \quad (4.1.5)$$

$$= \begin{bmatrix} 1 & 0 & 0 & & & \\ 0 & 1 & 0 & & 0 & \\ 0 & 0 & 1 & & & \\ & & & 0 & -i & 0 \\ 0 & & & -i & 0 & 0 \\ & & & & 0 & 0 & 1 \end{bmatrix} \quad (4.1.6)$$

この行列表現から、CNOT_{bit} は制御ビットが $|0\rangle$ のとき標的トリットに作用せず、制御ビットが $|1\rangle$ のとき標的トリットに X_{01} を作用させるゲートとわかる。また標的トリットに X_{01} を作用させると、ベリ一位相があることが確認できる。

CNOT_{trit}

n 制御ビットゲートのシーケンスにある (q_0, q_1) CNOT 以外の CNOT は制御ビット標的ビット共に量子トリットとして使用しているので、2.2.5 節で紹介した Qutrit echo cross-resonance CNOT を使用す

る [27]。行列表現は、(量子トリット $\text{dim}3 \otimes$ 量子トリット $\text{dim}3$) から (9×9) 行列となる。 $CNOT_{\text{trit}}$ の行列表現は図 2.2.2 のゲートシーケンスを行列計算することで、次のように表される：

$$CNOT_{\text{trit}} = \left(I \otimes Rx_{01} \left(-\frac{\pi}{3} \right) \right) (X_+ \otimes I) \text{CR}^- \left(\frac{\pi}{3} \right) (X_+ \otimes I) \text{CR}^- \left(\frac{\pi}{3} \right) (X_+ \otimes I) \text{CR}^+ \left(\frac{\pi}{3} \right) \quad (4.1.7)$$

$$= \begin{bmatrix} 1 & 0 & 0 & & & & & & \\ 0 & 1 & 0 & 0 & 0 & & & & \\ 0 & 0 & 1 & & & & & & \\ & & & 0 & -i & 0 & & & \\ & & & 0 & -i & 0 & 0 & & \\ & & & & 0 & 0 & 1 & & \\ & & & & & 1 & 0 & 0 & \\ 0 & & 0 & & & 0 & 1 & 0 & \\ & & & & & 0 & 0 & 1 & \end{bmatrix} \quad (4.1.8)$$

この行列表現から、 $CNOT_{\text{trit}}$ は制御トリットが $|0\rangle$ 、 $|2\rangle$ のとき標的トリットには作用せず、制御トリットが $|1\rangle$ のとき標的トリットに X_{01} が作用するゲートであるとわかる。また標的トリットに X_{01} を作用させるとき、ベリー位相があることが確認できる。

4.1.2 ベリー位相への対策

4.1.1 節で見たように、図 4.1.2 の n 制御ビットゲートシーケンスに含まれるゲートには、ベリー位相がある。これらが n 制御ビットゲートの出力に影響しないように、シーケンスを工夫していく。

まず、中央の $(q_{n-1}, q_n)CNOT_{\text{trit}}$ 以外の全てのゲートのベリー位相を解消する手法を説明する。今回使用する量子トリット n 制御ビットゲートのゲートシーケンスは、中央の $(q_{n-1}, q_n)CNOT_{\text{trit}}$ を除いて左右対象のゲート配置をしている。したがって、図 4.1.3 で示すように左半分のゲート群を大きなゲート O として考えたとき、右半分のゲートは O^\dagger というエルミート行列で実装できる。

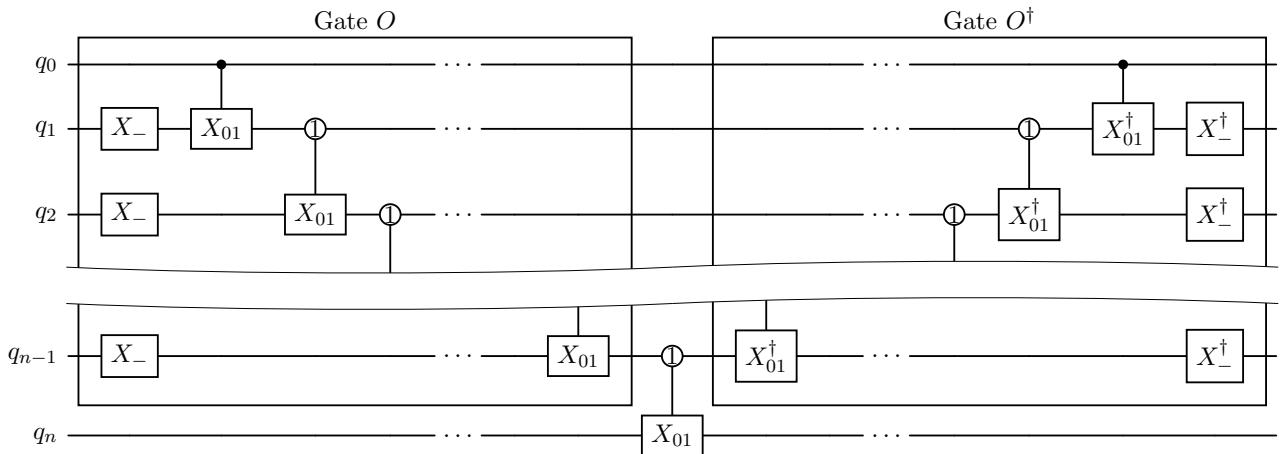


図 4.1.3: 量子トリット n 制御ビットゲートシーケンスの右半分を O^\dagger ゲートに変更したシーケンス。このシーケンスで、中央の $(q_{n-1}, q_n)CNOT_{\text{trit}}$ 以外の全てのゲートのベリー位相を解消できる。

このように右半分のゲート群を O^\dagger に置き換えた場合、中央の (q_{n-1}, q_n) CNOT_{trit} を除いたゲートシーケンスの出力は、ユニタリ行列の定義から次のように与えられる：

$$O^\dagger O = I \quad (4.1.9)$$

出力がアイデンティティであるとは、言い換えると出力にベリー位相が残っていないということである。つまり O^\dagger に含まれる個々のゲートにはベリー位相が残っていたとしても、シーケンス全体ではベリー位相が解消できる設計になっている。この手法を利用すれば、中央の (q_{n-1}, q_n) CNOT_{trit} 以外の全てのゲートのベリー位相を解消できる。

この手法を利用するためには、右半分のゲート全てをダガー付きゲートとして実装する必要がある。つまり O^\dagger に含まれる X_-^\dagger 、CNOT_{bit}[†]、CNOT_{trit}[†] をパルスで実装する必要がある。そこで新たにこれらのゲートの実装方法を考案した。

まず X_-^\dagger についてだが、

$$X_-^\dagger = (X_{12} X_{01})^\dagger = X_{01}^\dagger X_{12}^\dagger \quad (4.1.10)$$

という関係が成り立つので、 X_{01}^\dagger 、 X_{12}^\dagger が実装できれば良い。ここで 2.2.2 節で紹介したパルスによる X_{01} の有効ハミルトニアン（式 (2.2.16)）まで立ち返ると、ハミルトニアンの符号はパルス振幅 Ω_0 に依存していることがわかる。そこでパルス振幅を負にした $-\tilde{H}^{(x)}$ で回転角 π だけ時間発展するとこの時間発展による量子トリットゲートは次のように計算できる：

$$e^{-i(-\tilde{H}^{(x)}\tau)} = e^{i\lambda_1 \frac{\pi}{2}} = \begin{bmatrix} 0 & i & 0 \\ i & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (4.1.11)$$

ベリー位相が式 (4.1.1) と比べて反転しており、これは X_{01}^\dagger に他ならない。同様の原理を用いれば、 X_{12} もパルス振幅を負にすることで、 X_{12}^\dagger を実装することができる。これらを用いて X_-^\dagger は実装できる。

次に CNOT_{bit}[†] について説明する。CNOT_{bit} のゲートシーケンス（図 2.2.1）は、行列で表現すると式 (4.1.5) のようになる。ここから CNOT_{bit}[†] のゲートシーケンスは、行列計算により次のように構成できる：

$$\text{CNOT}_{\text{bit}}^\dagger = (CR^+(\pi/4)(X \otimes I)CR^-(\pi/4)(X \otimes I)(Rx_{01}(\pi/2) \otimes I))^\dagger \quad (4.1.12)$$

$$= (Rx_{01}(\pi/2)^\dagger \otimes I)(X^\dagger \otimes I)CR^-(\pi/4)^\dagger(X^\dagger \otimes I)CR^+(\pi/4)^\dagger \quad (4.1.13)$$

$$= (Rx_{01}(\pi/2)^\dagger \otimes I)(X \otimes I)CR^+(\pi/4)(X \otimes I)CR^-(\pi/4) \quad (4.1.14)$$

ここで二段目から三段目の式変形は $X = X^\dagger$ と $CR^\pm(\pi/4)^\dagger = CR^\mp(\pi/4)$ を利用している。CNOT_{bit} の制御ビットは量子ビットであるため、 X_{01} ではなく X と表記しており、量子ビットゲートはベリー位相がないことから

$$X = X^\dagger = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$$

が成立する。

また上の X_-^\dagger で説明した通り、 $CR^\pm(\pi/4)^\dagger$ を実装するためには有効ハミルトニアンのパルス振幅の符号を反転させれば良く、この CR^\pm の肩の符号は、式 (2.2.25) の有効ハミルトニアンのパルス振幅の符

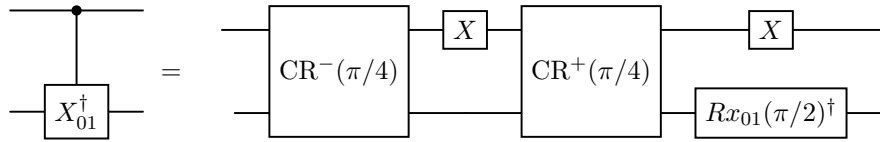


図 4.1.4: $\text{CNOT}_{\text{bit}}^\dagger$ のゲートシーケンス

号により定義されていたため、 $CR^\pm(\pi/4)^\dagger = CR^\mp(\pi/4)$ という関係が成り立つ。以上から、 $\text{CNOT}_{\text{bit}}^\dagger$ は図 4.1.4 のようなシーケンスで実装できる。

最後に $\text{CNOT}_{\text{trit}}^\dagger$ について説明する。 $\text{CNOT}_{\text{trit}}$ のゲートシーケンス (図 2.2.2) は、行列で表現すると次のようになる：

$$\text{CNOT}_{\text{trit}} = (I \otimes Rx_{01}(-\pi/3)) (X_+ \otimes I) CR^-(\pi/3) (X_+ \otimes I) CR^+(\pi/3) (X_+ \otimes I) CR^+(\pi/3) \quad (4.1.15)$$

ここから $\text{CNOT}_{\text{trit}}^\dagger$ のゲートシーケンスは、行列計算によって次のように構成できる：

$$\text{CNOT}_{\text{trit}}^\dagger = ((I \otimes Rx_{01}(-\pi/3)) (X_+ \otimes I) CR^-(\pi/3) (X_+ \otimes I) CR^-(\pi/3) (X_+ \otimes I) CR^+(\pi/3))^\dagger \quad (4.1.16)$$

$$= CR^+(\pi/3)^\dagger (X_+^\dagger \otimes I) CR^-(\pi/3)^\dagger (X_+^\dagger \otimes I) CR^-(\pi/3)^\dagger (X_+^\dagger \otimes I) (I \otimes Rx_{01}(-\pi/3)^\dagger) \quad (4.1.17)$$

$$= CR^-(\pi/3) (X_+^\dagger \otimes I) CR^+(\pi/3) (X_+^\dagger \otimes I) CR^+(\pi/3) (X_+^\dagger \otimes I) (I \otimes Rx_{01}(-\pi/3)^\dagger) \quad (4.1.18)$$

ここで二段目から三段目の式変形は $CR^\pm(\pi/3)^\dagger = CR^\mp(\pi/3)$ を利用しており、理由は式 (2.2.31) の肩の符号の定義の仕方によるもので、 CNOT_{bit} と全く同じである。以上から、 $\text{CNOT}_{\text{trit}}^\dagger$ は図 4.1.5 のようなシーケンスで実装できる。以上で O^\dagger が作成できるので、 $(q_{n-1}, q_n)\text{CNOT}_{\text{trit}}$ 以外のベリー位相対策設

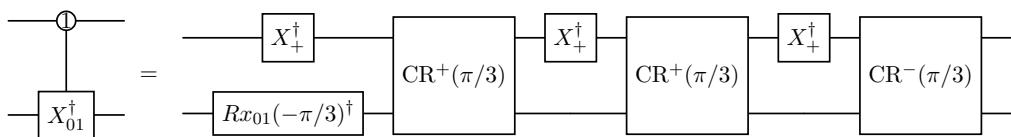


図 4.1.5: $\text{CNOT}_{\text{trit}}^\dagger$ のゲートシーケンス

計が実装可能となる。

次に中央の $(q_{n-1}, q_n)\text{CNOT}_{\text{trit}}$ のベリー位相を解消する手法を説明する。このゲートはゲートシーケンスで一度しか使わなかったため、エルミート共役の行列とセットで実装する、という先述の方法は使えない。そこで、図 4.1.2 の n 制御ビットゲートの q_n は、ゲートシーケンス中に常に量子ビットとして扱われていることに注目する。 q_n には $(q_{n-1}, q_n)\text{CNOT}_{\text{trit}}$ によって、 I か X_{01} の操作が加わるだけなので、 $|0\rangle$ と $|1\rangle$ しか使わない。よってこのシーケンスで q_n は量子ビットとして扱われており、 $|2\rangle$ は使わないため、 q_n の $|2\rangle$ にはベリー位相がかかっても出力に関係ない。

そこで (q_{n-1}, q_n) CNOT_{trit} のベリー位相を q_n の $|2\rangle$ に押し付ける、という手法をとる。まず q_n の $|0\rangle$ と $|1\rangle$ の位相のみを整えるため、 $Rz_{12}(-\frac{\pi}{3})$ と $Rz_{01}(\frac{\pi}{3})$ を CNOT_{trit} の制御トリットに適用すると、次のような計算結果となる：

$$(q_{n-1}, q_n)\text{CNOT}_{\text{trit}} \left(Rz_{12}\left(-\frac{\pi}{3}\right) \otimes I \right) \left(Rz_{01}\left(\frac{\pi}{3}\right) \otimes I \right) \quad (4.1.19)$$

$$= (q_{n-1}, q_n)\text{CNOT}_{\text{trit}} \left(\begin{bmatrix} 1 & & \\ & e^{i\frac{\pi}{6}} & \\ & & e^{-i\frac{\pi}{6}} \end{bmatrix} \otimes \begin{bmatrix} 1 & & \\ & 1 & \\ & & 1 \end{bmatrix} \right) \left(\begin{bmatrix} e^{-i\frac{\pi}{6}} & & \\ & e^{i\frac{\pi}{6}} & \\ & & 1 \end{bmatrix} \otimes \begin{bmatrix} 1 & & \\ & 1 & \\ & & 1 \end{bmatrix} \right) \quad (4.1.20)$$

$$= (q_{n-1}, q_n)\text{CNOT}_{\text{trit}} \left(\begin{bmatrix} e^{-i\frac{\pi}{6}} & & \\ & e^{i\frac{\pi}{3}} & \\ & & e^{-i\frac{\pi}{6}} \end{bmatrix} \otimes \begin{bmatrix} 1 & & \\ & 1 & \\ & & 1 \end{bmatrix} \right) \quad (4.1.21)$$

$$\sim \begin{bmatrix} 1 & 0 & 0 & & & \\ 0 & 1 & 0 & 0 & 0 & \\ 0 & 0 & 1 & & & \\ & & & 0 & 1 & 0 \\ & & & 0 & 0 & 0 \\ & & & 0 & 0 & i \\ & & & & 1 & 0 & 0 \\ & & & & 0 & 1 & 0 \\ & & & & 0 & 0 & 1 \end{bmatrix} \quad (4.1.22)$$

この行列表現から Rz_{01} と Rz_{12} を作用させた CNOT_{trit} は「制御トリットが $|1\rangle$ のときの標的トリットの $|2\rangle$ 」にのみベリー位相がつく、と読み取れる。したがってベリー位相が標的トリットである q_n の $|2\rangle$ に押し付けられている。

ここで標的トリット q_n が量子ビットとして扱われていることから、式 (4.1.19) の行列のうち q_n の $|2\rangle$ が関与する行と列は使わないため、 (q_{n-1}, q_n) CNOT_{trit} の行列表現は、次のような (6×6) 行列とみなしてよい：

$$\begin{bmatrix} 1 & 0 & 0 & & & \\ 0 & 1 & 0 & 0 & 0 & \\ 0 & 0 & 1 & & & \\ & & & 0 & 1 & 0 \\ & & & 0 & 0 & 0 \\ & & & 0 & 0 & i \\ & & & & 1 & 0 & 0 \\ & & & & 0 & 1 & 0 \\ & & & & 0 & 0 & 1 \end{bmatrix} \rightarrow \begin{bmatrix} 1 & 0 & & & & \\ 0 & 1 & & & & \\ & & 0 & 1 & & \\ & & & 0 & 1 & \\ & & & & 1 & 0 & \\ & & & & & 0 & 1 \end{bmatrix} \quad (4.1.23)$$

この (6×6) 行列にはベリー位相がないことから、量子トリット n 制御ビットゲートにおける (q_{n-1}, q_n) CNOT_{trit}

のベリー位相は解消できている。

このCNOT_{trit}のための $Rz_{12}(-\frac{\pi}{3})$ と $Rz_{01}(\frac{\pi}{3})$ は、2.2.4節で紹介したVirtual Z Gateの技術を量子トリットに応用すれば実装ができる。式(2.2.20)で $X(=Rx)$ のパルス位相を ϕ_0 変えると、 $Rz(\phi_0)$ が実装できることを説明した。これを量子トリットに拡張して考えると、 X_{01} のパルス位相を ϕ_0 変えると $Rz_{01}(\phi_0)$ が実装でき、 X_{12} のパルス位相を ϕ_1 変えると $Rz_{12}(\phi_1)$ が実装できる。図2.2.2を見ると、制御トリットには $X_+ = X_{01}X_{12}$ が実装されているため、この中の一番右の X_+ を利用して、 X_{01} はパルス位相を $\frac{\pi}{3}$ 、 X_{12} はパルス位相を $-\frac{\pi}{3}$ だけ変更して実装すれば、 (q_{n-1}, q_n) CNOT_{trit}のベリー位相は解消できる。

以上より、 O^\dagger とVirtual Z Gateのベリー位相対策設計を組み込んで、図4.1.6のようなゲートシーケンスに変更すれば、 n 制御ビットゲートの出力にベリー位相が生じない。

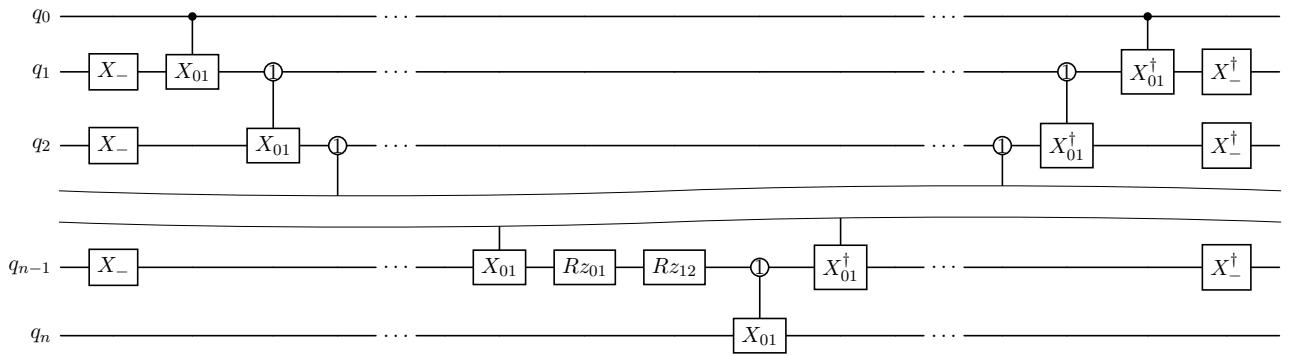


図4.1.6: ベリー位相対策済みの量子トリット n 制御ビットゲートシーケンス。

4.2 ZZ相互作用への設計

ここでは3章で説明したZZ相互作用という効果を解消するための設計を図4.1.6の n 制御ビットゲートシーケンスに組み込む方法説明する。4.2.1節ではZZ相互作用を解消する原理について説明し、4.2.2節ではZZ相互作用を解消する設計を量子トリット n 制御ビットゲートシーケンスへ組み込む方法を説明する。

4.2.1 ZZ相互作用を解消する原理

ZZ相互作用は量子ビット周波数や結合定数等、ハードウェアパラメータに依存している。ハードウェアパラメータは時間が経つと数値が変化するため、ZZ相互作用も時間が経つと影響の大きさが変化する。したがってZZ相互作用がどのような値であっても、解消できるような設計が必要である。

動的デカップリングは、静的にかかり続けるZZ相互作用を自動的にキャンセルできる技術であり、 H_{stat} に含まれるZZ相互作用の値に依らず機能する。従来から動的デカップリングは量子ビット間のZZ相互作用に使用されており[49]、この考え方を参考に量子ビット-量子トリット間のZZ相互作用の解消手段として流用した。

一方で量子トリット-量子トリット間には、従来の動的デカップリングの考え方を使えなかったため、ZZ 相互作用を解消する全く新しい方法を考案した。こちらも H_{stat} に含まれる ZZ 相互作用の値に依らず、自動的に ZZ 相互作用によるエラーを解消できるものとなっている。

以下で具体的な ZZ 相互作用の解消原理を説明していく。

量子ビット-量子トリット間

量子ビット-量子トリット間の ZZ 相互作用の解消原理について説明する。まず量子ビット (q_0) と量子トリット (q_1) 間に ZZ 相互作用が

$$\tilde{H}_{\text{stat}} = \nu_{z3} (\sigma_z \otimes \lambda_3) + \nu_{z8} (\sigma_z \otimes \lambda_8) \quad (4.2.1)$$

とかかり続けているモデルを考える。ここで σ_z は (2×2) パウリ Z 行列、 λ_3 と λ_8 は (3×3) ゲルマン行列、 ν_{z3}, ν_{z8} は各生成子の周波数である。このとき図 4.2.1 のように、 q_0 に時間 α 毎に $2n$ 回 X ゲートを適用することを考える(動的デカップリング)。X ゲートで挟まれた時間発展ハミルトニアンは次のように与えられる：

$$(\sigma_x \otimes I) \tilde{H}_{\text{stat}} (\sigma_x \otimes I) = \nu_{z3} (\sigma_x \sigma_z \sigma_x \otimes \lambda_3) + \nu_{z8} (\sigma_x \sigma_z \sigma_x \otimes \lambda_8) \quad (4.2.2)$$

$$= \nu_{z3} (-\sigma_z \otimes \lambda_3) + \nu_{z8} (-\sigma_z \otimes \lambda_8) \quad (\because \{\sigma_x, \sigma_z\} = 0) \quad (4.2.3)$$

$$= -\tilde{H}_{\text{stat}} \quad (4.2.4)$$

したがって図 4.2.1 のシーケンスでは、 H_{stat} で時間発展した分の効果が、X ゲートで挟まれた領域の $-H_{\text{stat}}$ による時間発展の効果によって自動的にキャンセルされている。また同時にこの計算は ν_{zz} と $\nu_{z\zeta}$ の値に関係なく成立することから、ZZ 相互作用の効果の大きさに依らず機能することも確認できる。

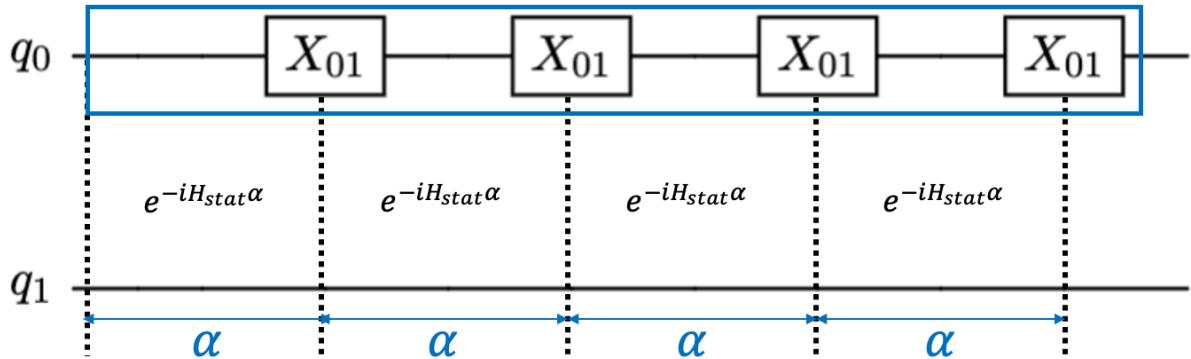


図 4.2.1: 量子ビット-量子トリットで ZZ 相互作用を解消するゲートシーケンス。等間隔に X_{01} を $2n$ 回実装することで、時間発展 $e^{-iH_{\text{stat}}4\alpha}$ を分割する。

量子トリット-量子トリット間

量子トリット-量子トリット間の ZZ 相互作用の解消原理について説明する。まず量子トリット (q_0) と量子トリット (q_1) 間に ZZ 相互作用が

$$\tilde{H}_{\text{stat}} = \nu_{33} (\lambda_3 \otimes \lambda_3) + \nu_{38} (\lambda_3 \otimes \lambda_8) + \nu_{83} (\lambda_8 \otimes \lambda_3) + \nu_{88} (\lambda_8 \otimes \lambda_8) \quad (4.2.5)$$

とかかっているモデルを考える。ここで図 4.2.2 のように、 q_0 に時間 α 毎に $3n$ 回 X_+ ゲートを適用することを考えると、時間発展は次のように与えられる：

$$(X_+ \otimes I)e^{-i(\tilde{H}_{\text{stat}})\alpha}(X_+ \otimes I)e^{-i(\tilde{H}_{\text{stat}})\alpha}(X_+ \otimes I)e^{-i(\tilde{H}_{\text{stat}})\alpha} \sim I \otimes I \quad (4.2.6)$$

具体的な計算は付録 A.2 に記すが、計算結果はアイデンティティとなるため、このシーケンスによって ZZ 相互作用が解消できている。またこの計算結果は ν_{33} 等の周波数の値に関係なく成立することから、ZZ 相互作用の効果の大きさに依らず機能することが確認できる。

こちらの手法は、適用方法は量子ビット-量子トリット間の ZZ 相互作用解消に用いた動的デカップリングに似ているが、動的デカップリングは時間発展を自動でキャンセルするという手法であるのに対し、この量子トリット-量子トリット間の ZZ 相互作用解消の手法は全ての量子状態に均等に ZZ 相互作用による位相エラーを分配することにより、位相エラーを絶対位相として無視する、という手法であるため、原理は全く別である。

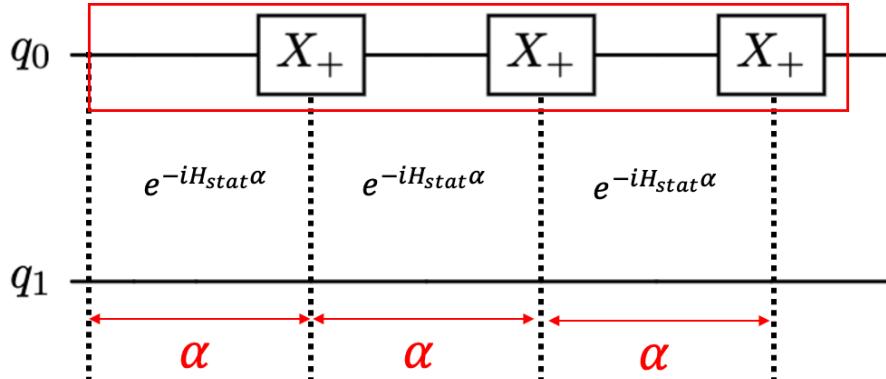


図 4.2.2: 量子トリット-量子トリットで ZZ 相互作用を解消するゲートシーケンス。等間隔に X_+ を $3n$ 回実装することで、時間発展 $e^{-iH_{\text{stat}}3\alpha}$ を分割する。

4.2.2 ゲートシーケンスへの組み込み方

ここでは、前節で説明した各 ZZ 相互作用を解消する設計を n 制御ビットゲートに組み込む方法を説明する。

まず注意する点として、CNOT_{bit}(図 2.2.1) の制御ビットには等間隔で X_{01} が 2 回適用されていて、CNOT_{trit}(図 2.2.2) の制御ビットには等間隔で X_+ が 3 回適用されていることから、CNOT の制御ビットには前節で説明した ZZ 対策設計と同じ構造が既に存在する。つまり既にこの n 制御ビットゲートの

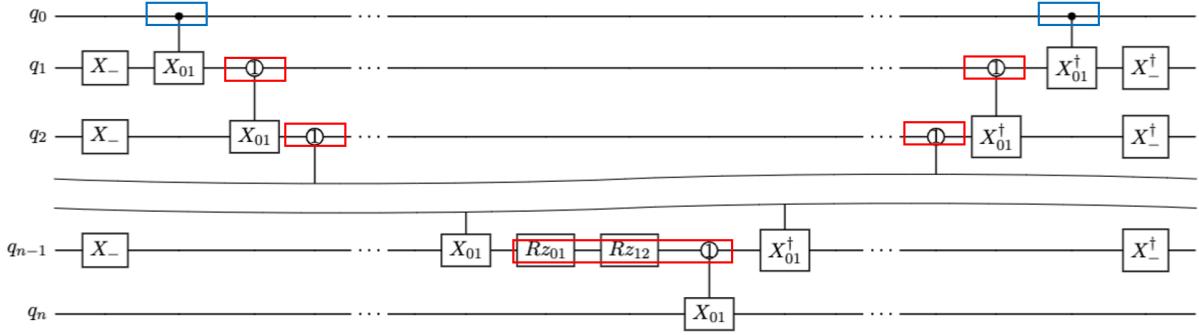


図 4.2.3: ZZ 対策設計が既に存在している領域を示した図。赤は量子トリット-量子トリット間、青は量子ビット-量子トリット間の対策設計である。

シーケンスによって、図 4.2.3 により一部の ZZ 相互作用は対策ができていたことになる。また前節で説明した ZZ 相互作用の対策設計は、結合しているトランズモンのうちの片方にゲートを追加するというものであった。したがって設計を適用するトランズモンが重ならないようにするために、トランズモン一つおきに ZZ 対策設計を実装する必要がある。つまり全てのトランズモン結合に生じる ZZ 相互作用を解消するためには、図 4.2.3 で既にある対策設計の位置を基準に、一つ飛ばしのトランズモンに ZZ 対策設計を追加すればよい。

以上を踏まえて、図 4.2.4 により ZZ 相互作用の対策設計を実装すれば、ベリー位相に加え、ZZ 相互作用も対策できている n 制御ビットゲートのシーケンスとなっている。

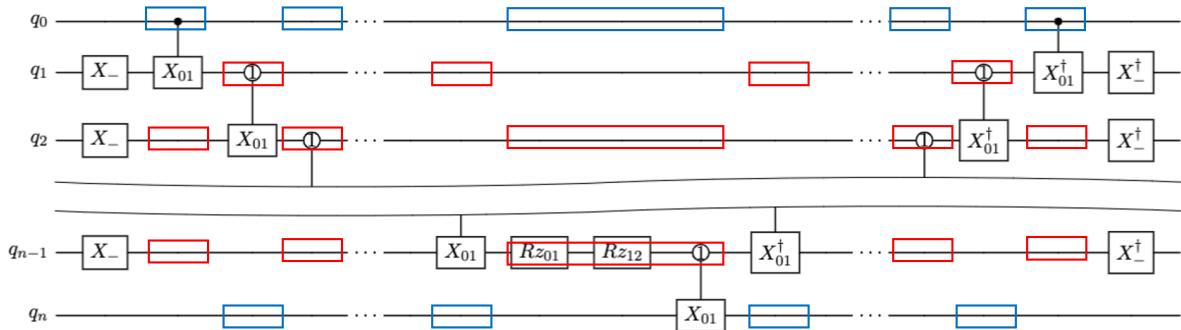


図 4.2.4: 全ての結合の ZZ 相互作用を解消するために、ZZ 対策設計を適用する領域を示した図。赤は量子トリット-量子トリット間、青は量子ビット-量子トリット間の対策設計である。図 4.2.3 で既に設計があつた領域を基準に、一つ飛ばしのトランズモンに ZZ 対策設計を追加した。

4.3 電荷分散への設計

この節では、3 章で説明した電荷分散による位相エラーを解消する手法と設計について説明する。

n 制御ビットゲートのシーケンスのうち、量子トリットとして使用するのは、図 4.1.6 の q_i ($1 \leq i \leq n-1$) である。この全ての量子トリットに対して、参考文献 [27] の手法を参考に、電荷分散による位相エラーを解消する設計を本研究で新たに考案した。

4.3.1 電荷分散を解消する手法

文献 [27] では Toffoli ゲートを実装する際、使用する 3 個のトランズモンのうち、中央のトランズモンのみ量子トリットとして使用しており、電荷分散による位相エラーに対処するために、動的デカッピング [49] を使用して「意図的にシーケンス中に量子トリットに delay を入れる」という設計を行っている。この設計を参考にして、 n 制御ビットゲートに拡張した場合、各量子トリットにどれだけの delay が必要かを算出した。以下では delay を入れることで電荷分散が解消できる理由を簡単に紹介した後、各量子トリットに必要な delay の算出結果を記す。

delay を入れる理由

電荷分散の影響は、3 章で述べたように、 $|1\rangle - |2\rangle$ 間の遷移周波数が短時間で変動する現象であり、その結果、 X_{12} は図 3.2.4 に示したように常に非共鳴の効果を引き起こしてしまう。最初のパルスを照射してから時間 t 後に、 $|1\rangle - |2\rangle$ 間の遷移周波数から δ だけ離れたドライブ周波数で X_{12} を実装した場合、電荷分散の影響を踏まえた $X_{12}^{(t)}$ は式 (2.2.30) の ζ と式 (2.1.12) の λ_8 を使用して次のように与えられる：

$$X_{12}^{(t)} \equiv e^{i\delta t \lambda_8} X_{12} e^{-i\delta t \lambda_8} \quad (4.3.1)$$

$$= X_{12} e^{-i\delta t \zeta} \quad (4.3.2)$$

つまり、本来の X_{12} の機能に加えて位相 $e^{-i\delta t \zeta}$ がかかる、という効果であり、これが電荷分散による位相エラーとなっている。この位相 $e^{-i\delta t \zeta}$ は、 X_{\pm} に対して次のような関係がある：

$$X_{\pm} e^{i\alpha z_n} = e^{i\alpha z_{n\pm 1}} X_{\pm} \quad (4.3.3)$$

ここで $z_0 = z_3 = \text{diag}(1, -1, 0) = z$ 、 $z_1 = \text{diag}(0, 1, -1) = \zeta$ 、 $z_2 = z_{-1} = \text{diag}(-1, 0, 1) = \lambda$ とした。

ここで文献 [27] にあった手法と同じように、図 4.1.6 のゲートシーケンスの q_0, q_1, \dots, q_n の一番右側にある X_-^{\dagger} を、遅延 τ だけ離した二つの X_- (以降、 $X_-^{(\tau)} X_-$ と記す) に置換してみる。 X_- の定義から、 $X_- X_- = X_-^{\dagger}$ であるため、この置換により n 制御ビットゲートの行列表現を変化させずに、遅延による位相のみを変化させることができる。

図 4.1.6 のゲートシーケンスの最後のゲートを X_-^{\dagger} のままにした場合、電荷分散による位相は、式 (4.3.1) と n 制御ビットゲート実装開始から X_-^{\dagger} を実装するまでの時間 D_{mcx} を用いて、次のように算出できる：

$$X_-^{\dagger(D_{\text{mcx}})} = X_{12}^{\dagger(D_{\text{mcx}})} X_{01}^{\dagger} = e^{i\delta D_{\text{mcx}} \zeta} X_-^{\dagger} \quad (4.3.4)$$

一方で図 4.1.6 のゲートシーケンスの最後のゲートを $X_-^{(\tau)} X_-$ に置換した場合、遅延 τ と D_{mcx} を用いて次のように算出できる：

$$X_-^{(D_{\text{mcx}}+\tau)} X_-^{(D_{\text{mcx}})} = e^{-i\delta(D_{\text{mcx}}+\tau)\zeta} X_- e^{-i\delta D_{\text{mcx}} \zeta} X_- \quad (4.3.5)$$

$$= e^{-i\delta(D_{\text{mcx}}+\tau)\zeta} e^{-i\delta D_{\text{mcx}} z} (X_-)^2 \quad (\because \text{式 (4.3.3)}) \quad (4.3.6)$$

$$= e^{-i\delta[D_{\text{mcx}}(\zeta+z) + \tau\zeta]} X_-^{\dagger} \quad (\because X_-^2 = X_-^{\dagger}) \quad (4.3.7)$$

ここで注目すべきは、 D_{mcx} はゲートシーケンスによって固定されたパラメータだが、 τ はこちらで自由に選べるパラメータであることである。そのため式 (4.3.4) の位相 $e^{-i\delta D_{\text{mcx}} \zeta}$ は完全に固定された位相

エラーだが、式(4.3.5)の位相 $e^{-i\delta[D_{\text{mcx}}(\zeta+z) + \tau\zeta]}$ は $e^{-i\delta\tau\zeta}$ だけ自由度のある位相エラーである。この $e^{-i\delta\tau\zeta}$ を適切な τ で利用すれば、位相 $e^{-i\delta D_{\text{mcx}}(\zeta+z)}$ を解消できることが文献[27]で示されている。

これ以降、 $e^{-i\delta D_{\text{mcx}}(\zeta+z)}$ のような n 制御ビットゲート (D_{mcx}) 中に蓄積する位相を単に「位相エラー」と呼び、 $e^{-i\delta\tau\zeta}$ のような遅延 (τ) 中に蓄積する位相を「補正用位相」と呼んで区別する。

式(4.3.4)で算出した位相エラーは D_{mcx} を用いて $e^{-i\delta D_{\text{mcx}}\zeta}$ と表したが、実際はゲートシーケンス中に含まれる各ゲートと電荷分散による位相エラーとの交換関係により、式(4.3.5)にある通り、 $e^{-i(\theta_\zeta\zeta + \theta_z z)}$ という形になる。この θ_ζ 、 θ_z の算出と、それに対する適切な τ を求める計算は非常に複雑であるため、具体的な計算は付録A.3に記し、ここでは量子トリット n 制御ビットゲートの q_0, q_1, \dots, q_n に対し、適切な補正用位相を入れるための遅延 τ の計算結果のみを記す。 (q_i, q_j) CNOT_{trit} のゲート実装時間を $d(q_i, q_j)$ 、 q_i に必要な遅延を τ_i と表記すると、結果は次のようにになった：

$$\begin{aligned} n = 2m+1 \text{ のとき } & \begin{cases} \tau_{2j+1} (0 \leq j < \frac{n-1}{2}) = 2 \left(\sum_{i=1}^{m-1} d(q_{2i}, q_{2i+1}) \right) + d(q_{2m}, q_{2m+1}) \\ \tau_{2j} (0 < j < \frac{n-1}{2}) = 2 \left(\sum_{i=1}^{m-1} d(q_{2i+1}, q_{2i+2}) \right) \end{cases} \\ n = 2m \text{ のとき } & \begin{cases} \tau_{2j+1} (0 \leq j < \frac{n}{2}) = 2 \left(\sum_{i=1}^{m-1} d(q_{2i}, q_{2i+1}) \right) \\ \tau_{2j} (0 < j < \frac{n}{2}) = 2 \left(\sum_{i=1}^{m-1} d(q_{2i+1}, q_{2i+2}) \right) + d(q_{2m-1}, q_{2m}) \end{cases} \end{aligned} \quad (4.3.8)$$

n の偶奇に依存して、各量子トリットに必要な τ_i は異なることが確認できた。また必要な遅延 τ_i の値は、奇数番目の量子トリット用と偶数番目の量子トリット用の2種類のみで統一されていることも確認できた。

4.3.2 ゲートシーケンスへの組み込み方

この節では図4.1.6に対し、前節で算出した遅延 τ_i だけ離した $X_-^{(\tau_i)} X_-$ の設計を組み込む方法を説明する。

参考文献[27]で使用されている電荷分散対策は、Toffoliでは量子トリットを一つしか使用しないことから、図4.3.1のように q_1 にのみ遅延 τ を入れており、この場合は隣接する q_0, q_2 に ZZ 相互作用対策を組み込めば、同時に遅延中の ZZ 相互作用が解消できた。しかしこの設計を量子トリット n 制御ビットゲートに拡張する際に、ZZ 相互作用の対策設計と電荷分散の対策設計が両立できないことが判明したため、両立ができない理由と、両立するための新しい設計を説明する。

ZZ 相互作用対策と電荷分散対策の衝突

図4.3.1のような Toffoli の電荷分散対策を n 制御ビットゲートに拡張することを考えたとき、図4.3.2のように、全ての量子トリットに遅延設計 $X_-^{(\tau_i)} X_-$ を入れ、その遅延の間に加わる ZZ 相互作用に対しても、ZZ 対策設計によって同時に解消する、というのが自然である。

しかし量子トリット-量子トリット間の ZZ 対策設計を図4.3.2のように遅延中の量子トリットに組み込むと、補正用位相が蓄積しないことが以下の計算から判明した。図4.2.2のように X_+ を等間隔で3回

実装した場合、補正用位相は次のようになる：

$$X_+^{(3\alpha)} X_+^{(2\alpha)} X_+^{(\alpha)} = X_+ e^{-i\zeta(3\alpha)} X_+ e^{-i\zeta(2\alpha)} X_+ e^{-i\zeta(\alpha)} \quad (\because \text{式 (4.3.1)}) \quad (4.3.9)$$

$$= e^{-i\lambda(3\alpha)} e^{-iz(2\alpha)} e^{-i\zeta(\alpha)} X_+ X_+ X_+ \quad (\because \text{式 (4.3.3)}) \quad (4.3.10)$$

$$= e^{iI\alpha} (X_+)^3 \quad (4.3.11)$$

$$= I \quad (4.3.12)$$

2段目から3段目の式変形では、 $(3\lambda + 2z + \zeta) = \text{diag}(-1, -1, -1) = -I$ という関係を用いた。この結果から X_+ を等間隔で $3n$ 回実装しているとき、補正用位相が蓄積せず、遅延が意味を成さないことがわかった。

そこで ZZ 相互作用対策と電荷分散対策が両立できるように、新たな設計（図 4.3.3）を考案した。これは、奇数番目の量子トリットの遅延 τ_{2j+1} と偶数番目の量子トリットの遅延 τ_{2j} を別々に入れる設計となっている。各量子トリットに入る遅延の時間については、本研究の計算結果（式 (4.3.8)）から判明した通り、 τ_{2j+1} と τ_{2j} の2種類であるため、以降は $\tau_{2j+1} = \tau_1$ 、 $\tau_{2j} = \tau_2$ と表記する。 τ_1 中は、偶数番目の量子トリットに ZZ 対策設計を組み込むことで、奇数番目の量子トリットの遅延では補正用位相が蓄積されており、遅延が電荷分散対策として機能している。また ZZ 相互作用の対策は 4.2.2 節で説明したように、一つおきのトランズモンに実装すればよいので、図 4.3.3 の設計では遅延 τ_1 中の全ての結合間の ZZ 相互作用も同時に解消できている。しかし τ_1 中は、偶数番目の量子トリットに対して補正用位相は ZZ 対策設計により解消されてしまっているため、偶数番目の量子トリットに対しては遅延の意味がない。そこで遅延 τ_1 で奇数番目の量子トリットの電荷分散を解消し終わった後、偶奇を入れ替えて同様の設計を行った。すると先程と対称的に全ての結合の ZZ 相互作用を解消しながら、偶数番目の量子トリットに補正用位相を蓄積させることができるために、電荷分散の対策設計が機能している。

この図 4.3.3 の設計は計算上は ZZ 相互作用と電荷分散の位相エラーを完全に解消できる設計となっているが、ゲートシーケンスの時間が長くなるという問題がある。 $\tau_1 + \tau_2$ は式 (4.3.8) の形から算出すると、 n 制御ビットゲートシーケンスに含まれる全ての CNOT の合計時間となるため、電荷分散対策ありとした場合のゲートシーケンス実装時間は、対策なしに比べて 2 倍弱となる。これにより、実機ではデコヒーレンスの影響が大きくなる。この電荷分散対策による実機でのデコヒーレンスの影響は、5.2.2 節の IBMQuantum を使用した実験結果の考察で触れることにする。

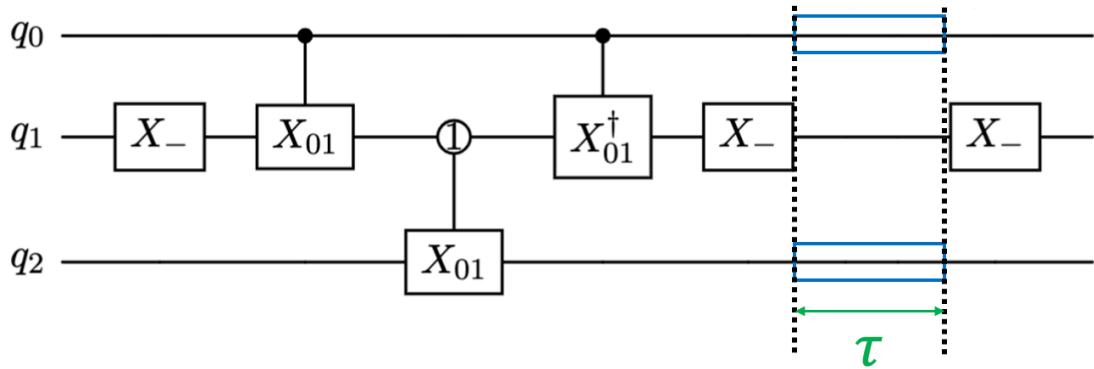


図 4.3.1: Toffoli の電荷分散への対策設計の入れ方。 $X_-^{(\tau)}X_-$ の設計を、量子トリットとして使用する q_1 に組み込んで、隣接する q_0, q_2 に ZZ 相互作用への対策設計を実装すれば、 τ 中の ZZ 相互作用を解消できた。[27]

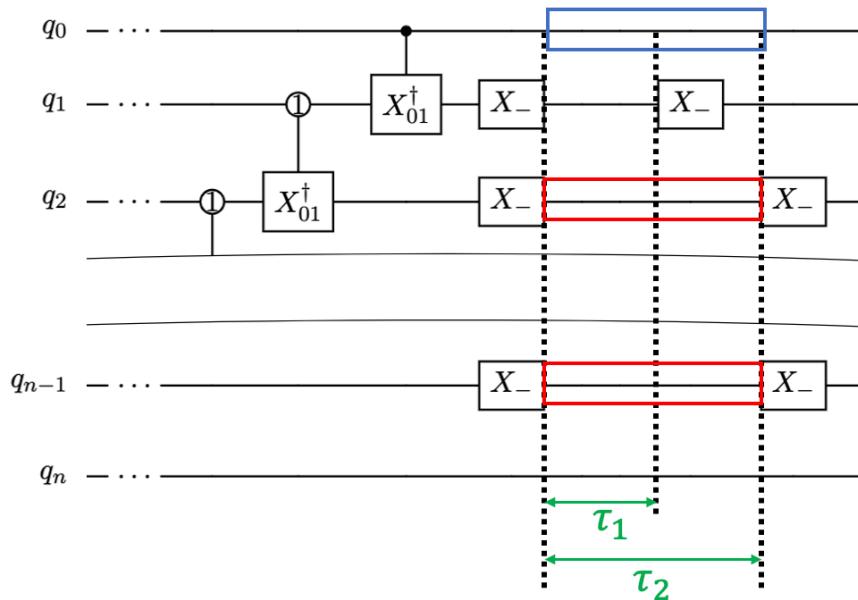


図 4.3.2: n 制御ビットゲートで使用する量子トリット全てに、Toffoli と同様に電荷分散のための遅延を $X_-^{(\tau_i)}X_-$ に入れた場合のゲートシーケンス。図 4.1.6 のうち、最後のシーケンスのみを表示している。式 (4.3.8) の結果から、 q_i の遅延は i が奇数のときは τ_1 と偶数のときは τ_2 である。

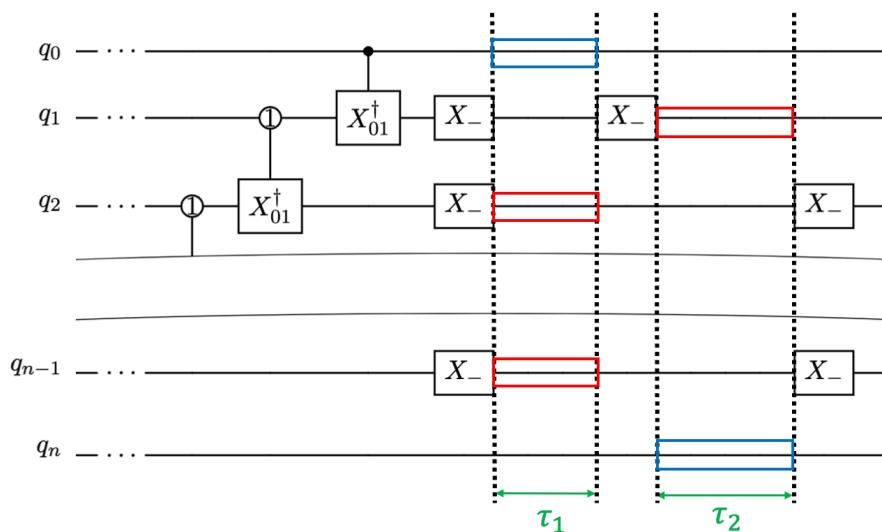


図 4.3.3: n 制御ビットゲートで使用する量子トリットの中で、奇数番目の量子トリットと偶数番目の量子トリットで分割して遅延 τ_1 と τ_2 を入れているシーケンス。図 4.1.6 のうち、最後のシーケンスのみを表示している。遅延を入れていない量子トリット（ビット）を利用して、遅延中の ZZ 相互作用対策も同時に実装できる。

第5章 シミュレーションと実機での実証

この章では、4章で考案したベリー位相、ZZ 相互作用、電荷分散への対策設計を GPU を用いた qudit-sim によるシミュレーションと実際の量子コンピュータである IBMQuantum を用いた量子計算で評価する。5.1 節ではシミュレーションによるベリー位相と ZZ 相互作用への対策設計の評価、5.2 節では IBMQuantum による ZZ 相互作用と電荷分散への対策設計の評価について、評価方法と結果を説明する。

5.1 qudit-sim での評価

5.1.1 qudit-sim の説明

シミュレーションには、qudit-sim[35] というシミュレータを用いる。qudit-sim は、量子回路を入力し、行列計算によって終状態の確率分布を求めるような一般的な量子コンピュータのシミュレータとは異なり、ハードウェアとパルスのパラメータを入力することで、トランズモンの時間発展を計算し、終状態を行列で出力するシミュレータとなっている。

qudit-sim の具体的な使用手順は次のようにになっている：

1. ハードウェアパラメータを定義

(Qudit の次元 d 、トランズモン数 n 、量子ビット周波数 ω_0 、非調和性 δ 、結合定数 J)

2. パルススケジュールの定義

(各パルスのドライブ開始時刻と終了時刻、ドライブ周波数、パルス振幅、パルス位相)

3. n Qudit 系で時間発展シミュレーション

4. 終状態を $(d^n \times d^n)$ 行列として算出

3 のシミュレーションにおける時間発展ハミルトニアンは、2.2.2 節で説明した $H_{\text{stat}} + H_d$ というモデルであり、1 で静的ハミルトニアン H_{stat} 、2 でドライブハミルトニアン H_d が定まっている。

4 で得られる $(d^n \times d^n)$ 行列は、量子状態トモグラフィにより得られる情報と同等である。この得られた行列を σ とおき、理想的な行列 ρ を用意すれば、式 (2.1.30) から同値変形で得られる $F = \text{Tr}[\rho\sigma]$ という行列のトレース計算で忠実度が算出できる。また qudit-sim では時間発展シミュレーションの際に、途中の状態を保持しておくことができ、それらを利用して各パルスがどのような有効ハミルトニアンによる時間発展を起こしていたかを算出できる機能がある。これにより忠実度の数値のみによる判断だけではなく、有効ハミルトニアンにどのような効果がエラーとして残っているかまで、考察できる。

評価に qudit-sim と IBMQuantum を併用する理由

本研究で実機のみではなく、シミュレータも併用して評価しているのにはいくつか理由がある。

1つ目は解析時間の短さである。実機では、測定をすると量子状態が変化してしまうため、時間発展の途中で信号は取り出せず、一度の実験で複数の状態を知ることはできない。したがって過渡現象を解析するためには、実験中の状態を知りたい時点で QPT を行い、最初から実験をやり直し、次の知りたい時点で QPT を行い…と繰り返す必要がある。[2.1.3 節](#)や[5.2.1 節](#)で説明する通り、QPT は測定回数や計算量が大きいことに加え、IBMQuantum は占有して使えるシステムではないため、一度の実験につき待ち時間が 7 時間ほどある。この工程は対策設計の評価だけでなく、試行錯誤による開発の段階でもあったため、IBMQuantum により膨大な試行回数を行うと時間がかかるてしまう。qudit-sim は一度のシミュレーションで途中の量子状態を複数得られ、有効ハミルトニアンが短時間で解析できるため、開発段階では使いやすい。

2つ目は量子トリット空間で解析が可能な点である。IBMQuantum の測定は量子ビット用のものになっており、量子状態トモグラフィを行う場合、量子ビット空間での同定となっている。qudit-sim では、指定した d 次元 Qudit に対し、終状態を $(nd \times nd)$ 行列で出力するため、量子トリット空間での状態を得ることができる。これは d 次元 Qudit の量子状態トモグラフィで得られる情報と同等であり、量子トリット空間での忠実度を算出できることに加えて、有効ハミルトニアンの解析によって量子トリット空間におけるエラーを特定できる。したがって本研究で考案した新設計が $|2\rangle$ まで制御できているかを確認できるので、本研究の評価方法として適切である。

5.1.2 評価手法

本研究の新規性である ZZ 相互作用と電荷分散の対策設計を利用するには、制御ビットが 3 個以上の多制御ビットゲートである。また本研究で開発した 3 つの対策設計は、4.2.2 章で説明した通り、図 4.1.3 や図 4.2.4、図 4.3.3 のように同じ設計を反復利用したり delay を変えたりするだけで、 n 制御ビットゲートに拡張可能であった。このとき制御ビットが 3 個以上増えても対策設計の原理は変わらないので、対策設計の評価には最小系である図 5.1.1 のような $n = 3$ の多制御ビットゲート（以降 CCCX と記す）を実装すれば十分だと考えた。したがって qudit-sim で 4 トランズモン系の CCCX を作成して、対策設計の評価を行う。

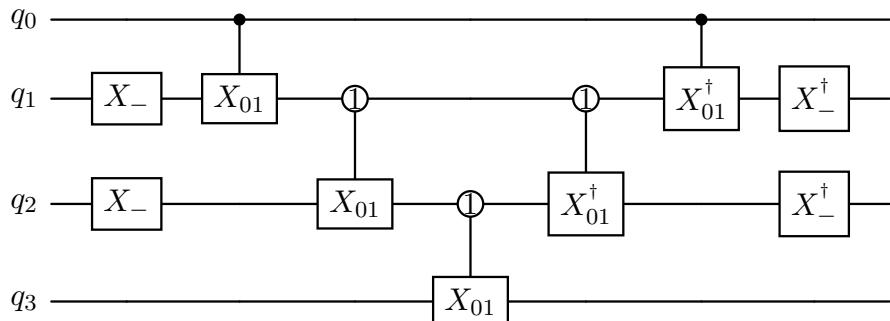


図 5.1.1: 量子トリット CCCX のゲートシーケンス

ベリー位相

4.2.2 章で説明した通り、CCCX シーケンス全体でベリー位相を解消するために、 $(q_0, q_1)\text{CNOT}^\dagger$ と $(q_1, q_2)\text{CNOT}^\dagger$ と Virtual Z Gate 付きの $(q_2, q_3)\text{CNOT}$ を実装する必要がある。そこで図 5.1.2 の示すように、まずは(i)-(v) のゲートについて、それぞれ (i) は図 2.2.1、(ii) は図 2.2.2、(iii) は図 2.2.2 と Virtual Z Gate を導入したもの、(iv) は図 4.1.5、(v) は図 4.1.4 のゲートシーケンスを個別で実装して、各々のゲートの忠実度を計測する。

そして (i) と (v)、(ii) と (iv) の忠実度を比較して、本研究で開発した図 4.1.4、4.1.5 のダガー付きのシーケンスが元々開発されていた図 2.2.1、2.2.2 のダガーなしシーケンスと同精度で機能することを確認する。また、(iii) は Virtual Z Gate を適用する前と後の忠実度を比較して、この手法が元々の $\text{CNOT}_{\text{trit}}$ の忠実度を下げずに導入できることを確認する。これらの結果を用いて、図 4.1.2 から本研究で開発したベリー位相対策を導入した図 4.1.6 に変更する際に、忠実度が下がっていないことを示す。

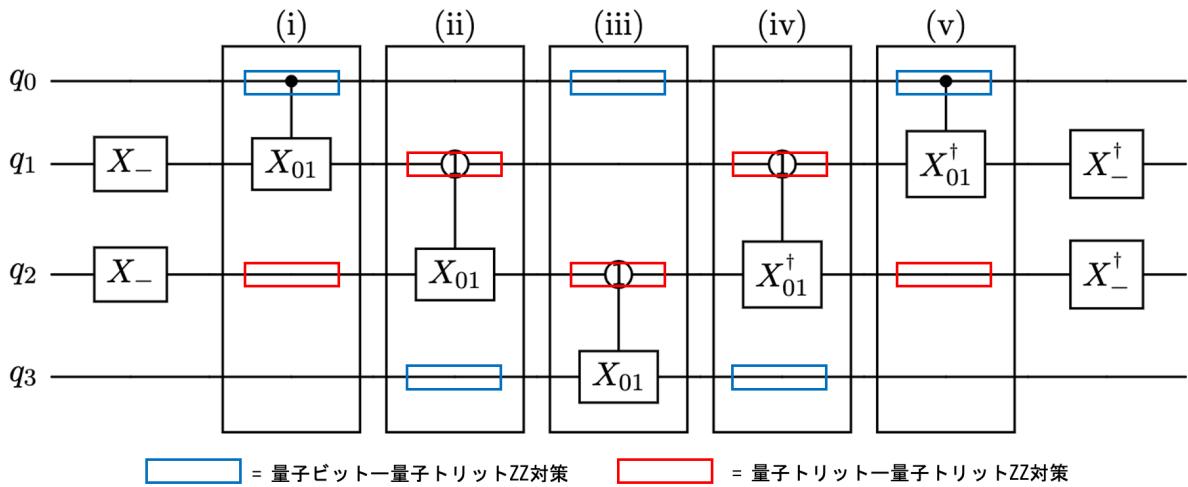


図 5.1.2: 忠実度測定のためにゲート毎に分割して行う実験の番号付け (i)-(v)。それぞれの実験は q_0, q_1, q_2, q_3 系で行うため、CNOT が作用しないトランズモンも ZZ 相互作用による位相エラーを受け続ける。

ZZ 相互作用

図 5.1.1 にある全ての CNOT に対して、ZZ 相互作用への対策設計を図 5.1.2 によく適用する。ZZ 相互作用への対策設計がある場合とない場合で (i)-(v) の各ゲートと CCCX 全体の忠実度を測定し、精度が向上しているか確認する。

5.1.3 実験

本研究において qudit-sim で使用するハードパラメータは、ibmq_kolkata の $(q_{18}, q_{21}, q_{23}, q_{24})$ の 4 量子ビットの値を参照している。

ゲート作成

はじめに図 5.1.2 の (i)-(v) の CNOT を 2.2.5 節で紹介した方法に基づいて作成する。 X_{01} 、 X_{12} は、qudit-sim にある pi-pulse 関数によって作成できるので、ここでは CR パルスの作成工程を見せる。まずは (q_0, q_1) CNOT_{bit} に必要な交差共鳴ゲート CR_{bit}[±]($\pi/4$) を作成する。

CR_{bit}[±]($\pi/4$) を作成するために、 $ZX(\theta)$ の回転角が $\theta = \pm\pi/4$ である必要がある。このようなパルスを作成するために、ドライブ時間を IBMQuantum で実装されている CNOT の値を参考に 192[ns] と定めて、パルス振幅を掃引した。その結果、パルス振幅と CR_{bit}[±] の ZX 回転成分 θ の関係は、図 5.1.3 のようになつた。青プロットはパルス振幅を掃引して測定した点、青直線はそのプロットによる fitting 結果となっている。これにより $\theta = \pi/4$ であるパルス振幅が左図から、 $\theta = -\pi/4$ であるパルス振幅が右図から、fitting によって求まっている（赤プロット）。fitting によりパルス振幅が 0.813 で $\theta = \pi/4$ 、-0.811 で $\theta = -\pi/4$ になると予想できる。このパルス振幅で 4 トランズモン系にドライブしたパルス $e^{-i\tilde{H}_{CR}^{\pm}t}$

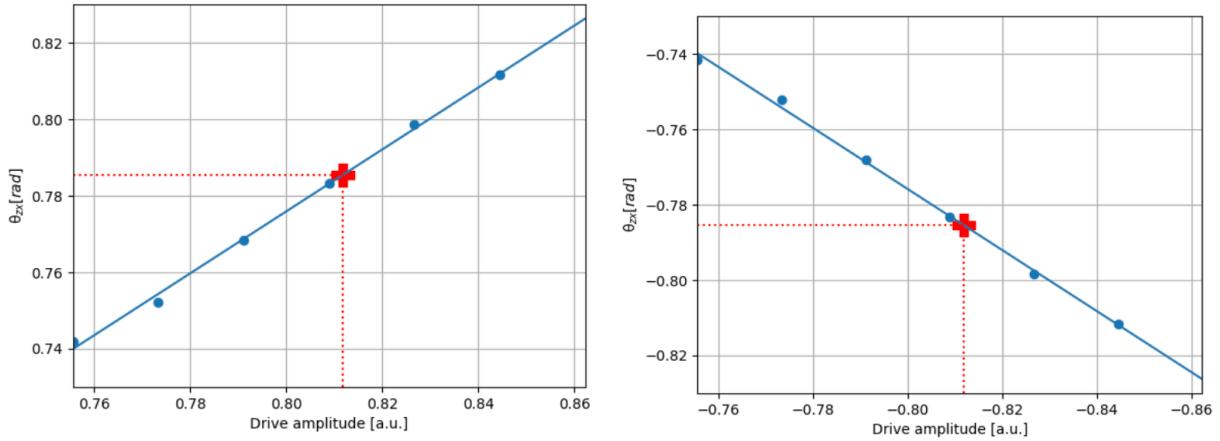


図 5.1.3: $ZX(\theta)$ におけるパルス振幅と θ の関係。横軸がパルス振幅、縦軸が θ である。青プロットはパルス振幅を掃引して測定した点、青直線はそのプロットによる fitting 結果となっている。これにより $\theta = \pi/4$ であるパルス振幅が左から、 $\theta = -\pi/4$ であるパルス振幅が右から、fitting により赤プロットで求まっている。

に対し、 $\tilde{H}_{CR}^{\pm}(= i \log[\text{CR}_{\text{bit}}^{\pm}])$ の構成要素を qudit-sim で解析すると、次のようになつた：

$$i \log[\text{CR}_{\text{bit}}^+] = 3.831\pi \frac{III}{2} + 0.146\pi \frac{IXII}{2} + 7.047\pi \frac{ZIII}{2} + 0.250\pi \frac{ZXII}{2} \quad (5.1.1)$$

$$i \log[\text{CR}_{\text{bit}}^-] = 3.970\pi \frac{III}{2} - 0.151\pi \frac{IXII}{2} + 7.180\pi \frac{ZIII}{2} - 0.250\pi \frac{ZXII}{2} \quad (5.1.2)$$

これは式 (2.2.25) の示されていた通りの結果であり、式 (2.2.28) と同じく $[Rz(\phi_0) \otimes Rx(\mp\phi_1)]$ 要素が存在し、 ZX 要素の回転角が $\pm\pi/4$ になっていることから、CR_{bit}[±]($\pi/4$) が作成できているとわかる。よつて X_{01} とこの CR_{bit}[±]($\pi/4$) を図 2.2.1 のように実装すれば、 (q_0, q_1) CNOT_{bit} が作成できる。

次に (q_1, q_2) CNOT_{trit} と (q_2, q_3) CNOT_{trit} に必要な交差共鳴ゲート CR_{trit}[±]($\pi/3$) を作成する。2.2.5 節で説明したように CR_{trit}[±] にはパルス A、B を較正して、式 (2.2.31) で $\nu_{zx} = -\nu_{\zeta x}$ となるような \tilde{H}_{CR}^{\pm} を構成すれば、CR_{trit} が作成できるのだった。CR_{trit}[±] の作成工程は土によらず同じなので、以降は CR_{trit}⁺ の作成過程のみを記していく。まずは $\nu_{zx} = -\nu_{\zeta x}$ となるパルス A の振幅を探すため、パルス A の振幅

を掃引して、それに対する \tilde{H}_{CR}^+ の ν_{zx} 、 $\nu_{\zeta x}$ 、 ν_{Ix} をプロットした。その結果、図 5.1.4 のようになった。下図の fitting による交点から、 (q_1, q_2) のパルス A の振幅は 0.637、 (q_2, q_3) のパルス A の振幅は 0.254 で

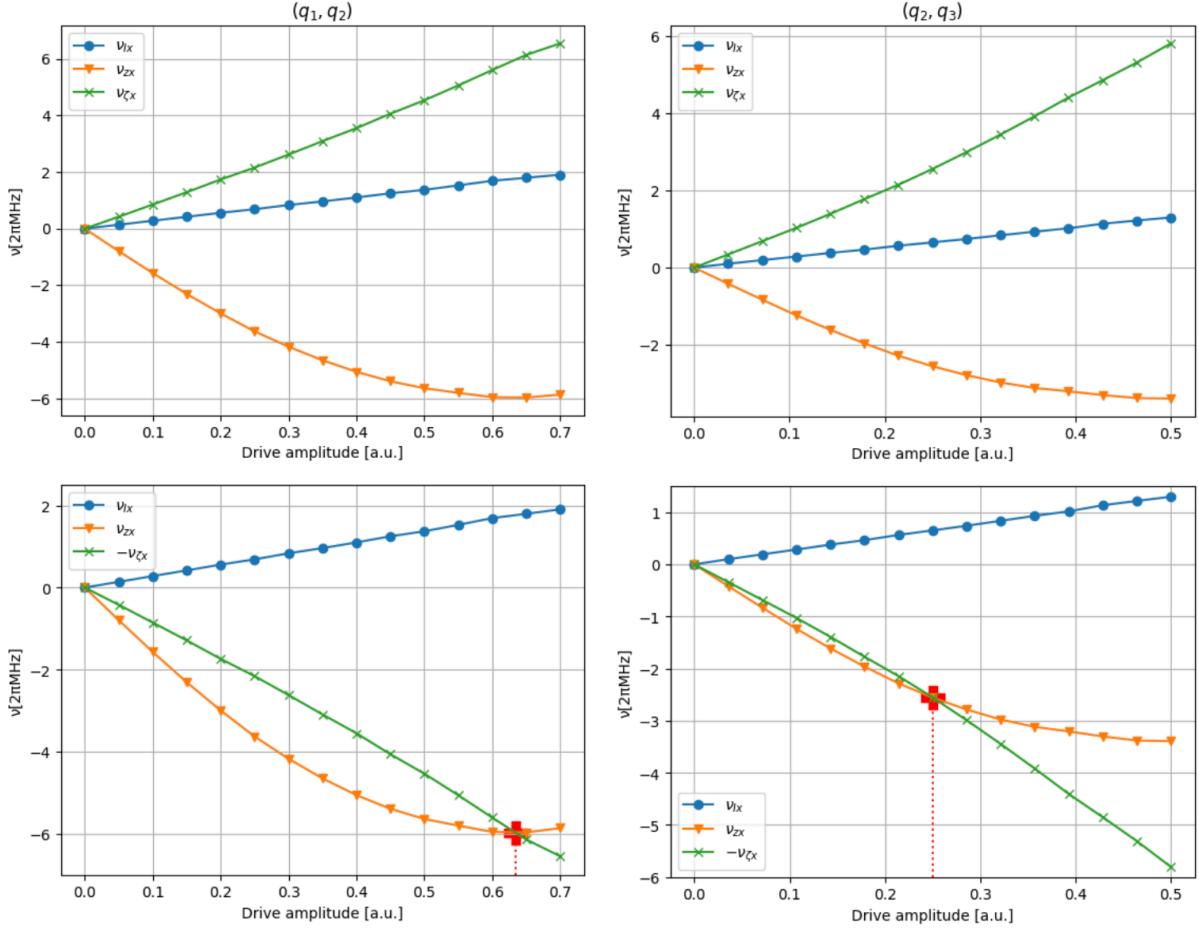


図 5.1.4: CR_{trit}⁺におけるパルス A の振幅と ν_{zx} 、 $\nu_{\zeta x}$ 、 ν_{Ix} の関係。左が (q_1, q_2) 、右が (q_2, q_3) のパルス A についてのプロットである。上図は ν_{zx} 、 $\nu_{\zeta x}$ 、 ν_{Ix} をそのままプロットし、下図は $\nu_{\zeta x}$ の符号を反転して $\nu_{zx} = -\nu_{\zeta x}$ となる振幅を fitting による交点から予想している。

$\nu_{zx} = -\nu_{\zeta x}$ となると予想できる。また図 5.1.4 の ν_{Ix} を見ると、パルス A により標的トリットに Rx_{01} 成分があることがわかる。この Rx_{01} 成分を打ち消すようにパルス B の振幅を定めれば、式 (2.2.31) の \tilde{H}_{CR}^\pm が構成できる。実際にこのようなパルス B の振幅を特定し、パルス A、B を同時にドライブすることで構成された有効ハミルトニアン \tilde{H} は、次のようにになった：

$$\frac{\tilde{H}^{(1,2)}}{2\pi\text{MHz}} = 5.552 \frac{III}{2} - 20.006 \frac{IZ_{01}II}{2} - 0.794 \frac{IZ_{01}X_{01}I}{2} + 14.349 \frac{IZ_{12}II}{2} \quad (5.1.3)$$

$$+ 0.796 \frac{IZ_{12}X_{01}I}{2} + 0.207 \frac{IZ_{12}Z_{01}I}{2} \quad (5.1.4)$$

$$\frac{\tilde{H}^{(2,3)}}{2\pi\text{MHz}} = 0.839 \frac{III}{2} - 0.053 \frac{III Z_{01}}{2} - 3.000 \frac{II Z_{01}I}{2} - 0.330 \frac{II Z_{01}X_{01}}{2} \quad (5.1.5)$$

$$+ 0.052 \frac{II Z_{01}Z_{01}}{2} + 2.499 \frac{II Z_{12}I}{2} + 0.333 \frac{II Z_{12}X_{01}}{2} + 0.129 \frac{II Z_{12}Z_{01}}{2} \quad (5.1.6)$$

$$+ 0.098 \frac{II Z_{12}Z_{12}}{2} - 0.137 \frac{I Z_{01}II}{2} + 0.077 \frac{I Z_{01}Z_{01}I}{2} - 0.311 \frac{I Z_{12}II}{2} \quad (5.1.7)$$

$$+ 0.182 \frac{IZ_{12}Z_{01}I}{2} + 0.140 \frac{IZ_{12}Z_{12}I}{2} \quad (5.1.8)$$

$\tilde{H}^{(i,j)}$ は $(q_i, q_j)CR_{\text{trit}}$ の有効ハミルトニアンであることを表している。各係数が大きいため、両辺を $2\pi\text{MHz}$ で割った形で表示している。 $\tilde{H}^{(1,2)}$ と $\tilde{H}^{(2,3)}$ で表示されている項数が異なる理由は、ハミルトニアンに含まれる最大係数の 1%以上の項を表示させているからであり、 $\tilde{H}^{(1,2)}$ にも閾値は超えていないが、 $\tilde{H}^{(2,3)}$ と同様の項が含まれている。この結果を見ると、式 (2.2.31) の \tilde{H}_{CR}^+ が再現できており、 $\nu_{zx} = -\nu_{\zeta x}$ も満たしている。また [27] には記載されていなかった ZZ 相互作用が顕著に現れていることも確認できる。これが今回 ZZ 相互作用の対策設計を新たに開発するに至った理由である。

このパルス A、B の振幅を固定し、今度はドライブ時間を掃引することで $CR_{\text{trit}}^+(\theta)$ の $\theta = \pi/3$ となるようにする。 CR_{trit}^+ の中の ZX 成分の回転角を θ としたとき、 CR_{trit}^+ のドライブ時間と $\cos(\theta)$ の関係は、図 5.1.5 のようになった。

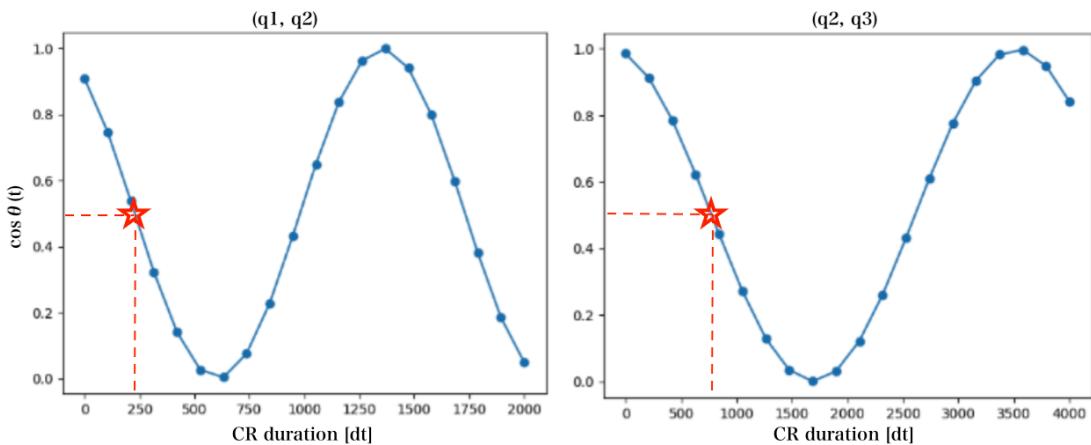


図 5.1.5: CR_{trit}^+ のドライブ時間と、ZX 成分の回転角 θ に対する $\cos \theta$ の関係。左が (q_1, q_2) 、右が (q_2, q_3) の CR_{trit}^+ についてのプロットである。 $\theta = \pi/3$ を特定するために、縦軸が $\cos(\frac{\pi}{3}) = 0.5$ となるドライブ時間を求めていて、星型プロットがその点に対応する。[dt] とは実機で用いられる時間の単位で、 $1[\text{dt}] = 0.22[\text{ns}]$ である。

図 5.1.5 の結果から、 $(q_1, q_2)CR_{\text{trit}}^+$ は $51[\text{ns}]$ 、 $(q_2, q_3)CR_{\text{trit}}^+$ は $172[\text{ns}]$ とドライブ時間を定めれば、 $CR_{\text{trit}}^+(\pi/3)$ が作成できることがわかる。同様の工程で $CR_{\text{trit}}^-(\pi/3)$ も作成できる。以上より作成した X_{01} 、 X_{12} と $CR_{\text{trit}}^\pm(\pi/3)$ を図 2.2.2 のように実装すれば、 $(q_1, q_2)CNOT_{\text{trit}}$ と $(q_2, q_3)CNOT_{\text{trit}}$ が作成できる。

ベリーワイド相対策

まず (i) $(q_0, q_1)CNOT_{\text{bit}}$ と (v) $(q_0, q_1)CNOT_{\text{bit}}^\dagger$ 、(ii) $(q_1, q_2)CNOT_{\text{trit}}$ と (iv) $(q_1, q_2)CNOT_{\text{trit}}^\dagger$ の忠実度を比較する。なおこの段階では ZZ 相互作用対策が実装されていない忠実度である。忠実度を算出したところ、(i) $(q_0, q_1)CNOT_{\text{bit}}$ は 0.9504、(v) $(q_0, q_1)CNOT_{\text{bit}}^\dagger$ は 0.9504、(ii) $(q_1, q_2)CNOT_{\text{trit}}$ は 0.9750、(iv) $(q_1, q_2)CNOT_{\text{trit}}^\dagger$ は 0.9751 であった。したがって図 4.1.4 と図 4.1.5 で紹介した CNOT † 用のシーケンスによって、元々の CNOT と同等の忠実度を保ったまま、新たに考案したベリーワイド相消設計が導入できている。

続いて 4.1.2 節で説明したように (q_2, q_3) CNOT_{trit} に Virtual Z_{01} と Virtual Z_{12} を導入して忠実度を計測し、導入前の忠実度と比較する。 (q_2, q_3) CNOT_{trit} の忠実度は Virtual Z の導入前が 0.9435、導入後が 0.9435 であった。したがって 4.1.2 節で紹介した Virtual Z の導入により、 (q_2, q_3) CNOT_{trit} も忠実度を保ったままベリー位相を解消できている。

ZZ 相互作用対策

次に CCCX のシーケンスに含まれる CNOT(i)-(v) に対して、ZZ 相互作用対策がある場合とない場合で忠実度を比較し、各ゲートの忠実度が改善するかを確認した。結果は表 5.1.1 のようになり、全てのゲートで ZZ 相互作用対策による忠実度の向上が見られた。特に CNOT の中では (i)(iii)(v) の上昇幅が大きい。

(iii) に関しては、実装時間が長く、改善前の ZZ 相互作用によるエラーが他 CNOT と比べて大きいからと考えられる。実装時間は (i)(v) が 427[ns]、(ii)(iv) が 365[ns]、(iii) が 730[ns] となっており、(iii) は ZZ 相互作用によるエラーを受ける時間が最も長い。ZZ 相互作用は 3.2.2 節で説明した通り、 H_{stat} に含まれる効果であり、時間発展中は常に存在するエラーなので、実装時間に比例してエラーが大きくなる。

(i)(v) に関しては、結合 2 分の ZZ 相互作用が対策できるからだと考えられる。CNOT の制御ビットには、シーケンス内に既に ZZ 相互作用の対策設計が含まれていることを 4.2 章で述べた。したがって図 5.1.2 に含まれる CNOT の制御ビットの位置を見ると、(i)(v) は q_0 、(ii)(iv) は q_1 、(iii) は q_2 に ZZ 対策設計が含まれているため、表 5.1.1 の ZZ 対策なしの忠実度計測の時点で、既に (i)(v) の (q_0, q_1) 間、(ii)(iv) の $(q_0, q_1)(q_1, q_2)$ 間、(iii) の $(q_1, q_2)(q_2, q_3)$ 間の ZZ 相互作用は解消されていることがわかる。よって表 5.1.1 の ZZ 対策前後での忠実度差は (i)(v) の $(q_1, q_2)(q_2, q_3)$ 間、(ii)(iv) の (q_2, q_3) 間、(iii) の (q_0, q_1) 間の ZZ 相互作用を解消することによる差であり、(i)(v) の解消されるエラーは他 CNOT よりも結合一つ分大きいことがわかる。

各 CNOT の ZZ 相互作用は解消され、表 5.1.1 の通り CCCX 全体では 24%ほど忠実度が向上した。また CCCX 全体の忠実度が 0.9 以上であり、ゲート毎に確認したベリー位相対策がシーケンス全体でも機能しているといえる。

	(i)	(ii)	(iii)	(iv)	(v)	CCCX
ZZ 対策なし	0.9504	0.9750	0.9435	0.9751	0.9504	0.7289
ZZ 対策あり	0.9936	0.9885	0.9861	0.9883	0.9935	0.9028

表 5.1.1: CCCX とシーケンスに含まれる CNOT の忠実度。ZZ 相互作用対策がある場合とない場合で測定している。

ハードウェアパラメータによる制約

本研究で CNOT_{trit} をシミュレータで作成している際に、CNOT_{trit} が作成できる条件である $\nu_{zx} = -\nu_{\zeta x}$ となるようなパルス振幅がないようなトランズモンペアが存在することが判明した。実際に複数のトランズモンペアで、パルス振幅を掃引して CR_{trit}⁺ をドライブした時の \tilde{H}_{CR} の $\nu_{Ix}, \nu_{zx}, -\nu_{\zeta x}$ をプロットした結果が図 5.1.6 である。

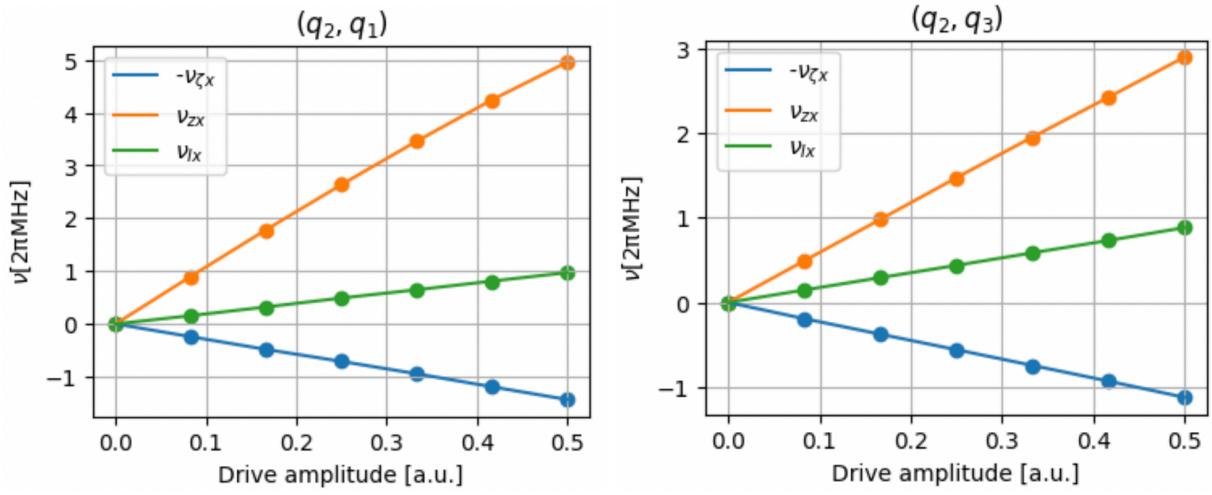


図 5.1.6: CR_{trit}⁺におけるパルス A の振幅と ν_{zx} 、 $-\nu_{\zeta x}$ 、 ν_{Ix} の関係。ibm-hanoi のハードウェアパラメータを参照してシミュレートした結果となっている。タイトルの (q_i, q_j) は q_i が制御、 q_j が標的であることを表している。このパターンの係数変化だと、 $\nu_{zx} = -\nu_{\zeta x}$ となるパルス振幅が現実的な値の範囲に存在しなかった。このパターンと同様の係数の振る舞いをするトランズモンペアが複数存在した。

先行研究 [47] では、2 個のトランズモン間の CR パルスにおいて、パルス振幅に対する ν_{zx} の振る舞いは、2 個のトランズモンの遷移周波数 ω_{01} と非調和性 δ というハードウェアパラメータによって定まることが、計算によって明らかになっている。 $\nu_{\zeta x}$ の振る舞いは先行研究 [47] では触れられていないが、 $|1\rangle|2\rangle$ 部分空間における遷移周波数 $\omega_{12} = \omega_{01} + \delta$ と非調和性 δ を、 $|0\rangle|1\rangle$ 部分空間における ω_{01} と非調和性 δ と対応づけて計算できることから、 $\nu_{\zeta x}$ も遷移周波数と非調和性に依存していると考えられる。したがって $\nu_{zx} = -\nu_{\zeta x}$ が成立するか否かは遷移周波数と非調和性というハードウェアパラメータに依存していると考えられる。

本研究で提案している量子トリットを用いた多制御ビットゲートのシーケンスは、制御ビット数 n の時に CNOT_{trit} を $n-1$ 個連続で使用することから、CNOT_{trit} を作成できるトランズモンペアが $n-1$ 組連続で結合している必要があるため、 n が大きくなるほどこのハードウェアパラメータの制限は厳しくなる。実際に ibm-hanoi という 27qubits の実機のハードウェアパラメータを引用して、 $n=3$ の CCCX が作成できる ($\nu_{zx} = -\nu_{\zeta x}$ が成立するトランズモンペアが 2 組連続で結合している) トランズモンペアをシミュレータで確認したところ、7 組であった。

この制限を解消し、実機での汎用性を向上させるためには、 $\nu_{\zeta x} = -\nu_{zx}$ という条件を使用しない CNOT_{trit} の作成方法を開発する必要がある。本研究では開発まではできなかったため、今後の課題とする。

5.2 IBMQuantum での評価

5.2.1 実験手法

評価項目

5.1 節の qudit-sim による結果は、電荷分散が考慮されていないことに加え、qudit-sim に動的にエネルギー準位 $|2\rangle$ がゆらぐという電荷分散の効果を時間発展に組み込む機能がなかった。一方で、IBMQuantum では電荷分散が生じていることが [27] でも確認されている。したがって IBMQuantum では主に電荷分散への対策設計の評価を行う。実機の評価で注意すべき点は、有効ハミルトニアンの解析が使えないため、忠実度がどの効果によって下がっているかを区別できないことである。電荷分散と ZZ 相互作用によるエラーは、両方とも位相エラーとして現れるため、実機で位相エラーが生じていても、どちらの効果によるものかの同定が困難である。したがって電荷分散の対策設計によって電荷分散の位相エラーが解消されているかを確認するためには、ZZ 相互作用による位相エラーが解消されていることが前提となる。本研究で確認したい電荷分散の対策設計は、多制御ビットゲート用（特に実際に評価するのは CCCX 用）のものであり、シーケンス中には ZZ 相互作用の対策設計も含まれている。そこで実機環境でも、qudit-sim で評価した ZZ 相互作用対策が機能するかを簡単なテストで評価してから、電荷分散対策の評価を行う。

QPT 実験の方法

実は図 4.1.2 に含まれる全ての CR ゲートは、電荷分散の位相エラー対策を設計する際に無視して考えられる。電荷分散の問題点は、 $|2\rangle$ が揺らぐことによる非共鳴効果で位相エラーを $|2\rangle$ に引き起こし続けるというものであった。CR ゲートの有効ハミルトニアン（式 (2.2.25)(2.2.31)）には、 $|2\rangle$ の状態を遷移させる x_{12} のような生成子が存在しないので、CR ゲートがあってもなくても電荷分散による位相エラーは $|2\rangle$ に加わり続けるだけである。したがって、設計の段階でも実証の段階でも、CR を無視しても問題ない。

そこで CCCX から CR ゲートを除いたゲートシーケンスで電荷分散の対策設計を評価する。

この場合、量子ビットごとに QPT が実装できるという大きなメリットがある。CCCX から CR ゲートを除いたゲートシーケンスでは、重ね合わせの状態が発生しない。重ね合わせ状態がない場合、ZZ 相互作用は初期状態に依存した位相エラーとして現れるため、量子状態がエンタングルしない。そこで QPT を実装する量子ビット以外の 3 つの量子ビットの初期状態を $|000\rangle - |111\rangle$ の 8 通り用意して、それぞれの初期状態の場合に対して、QPT を行って忠実度を計測する、という工夫をする。

この工夫により実験数を大幅に減らすことができる。本実験でのトモグラフィ実験数は

$$8(\text{initial states}) \times 4(\text{qubits}) = 32(\text{experiments})$$

になり、1 量子ビット系のトモグラフィ実験は

$$4(\text{initial states}) \times 3(\text{measurement basis}) = 12(\text{circuits})$$

なので、一度の評価実験に対して実行する総量子回路数は

$$12(\text{circuits}) \times 32(\text{experiments}) = 384(\text{circuits})$$

となる。実際にこの解析を実機で行うと、一度で 20~30 分 (+実機の順番待ちで数時間) であった。4 量子ビット系での QPT では

$$4^4(\text{initial states}) \times 3^4(\text{measurement basis}) = 20736(\text{circuits})$$

であり、一度の解析に非常に時間がかかることが予想されるため、本研究で 4 量子ビット系の QPT 解析は行わなかった。

本来であれば、CR も含めた量子トリット CCCX と、IBMQQuantum で採用されているデフォルトの量子ビット CCCX を QPT を用いて忠実度を測定し、精度比較をしたいところだが、長時間 IBMQuantum を占有することが難しかったため、本研究ではその足がかりとして、多制御ビットゲートを量子トリットで実装する際に予想される 3 つの大きなエラーへの対策設計を評価するに留めている。

5.2.2 実験

電荷分散の対策設計を確認するために行った実験と結果を記していく。なお実機は ibm_cairo と ibm_hanoi のうち、実験の際に順番待ちの少ない方を選んで使用している。実機では時間の単位として [dt] が用いられているが、 $1[\text{dt}] = 0.22[\text{ns}]$ である。また忠実度を求める際には、QPT を 10000 回実行することで、量子過程を同定して一つのプロットを出している。

ZZ 相互作用対策テスト

まずは簡単なテストで、実機環境でも ZZ 相互作用の対策設計が機能することを確認する。使用した 4 量子ビットは cairo の (q_0, q_1, q_4, q_7) である。テストには、ZZ 相互作用の対策設計がある場合とない場合で忠実度を比較する実験を行った。

ZZ 相互作用対策ありのパルススケジュールとして、図 5.2.1 のように (q_0, q_1) の ZZ 相互作用を解消するために q_0 に X_{01} ゲート 2 回を 12 セット、 $(q_1, q_2)(q_2, q_3)$ の ZZ 相互作用を解消するために q_2 に X_+ ゲート 3 回を 4 セット、等間隔の delay を入れて組んだものを使用した。実際のパルススケジュールは図 5.2.1 である。

ZZ 相互作用対策なしのパルススケジュールとして、今度は上記のスケジュールの合計 delay 時間分の ZZ 相互作用エラーを測定するため、ZZ 相互作用対策ありで使用したゲートをスケジュールの最初に詰めて、その後の ZZ 対策設計ありの合計 delay 時間分の delay を入れたものを使用した。実際のパルススケジュールは図 5.2.2 である。

使用したゲートを最初に詰める理由は、ゲートエラーによる忠実度の低下に対して対照な実験にすることで、忠実度の差として考えられる主な要因を ZZ 相互作用によるエラーに絞るためである。なおスケジュール時間も同じであるため、デコヒーレンスによる忠実度の低下に対しても対照な実験になっている。また図 5.2.1 と図 5.2.2 のパルスシーケンスは両方とも電荷分散による位相エラーが生じない形で設計したので、忠実度の結果に電荷分散の影響はない。したがって、この二つのパルススケジュールによる忠実度差の主な要因は ZZ 相互作用と考えられる。

この二つの実験に対し、5.2.1 節で述べた QPT 方法により忠実度を測定した結果、図 5.2.3 のようになった。まず 5.2.1 節で説明したように、本実験の ZZ 相互作用は初期状態に依存した位相エラーとして

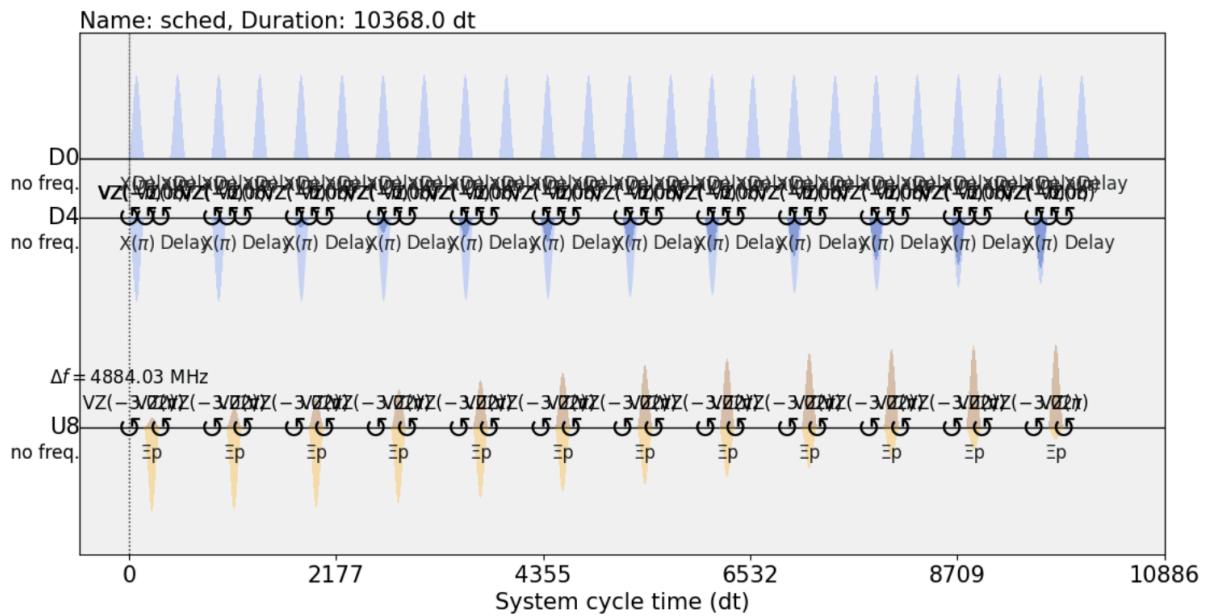


図 5.2.1: ZZ 相互作用対策のテストのうち、ZZ 相互作用対策ありのパルススケジュール。D0 は q_0 に X_{01} を、D4 は q_2 に X_{01} を、U8 は q_2 に X_{12} を実装するためのチャンネルとなっている。青のパルスが X_{01} 、黄のパルスが X_{12} を表している。

表れる。それを踏まえて ZZ 対策なしのプロットを見ると、確かに初期状態に依存して忠実度が上下しており、特に QPT の対称となっている量子ビットと結合している隣の量子ビットの初期状態が $|1\rangle$ のときに忠実度が下がっているため、ZZ 相互作用による位相エラーが生じていることが確認できる。一方で ZZ 対策ありのプロットを見ると忠実度は初期状態によらず横並びで一定なので、ZZ 相互作用の効果は解消されていると考えられる。

しかしここで予想外にも、ZZ 対策ありの実験において X_{01} や X_+ を施していない q_1 と q_3 の忠実度が ZZ 対策なしの実験よりも低いことが判明した。量子計算においてエラーを生む主な原因に対しては対照的な実験となっていることから、これまでの議論で考慮されていないエラーであると考え、エラーの特定を始めた。

想定外のエラーの特定

まず q_0 と q_2 に入れていた ZZ 対策設計を q_1 と q_3 に入れてみたが、結果は q_1 と q_3 の忠実度が高く q_0 と q_2 の忠実度が低い、という結果になったため、「ZZ 対策設計を入れている量子ビットの忠実度が保たれ、入れていない量子ビットの忠実度が低下する」ということが特定できた。また先程の図 5.2.1 の X_+ や X_{01} 間の delay を変化させて複数回実験を行ったところ、delay に依存して q_1 と q_3 の忠実度が変化していることが判明した。したがって「時間に依存して生じる形のエラー」であると予想できる。

ここでこのエラーが $q_0 - q_3$ の全ての量子ビットに加わっている時間依存の位相エラー $e^{-i\sigma_z\nu_z t}$ という形を仮定すると、上の予想に対して無矛盾であった。なぜなら ZZ 対策設計を入れている q_0 と q_2 は、

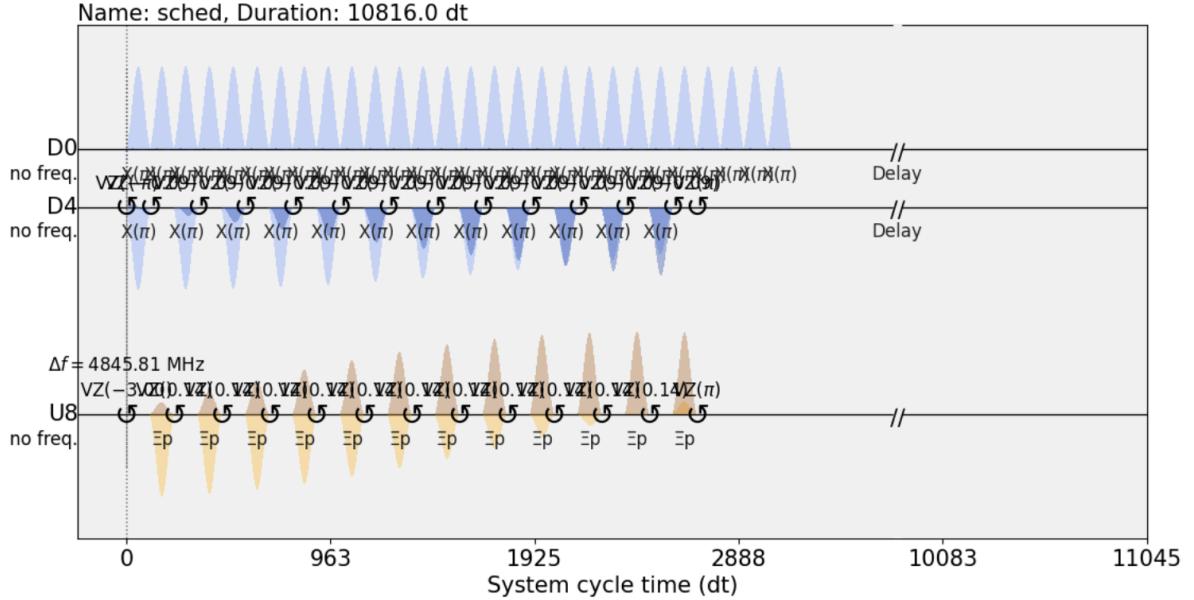


図 5.2.2: ZZ 相互作用対策のテストのうち、ZZ 相互作用対策なしのパルススケジュール。D0 は q_0 に X_{01} を、D4 は q_2 に X_{01} を、U8 は q_2 に X_{12} を実装するためのチャンネルとなっている。青のパルスが X_{01} 、黄のパルスが X_{12} を表している。スケジュール時間は 2404[ns] である。

X_{01} と X_+ の実装間隔をそれぞれ $t = \alpha$ 、 $t = \beta$ とおくと、

$$e^{-i\sigma_z\nu_z\alpha} X_{01} e^{-i\sigma_z\nu_z\alpha} X_{01} = e^{-i\sigma_z\nu_z\alpha} e^{i\sigma_z\nu_z\alpha} = I \quad (5.2.1)$$

$$e^{-i\sigma_z\nu_z\beta} X_+ e^{-i\sigma_z\nu_z\beta} X_+ e^{-i\sigma_z\nu_z\beta} X_+ = I \quad (5.2.2)$$

となり、時間依存の位相エラーが解消できるが、ZZ 対策設計なしの q_1 と q_3 はこの位相エラーが蓄積し続けるからである。また時間依存であることから、ZZ 相互作用と同じく H_{stat} に含まれる効果であると考え、「 H_{stat} に σ_z の効果が含まれている」という仮説を立てた。この仮説を検証するため、追加実験を行なった。

まずエラーの効果が x や y 軸回転といった状態遷移ではなく、 z 軸回転の位相エラーであることを同定するために、次のような解析を行った。先程の ZZ 対策あり実験の際に QPT で特定した量子演算 G に対し、図 5.2.3 ではアイデンティティ I との忠実度をプロットに出したが、今度は I ではなく、 $Rz(\phi) = e^{-i\sigma_z\frac{\phi}{2}}$ との忠実度を測定する。この測定を行う理由は、もし ϕ を掃引したときの最大忠実度が 1 に近いようであれば、実験結果が $Rz(\phi)$ だけアイデンティティから誤差があったという意味になり、忠実度低下の要因が位相エラーであると同定できるからである。

この解析の結果は図 5.2.4 のようになった。二段目の q_1 と四段目 q_3 の最大忠実度が 0.8 以上であるため、図 5.2.3 の q_1 と q_3 に生じていた忠実度低下の要因の多くは Rz 要素であったことが判明し、主なエラーの原因は位相エラーであると同定できた。

次に上で確認した位相エラーを引き起こす σ_z 項が H_{stat} に含まれるものであるかどうかを同定する。もし H_{stat} に σ_z 項が含まれている場合、パルススケジュール中の合計 delay 時間 τ に応じて $e^{-iH_{\text{stat}}\tau} = e^{-i\nu_z\sigma_z\tau}$ という時間発展が生じていることになる。このとき、 I に対する忠実度は τ を用いて次のように与えら

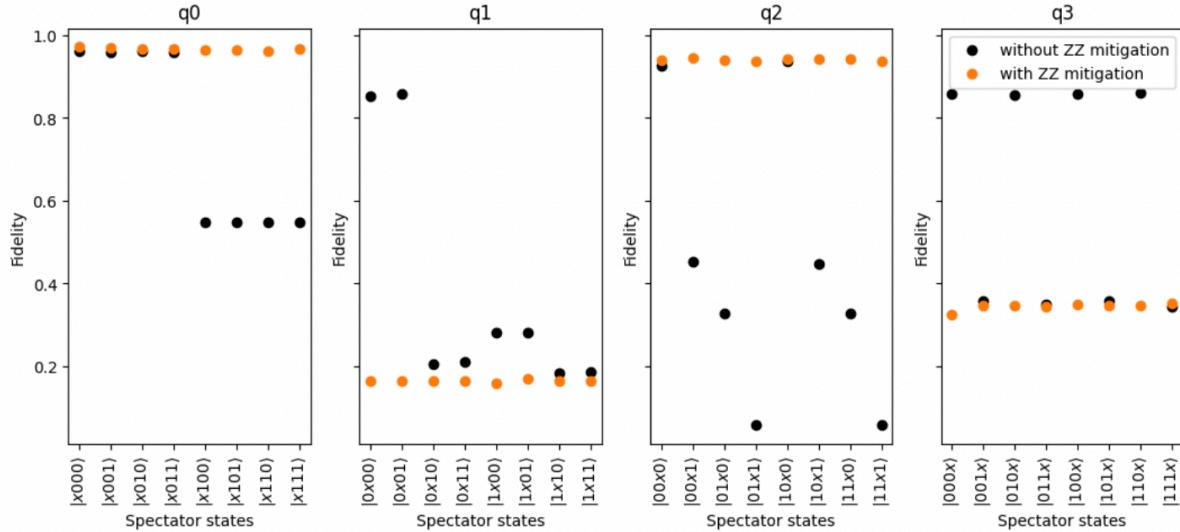


図 5.2.3: ZZ 相互作用対策テストのシーケンスを実装し、QPT を使ってアイデンティティに対する忠実度を測定した結果。 q_0 q_3 の中で、横軸ラベル内の x が QPT を行なっている量子ビットで、他の 3 量子ビットを $|000\rangle - |111\rangle$ の 8 通りの初期状態から実験している。黒が ZZ 相互作用対策なし、オレンジが ZZ 相互作用対策ありを表している。スケジュール時間は 2404[ns] である。

れる：

$$F = \text{Tr}[e^{-i\nu_z\sigma_z\tau} I] = (e^{-i\nu_z\tau} + e^{i\nu_z\tau})/2 = \cos(\nu_z\tau) \quad (5.2.3)$$

したがって τ を掃引すると、忠実度は cos 波で fitting できるはずである。この効果が見られるか確認するため、 delay_T を変えて複数の忠実度を測定したところ、図 5.2.5 のような結果となった。

式 (5.2.3) の示す通り、忠実度が sin カーブで fit できたことから、 delay に依存した大きさの位相エラーが加わっていることが確認できた。これはパルススケジュール中、パルスの有無によらず常に σ_z の効果が生じていることを意味するため、仮説通り σ_z 項が H_{stat} に含まれていると考えられる。この位相エラーが生じる原因についての考察は 5.2.4 節に記す。

ここで実機で生じている H_{stat} による位相エラーについて、式 (2.1.12) で定義したゲルマン行列による生成子と共に整理する。実機による ZZ 対策テスト前までは、量子トリット空間で H_{stat} に含まれる大きなエラーは ZZ 相互作用 ($\lambda_3 \otimes \lambda_3, \lambda_3 \otimes \lambda_8, \lambda_8 \otimes \lambda_3, \lambda_8 \otimes \lambda_8$) と電荷分散 (λ_8) のみだと考えていた。しかし実機環境では、上記のエラーに加えて σ_z が含まれていることが確認できた(以降 local Rz と記す)。この local Rz の生成子は量子トリット空間では、 λ_3 に対応する。したがって H_{stat} に含まれるエラーは

- ZZ 相互作用 ($\lambda_3 \otimes \lambda_3, \lambda_3 \otimes \lambda_8, \lambda_8 \otimes \lambda_3, \lambda_8 \otimes \lambda_8$)
- 電荷分散 (λ_8)
- local Rz(λ_3)

の 3 種類あることが判明した。

留意する点として、この 3 種類のエラーはすべて位相エラーとして観測される。5.2.1 節で述べた通り、実機の QPT では位相エラーがどの効果によるものかの同定が困難である。したがって電荷分散の対策

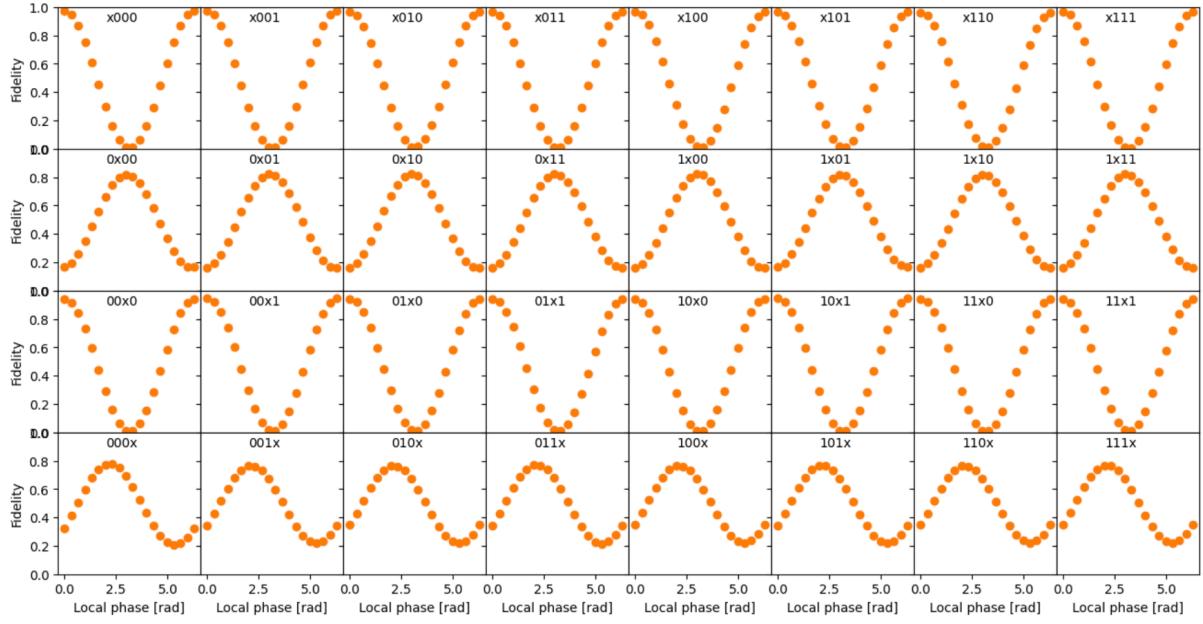


図 5.2.4: ZZ 対策ありの実験において、 q_0-q_3 の $|000\rangle - |111\rangle$ という 8 通りの初期状態での $Rz(\phi)$ に対する忠実度。各プロットの上部に記されているラベル内の x が QPT を行なっている量子ビットである。横軸が位相 ϕ 、縦軸が忠実度となっている。スケジュール時間は 2404[ns] である。

設計を評価するためには、ZZ 相互作用に加えてこの local Rz の位相エラーも解消する必要がある。

local Rz 対策

まず local Rz のエラーは演算時間中にパルスの有無によらず、常にかかり続ける λ_3 由来の位相エラーであるため、時間 t の間に生じる local Rz は、単位時間あたりに生じる位相エラーの大きさ ν_3 を用いて、 $Rz_{01}(\nu_3 t)$ と表せる。例えばパルススケジュールの中で、時刻 $t = t_0$ と $t = t_1$ で X_{01} をドライブした場合、local Rz を考慮した等価回路は図 5.2.6 となる。このゲートシーケンスを行列計算すると、次のような結果となる：

$$X_{01} Rz_{01}(\nu_3(t_1 - t_0)) X_{01} Rz_{01}(\nu_3(t_0)) = Rz_{01}(-\nu_3(t_1 - t_0)) Rz_{01}(\nu_3(t_0)) \quad (5.2.4)$$

$$= Rz_{01}(\nu_3(2t_0 - t_1)) \quad (5.2.5)$$

したがって、 $t_1 = 2t_0$ 、つまり等間隔で X_{01} を実装すれば local Rz は対策できることがわかる。ここで問題となるのは、ZZ 対策設計を入れていないトランズモンに対して、他トランズモンの ZZ 対策設計を邪魔しないように、local Rz 対策設計を組み込む必要がある、という点である。

実はこのような組み方は複数考えられるのだが、今回は一番分かり易い設計と、実際にこの後 IBMQQuantum で使用する設計の 2 種類を紹介する。まず最もわかりやすい local Rz の組み込み方として、図 5.2.7 のようなゲートシーケンスを考える。 q_0 と q_2 に実装する ZZ 対策設計を「 q_0 に X_{01} 6 回と q_2 に X_{+3} 回」で 1 セットとして考えた場合、この 1 セットで、1 セット内に生じる分の $(q_0, q_1)(q_1, q_2)$ 間の ZZ 相互作用は完全に解消できている。その設計を 2 セット（図 5.2.7 の青領域と赤領域）用意した場合、片方のセットを q_1 の X_{01} で挟んでも、1 セット毎にセット内の ZZ 相互作用は解消されているため、 q_1

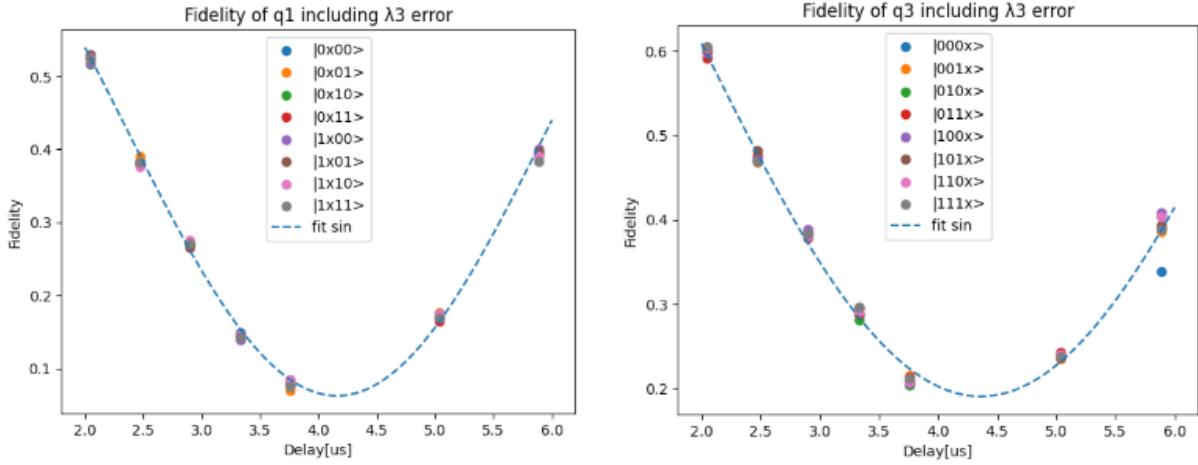


図 5.2.5: ZZ 対策ありのパルスシーケンスにおいて、複数の delay で実験を行い、 q_1 (左) と q_3 (右) の忠実度を計測した。cos 波の代わりに sin 波で fitting を行い、プロットが sin 波上に出ていることが確認できる。

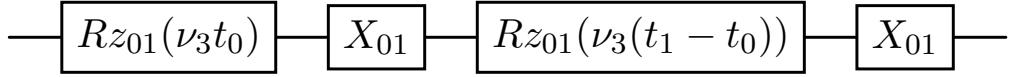


図 5.2.6: 時刻 $t = t_0$ と $t = t_1$ で X_{01} をドライブした場合の local Rz を考慮した等価回路。

の X_{01} は ZZ 対策設計の機能に関与しない。しかし q_1 には等間隔に X_{01} が実装できているため、 q_1 の local Rz 対策ができている。

この図 5.2.7 の local Rz 対策は、ZZ 対策設計 2 セット毎に local Rz を解消できる設計になっているが、実際の CCCX ゲートシーケンスでは図 5.1.2 が示す通り、ZZ 対策設計を入れるトランズモンは CNOT の実装時間ごとに切り替える必要がある。実は CNOT の実装時間は、ZZ 対策設計を 2 セット組み込むほど長くないため、他の local Rz 対策設計が必要となる。

そこで図 5.2.8 のようにアップデートを行なった。これは、ZZ 相互作用よりも短い時間間隔で local Rz 対策設計を組み込んだものになっていて、図 5.2.7 よりも青赤領域の切り替わりが多くなっている。これは図 5.2.7 の青(赤)領域の時間発展を図 5.2.8 の短い青(赤)領域 6 個に分割して時間発展させているようなゲートシーケンスになっている。

図 5.2.7 の青領域の時間発展ハミルトニアンを H_a 、赤領域を H_b とし、青(赤)領域の時間が τ であるとすると、図 5.2.7 のゲートシーケンスによる時間発展は $e^{-iH_b\tau}e^{-iH_a\tau}$ と表現でき、図 5.2.8 は

$$\left(e^{-iH_b \frac{\tau}{6}} e^{-iH_a \frac{\tau}{6}} \right)^6 = e^{-iH_b\tau} e^{-iH_a\tau} \quad (5.2.6)$$

と表現できるため、図 5.2.8 の(短い青領域 6 個 + 短い赤領域 6 個)をセットで考えると、図 5.2.7 の(青領域 1 個 + 赤領域 1 個)のセットと同じだけの時間発展を起こしている。したがって図 5.2.8 の設計は図 5.2.7 の設計と同様に ZZ 相互作用対策 + local Rz 対策ができているのである。

この設計の場合、ZZ 対策設計が 1 セットであっても local Rz 対策と併用できていることから、CNOT の実装時間中にも組み込めるため、CCCX のゲートシーケンス中にも使用できる形になった。

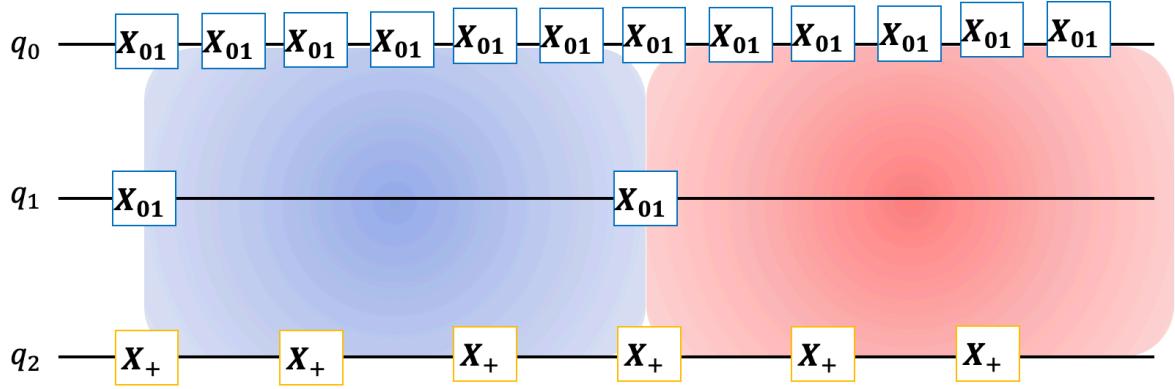


図 5.2.7: ZZ 相互作用 + local Rz 対策ありのゲートシーケンス。 q_0 と q_2 には ZZ 相互作用の対策設計、 q_1 には新たに local Rz 対策設計を実装している。こちらは最もわかりやすい local Rz 対策設計の入れ方として紹介している。赤領域、青領域は ZZ 対策設計 1 セットを表しており、青赤領域を合わせて 1 セットと考えれば、local Rz も解消できている。

一度この設計が機能するかを確かめるため、図 5.2.9 のパルスシーケンスのように q_0 と q_2 に ZZ 対策設計を入れ、 q_1 と q_3 に local Rz 対策設計を入れたものでテストを行う。なお、ここからは $\text{hanoi}(q_0, q_1, q_4, q_7)$ の 4 量子ビットを使用している。この設計が ZZ 相互作用と local Rz による位相エラーを両方とも解消できているか確認するため、再度理想的なゲートを $Rz(\phi)$ とし、 ϕ を掃引して最大忠実度を求めた。この ϕ を掃引する実験を行う理由は $\phi = 0$ で最大忠実度となっていたら、このパルスシーケンスによって全ての位相エラーが解消できていると言えるからである。この実験結果は図 5.2.10 となった。全ての量子ビットで、全ての初期状態に対して $\phi = 0$ で最大忠実度となっているため、位相エラーが解消できていることが確認できた。したがって ZZ 相互作用に加えて想定外の位相エラーである local Rz も解消できた。

また再度図 5.2.3 で行なったように、ZZ 相互作用と local Rz の対策なしのパルスシーケンスと ZZ 相互作用と local Rz の対策ありのパルスシーケンスを用意して、 I に対する忠実度を比較した。その結果、図 5.2.11 のようになった。ZZ 相互作用+local Rz 対策を施した場合、対策なしと比較して、全ての量子ビットで全ての初期状態において忠実度が高く保たれている。

電荷分散対策の評価

位相エラーを引き起こす ZZ 相互作用と local Rz の対策が機能することが確認できたので、最後に本来の目的であった電荷分散の対策設計を評価する。

まずは使用する量子トリットで電荷分散が生じていることを確認するため、Ramsey 実験を利用する [50]。Ramsey 実験とは、ドライブ周波数と $|0\rangle - |1\rangle$ 遷移周波数の差分 δ を計測する実験であり、ドライブ周波数の較正としてしばしば使用される。実験の概要は次のようにになっている。

ドライブ周波数が $|0\rangle - |1\rangle$ 遷移周波数から δ だけ誤差がある場合、非共鳴効果によって時間 t の間に $e^{-i\delta\sigma_z \frac{t}{2}}$ の位相エラーが生じる。このエラーを次のような回路で測定する：

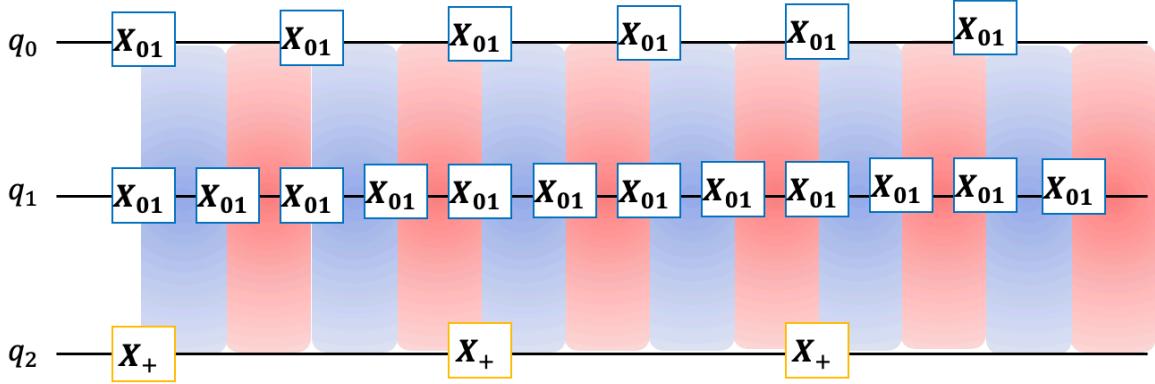
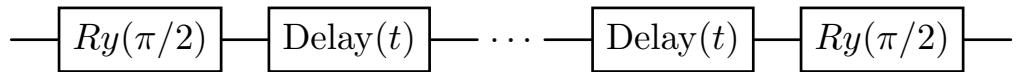


図 5.2.8: ZZ 相互作用 + local Rz 対策ありのゲートシーケンス。\$q_0\$ と \$q_2\$ には ZZ 相互作用の対策設計、\$q_1\$ には新たに local Rz 対策設計を実装している。こちらは実際に IBMQuantum で電荷分散対策の評価の際に使用した local Rz 対策設計の入れ方となっている。



この回路による終状態を初期状態 \$|0\rangle\$ から計算で追うと、次のようになる：

$$\begin{aligned}
 Ry(\pi/2)[Delay(t)]^n Ry(\pi/2)|0\rangle &= Ry(\pi/2)[Delay(t)]^n \frac{(|0\rangle + |1\rangle)}{\sqrt{2}} \\
 &= Ry(\pi/2) \frac{\left(e^{-i\delta\frac{nt}{2}}|0\rangle + e^{i\delta\frac{nt}{2}}|1\rangle \right)}{\sqrt{2}} \\
 &= \frac{\left(e^{-i\delta\frac{nt}{2}} \left(\frac{|0\rangle + |1\rangle}{\sqrt{2}} \right) + e^{i\delta\frac{nt}{2}} \left(\frac{-|0\rangle + |1\rangle}{\sqrt{2}} \right) \right)}{\sqrt{2}} \\
 &= -i \sin\left(\frac{\delta nt}{2}\right) |0\rangle + \cos\left(\frac{\delta nt}{2}\right) |1\rangle
 \end{aligned} \tag{5.2.7}$$

したがって終状態を測定して \$|1\rangle\$ となる確率 \$P(1)\$ は、

$$P(1) = \left| \cos\left(\frac{\delta nt}{2}\right) \right|^2 = \frac{1 + \cos(\delta nt)}{2} \tag{5.2.8}$$

となるため、delay 回数 \$n\$ を掃引したときの \$P(1)\$ をプロットすれば、\$\cos\$ 波でフィットすれば \$\delta\$ が測定できる。

この Ramsey 実験を \$|1\rangle - |2\rangle\$ 間で実装することを考えた。この場合も \$|1\rangle - |2\rangle\$ 遷移周波数とドライブ周波数の間に誤差 \$\delta\$ があれば、\$|0\rangle - |1\rangle\$ の Ramsey 実験同様、\$P(1)\$ が式 (5.2.8) のように、振動中心 1/2、振幅 1/2 の \$\cos\$ 波として現れる。しかし電荷分散がある場合、\$|1\rangle - |2\rangle\$ 遷移周波数が短時間で変動するため、誤差 \$\delta\$ は 2 種類以上存在する。例えば誤差が \$\delta_1\$ と \$\delta_2\$ の 2 種類存在するというモデルを仮定すると、このときの \$P^{(c)}(1)\$ は誤差 \$\delta_1\$ のときの \$P^{(1)}(1)\$ と誤差 \$\delta_2\$ のときの \$P^{(2)}(1)\$ の平均として定めら

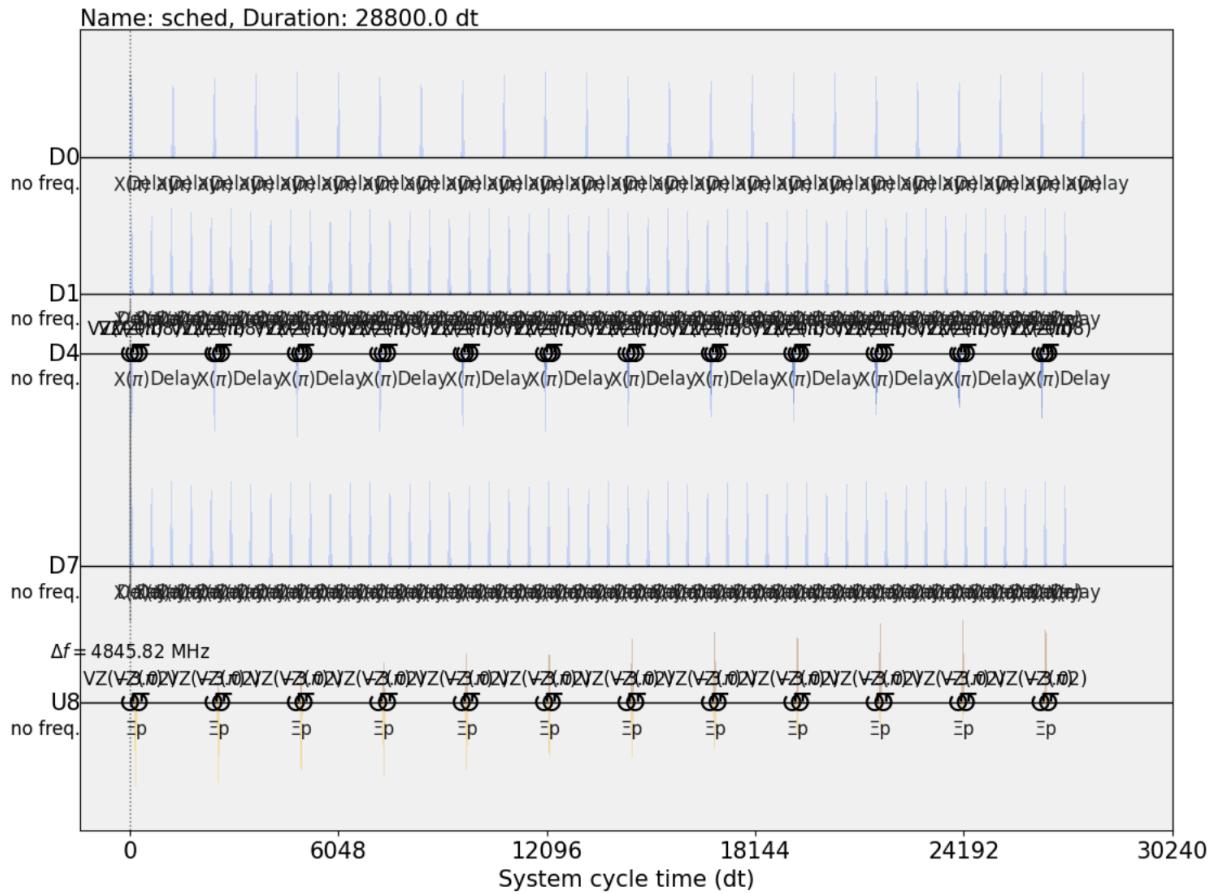


図 5.2.9: ZZ 相互作用 + local Rz 対策ありのパルスシーケンス。 q_0 と q_2 には ZZ 相互作用の対策設計、 q_1 と q_3 には新たに local Rz 対策設計を実装している。

れ、次のように与えられる：

$$\begin{aligned}
 P^{(c)}(1) &= \frac{1}{2} \left(P^{(1)}(1) + P^{(2)}(1) \right) \\
 &= \frac{1}{2} (\cos^2(\delta_1 nt) + \cos^2(\delta_2 nt)) \\
 &= \frac{\cos(2\delta_1 nt) + \cos(2\delta_2 nt)}{4} + \frac{1}{2} \\
 &= \frac{\cos(2(\delta_1 + \delta_2)nt) \cos(2(\delta_1 - \delta_2)nt)}{2} + \frac{1}{2}
 \end{aligned} \tag{5.2.9}$$

したがって $P^{(c)}(1)$ は周波数 $(\delta_1 + \delta_2)$ と $(\delta_1 - \delta_2)$ の cos 波が積で重ね合わせられた関数となり、概形は図 5.2.12 のような関数になると考えられる。電荷分散による $|1\rangle - |2\rangle$ が 3 種類以上だと更に cos 波が積で合成された概形になると予想される。

この予想を踏まえて、CCCX で量子トリットとして使用する q_1 と q_2 に対して $|1\rangle - |2\rangle$ で Ramsey 実験を行なった。結果は図 5.2.13 のようになった。予想される $P^{(c)}(1)$ の概形である図 5.2.12 の通り、複数の cos 波が合成されたようなプロットとなっている。

電荷分散が生じていることが確認できたため、次に行う実験は no CR CCCX（以降、no CR CCCX と記す）ゲートシーケンスに対して、「ZZ 対策あり+local Rz 対策あり+電荷分散対策なし」と「ZZ 対策あり+local Rz 対策あり+電荷分散対策あり」である。これまでの実験結果と考察を踏まえ、この2つの実験の忠実度を比較すれば、その忠実度差の主な要因は電荷分散に絞れていると考えられる。この実

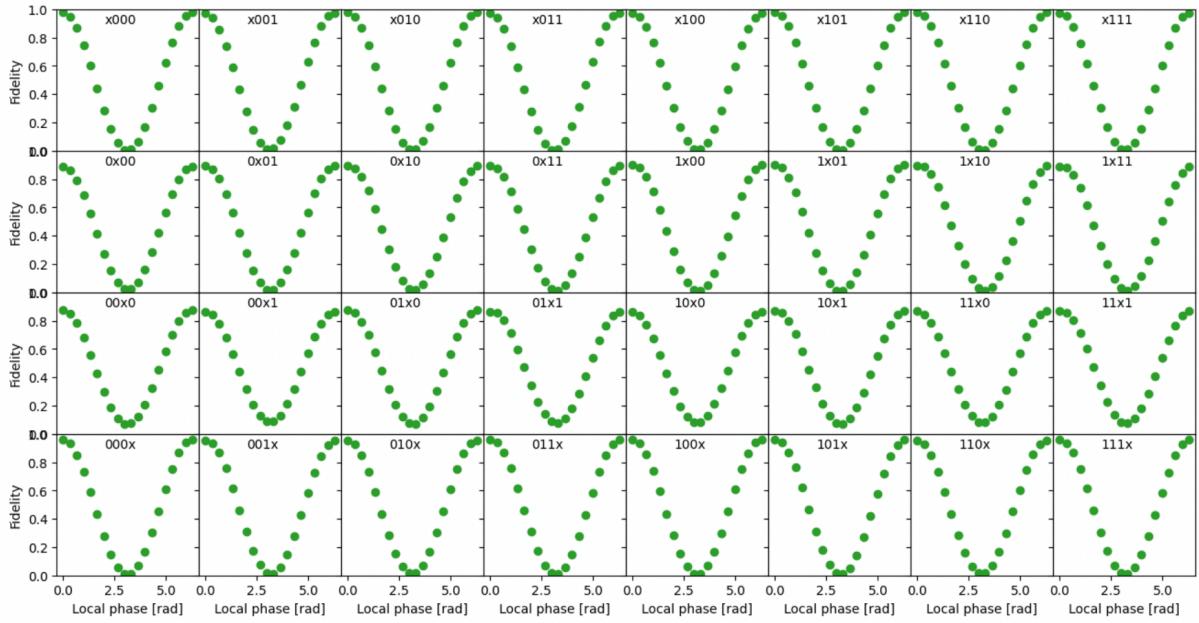


図 5.2.10: ZZ 相互作用 + local Rz 対策ありのパルスシーケンスにおける q_0-q_3 の $|000\rangle - |111\rangle$ という 8 通りの初期状態での $Rz(\phi)$ に対する忠実度。横軸が位相 ϕ 、縦軸が忠実度となっている。スケジュール時間は 6400[ns] である。

験のパルスシーケンスは図 5.2.14 となっており、左図が電荷分散対策設計なしの no CR CCCX、右図が電荷分散対策設計ありの no CR CCCX となっている。

本研究で開発した電荷分散対策が正しく機能する場合、対策後はゲート実装時間やハードウェアパラメータ、電荷分散の大きさに依らず位相エラーが解消できるはずである。これを確かめるため、実験は以下のように条件を変えて行う：

- case1: ibm_hanoi(q_0, q_1, q_4, q_7) の 4 量子ビットを利用して図 5.2.14 のパルスシーケンスを実装する
- case2: ibm_hanoi(q_0, q_1, q_4, q_7) の 4 量子ビットを利用して図 5.2.14 の ZZ 対策設計の delay を伸ばすことによってパルスシーケンスを case1 よりも長くして実装する
- case1: ibm_cairo(q_0, q_1, q_4, q_7) の 4 量子ビットを利用して図 5.2.14 のパルスシーケンスを実装する

これらの実験に対して忠実度を測定する。この実験の目的は、位相エラーが解消できていることを確認することなので、再度 ϕ を掃引して $Rz(\phi)$ との忠実度を計測する手法を用いて、 $\phi = 0$ で最大忠実度になるかどうか確かめる。

電荷分散対策なしの結果は図 5.2.15、電荷分散対策ありの結果は図 5.2.16 となった。電荷分散対策ありでは全ての case において、 $\phi = 0$ で最大忠実度となっており、パルスシーケンス中の全ての位相エラーが解消できていることが確認できた。一方で電荷分散対策なしでは $|2\rangle$ を使用する q_1 と q_2 において $\phi = 0$ で最大忠実度となっている case はなく、電荷分散による位相エラーが生じていることがわかる。

また電荷分散対策ありでは図 5.2.16 に通り sin 波の振幅が大きいが、電荷分散対策なしでは図 5.2.15 の通り忠実度 0.5 を中心として振幅の小さい sin 波が見られる。この現象については図 3.2.4 で見たように $|2\rangle$ が揺らいでいる効果が観測できたと考察した。例えば電荷分散によって $|2\rangle$ が $\omega = \omega_1$ と ω_2 に短

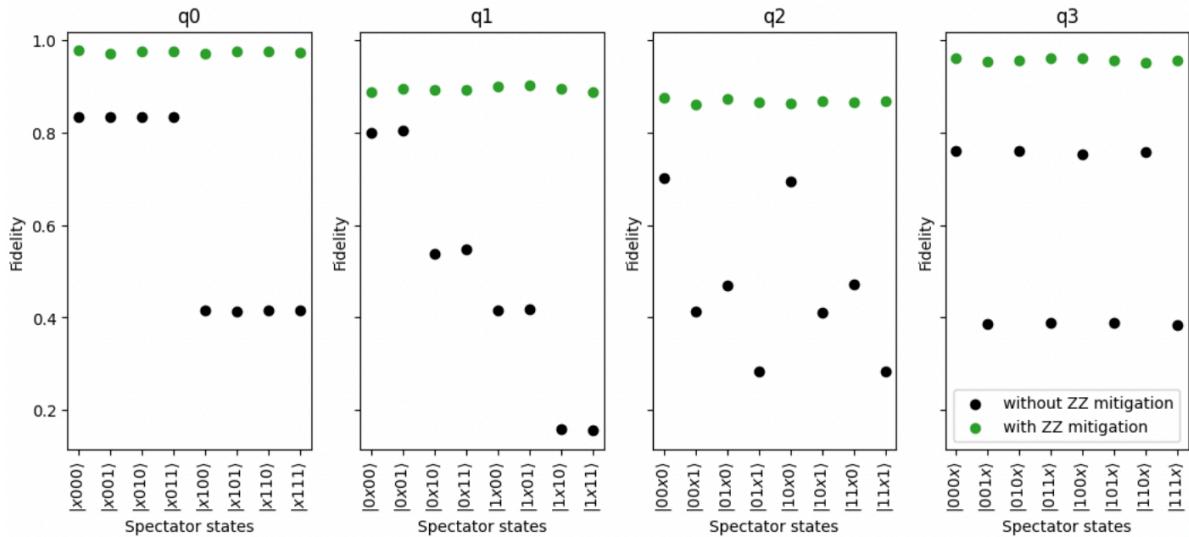


図 5.2.11: ZZ 相互作用+local Rz 対策ありの q_0 - q_3 において、 $|000\rangle - |111\rangle$ の 8 通りの初期状態でのアイデンティティに対する忠実度。黒が対策なし、緑が対策ありを表している。スケジュール時間は 6400[ns] である。

時間で切り替わるモデルを考えてみる。このとき電荷分散対策なしのパルスシーケンス(左図 5.2.14)を実装した結果、 ω_1 のときは $\phi = \phi_1$ 、 ω_2 のときは $\phi = \phi_2$ という別々の位相エラーが生じることになる。これを図 5.2.15 によって ϕ を掃引する実験のプロットとして表した場合、 $\phi = \phi_1$ 、 $\phi = \phi_2$ と別々の ϕ で忠実度のピークが生じるため、図 5.2.17 の 2 本の黒点線のように、重なっていない 2 つの sin 波として表れるはずである。ここで留意すべき点として、先程 5.2.2 節のはじめで述べた通り本研究では量子過程を 10000 回の QPT 実験により同定している。つまり 1 回ごとの QPT 実験では図 5.2.17 の黒点線で示した $\phi = \phi_1$ と $\phi = \phi_2$ のどちらかの sin 波に従う量子過程として同定されていたとしても、最終的に忠実度としてプロットを出すときには 10000 回を平均化したものとして同定されるため、黒点線が平均化されて図 5.2.17 の赤実線で示した sin 波のようなプロットとなる。したがって図 5.2.15 で見られた sin 波の振幅減衰は、電荷分散による影響だと考えられる。

5.2.3 最終的な忠実度比較と考察

最後に電荷分散対策ありとなしのパルスシーケンスを実装し、アイデンティティに対する忠実度を計測した結果を比較する。実験の条件は case3 を用いて、その結果は図 5.2.18 となった。量子トリットとして使う q_1 と q_2 では、全ての初期状態において対策前よりも対策後の方が忠実度が上回っているため、ZZ 対策設計、local Rz 対策設計も含めた電荷分散対策設計によって H_{stat} のエラーを抑えられ、電荷分散対策設計は機能したと考えられる。またこの設計はハードウェアパラメータに依らずに機能する設計になっているため、任意のトランズモンに対して有効であるといえる。

q_0 と q_3 は電荷分散対策なしの方が忠実度が高い結果を示しているが、これは電荷分散対策なしが 3.38[μ s]、ありが 6.36[μ s] のパルススケジュールとなっているので、デコヒーレンスによる忠実度差だと考えられる。また電荷分散対策ありでは q_0 と q_3 が同程度の忠実度、そこから少し値が下がって q_1 と q_2

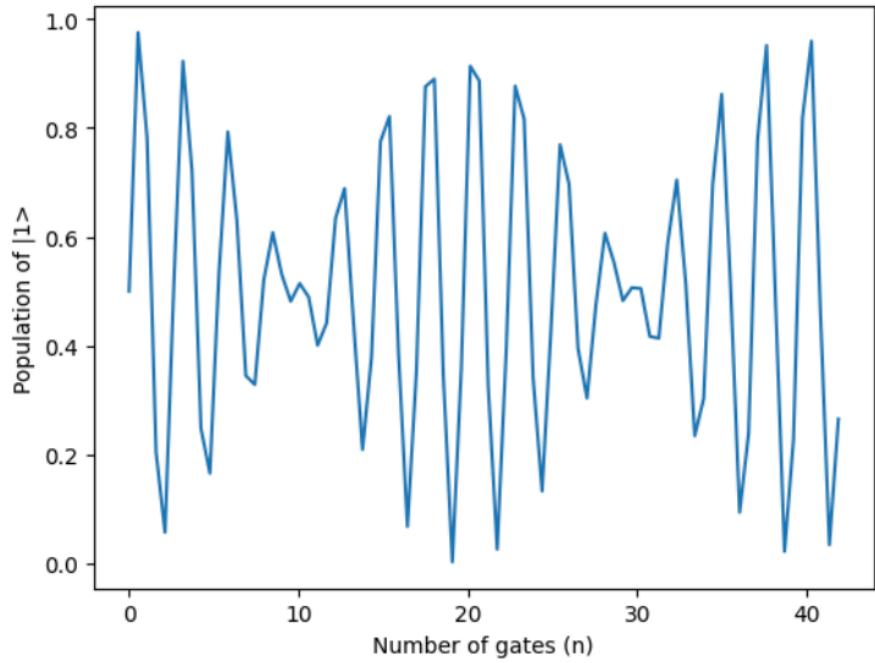


図 5.2.12: 電荷分散により $|1\rangle - |2\rangle$ 遷移周波数が離散的に 2 種類存在するというモデルにおける $P^{(c)}(1)$ の概形。縦軸は $P^{(c)}(1)$ (Population of $|1\rangle$)、横軸は delay ゲートの数 (n) である。

が同程度の忠実度を示しているが、こちらもデコヒーレンスによる忠実度差が現れていると考えられる。量子計算におけるデコヒーレンスの影響は、 $|0\rangle$ にいる時間、 $|1\rangle$ にいる時間、 $|2\rangle$ にいる時間によって定まり、エネルギー状態が高いほどデコヒーレンスによって忠実度は低下しやすい。ここで電荷分散対策ありのパルススケジュールにおいて、 q_0-q_3 の $|0\rangle$ にいる時間、 $|1\rangle$ にいる時間、 $|2\rangle$ を算出すると、 q_0 と q_3 、 q_1 と q_2 のデコヒーレンスによる影響はそれぞれ対称であることがわかり、かつ q_0 と q_3 は量子ビットとして使うため $|2\rangle$ にはならないが、 q_1 と q_2 は X_+ によって $|2\rangle$ にいる時間があることから、 q_0 と q_3 よりも q_1 と q_2 のデコヒーレンスによる影響は大きいことが予想される。以上から図 5.2.18 の電荷分散対策ありのプロットで見られる忠実度低下はデコヒーレンスの効果が見えていると考えられ、このデコヒーレンス効果が予想通りに見られるほどに他の ZZ 相互作用、local Rz、電荷分散を含むエラーが解消できていると考察した。しかしこの実験はデコヒーレンスを確かめる実験ではなく、この考察を確かめるには追加実験をする必要があるので、ここでは断定はせずに最も可能性の高い仮説として留めておく。

5.2.4 local Rz の原因考察

ここでは、local Rz の原因についての考察を記していく。5.2.2 節で、local Rz はパルスの有無に依らず、常に時間に依存した位相エラーを起こすことから、 H_{stat} による影響だという考察を行なった。また 2.2.2 節で説明したように、量子ビットの時間発展は有効ハミルトニアンによるものであり、どのような回転系を選択して有効ハミルトニアンを定めるかで量子演算の結果は異なる。

これらの事実を踏まえて、 H_{stat} から特定の回転系 U を使用して有効ハミルトニアンに変換する際に、 λ_3 の効果が残るような回転系を選択しているという仮説を立てた。そこで参考文献 [35] でいくつかの種類の回転系が説明されている中で qudit frame と dressed frame という二つの回転系に注目した。なぜな

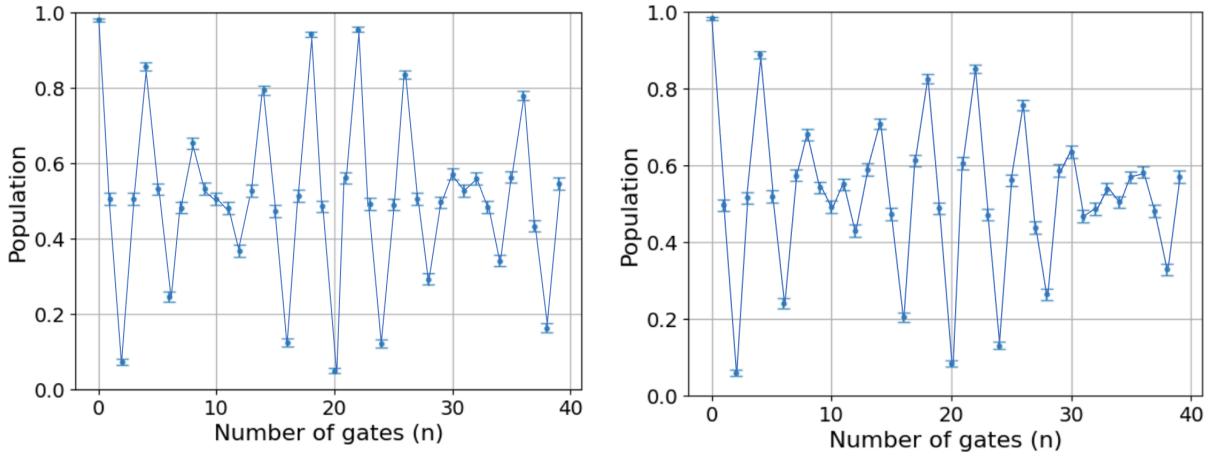


図 5.2.13: q_1 (左) と q_2 (右) の $|1\rangle - |2\rangle$ 間の Ramsey 実験の結果である $P^{(c)}(1)$ のプロット。縦軸は $P^{(c)}(1)$ (Population of $|1\rangle$)、横軸は delay ゲートの数 (n) である。予想される $P^{(c)}(1)$ の概形である図 5.2.12 の通り、複数の cos 波が合成されたような概形となっている。

ら本研究では 2.2.2 節から qudit frame という回転系を前提に有効ハミルトニアンの ZZ 相互作用等の計算をしていたが、この節では IBMQuantum で採用されている回転系は dressed frame である可能性が高い、という考察をするからである。したがって、まずは各 frame についての説明と性質を説明して、dressed frame の場合に H_{stat} はどのような有効ハミルトニアンになるのかを説明した後、実験結果を照らし合わせて考察を行うことにする。

まず qudit frame についてだが、これはまさに 2.2.2 節で紹介した回転系 $U_q(t) = e^{iH_0 t}$ に対応している。この回転系はトランズモン結合がないような系、つまり $H_{\text{int}} = 0$ の場合には $U_q H_{\text{stat}} U_q^\dagger = 0$ であるため、 H_{stat} によるエラーが完全になくなるという性質をもつ回転系である。実際はトランズモン結合が存在するため、 \tilde{H}_{int} が ZZ 相互作用の形で現れていた。

次に dressed frame についてだが、これは特定のトランズモン q の回転系を設定するときに、 q と結合している全てのトランズモン状態が $|0\rangle$ のときの q の遷移周波数を使用して、ユニタリ変換 U_d を行なっている。したがって q と結合している全てのトランズモン状態が $|0\rangle$ であれば、 q に結合が存在しても $U_d H_{\text{stat}} U_d^\dagger = 0$ が成立し、 H_{stat} によるエラーが完全になくなるという性質をもつ回転系になっている。このユニタリ変換も $U_d(t) = e^{iD t}$ という形でかけるが、 D の定義は複雑であるため、詳細は付録 A.4 に記すことにして、ここでは dressed frame の性質と簡単な計算のみで local Rz が生じることを説明する。

この dressed frame が local Rz を起こす説明を具体的に 2 量子ビット系の行列計算で見てみる。まず qudit frame における $U_q H_{\text{stat}} U_q^\dagger = \tilde{H}_{\text{stat}}^{(q)} = \Delta(\sigma_z \otimes \sigma_z)$ によって引き起こされる時間発展を行列表示すると、次のようになる：

$$e^{-i\tilde{H}_{\text{stat}}^{(q)}t} = \begin{bmatrix} e^{-i\Delta t} & & & \\ & e^{i\Delta t} & & \\ & & e^{i\Delta t} & \\ & & & e^{-i\Delta t} \end{bmatrix} \quad (5.2.10)$$

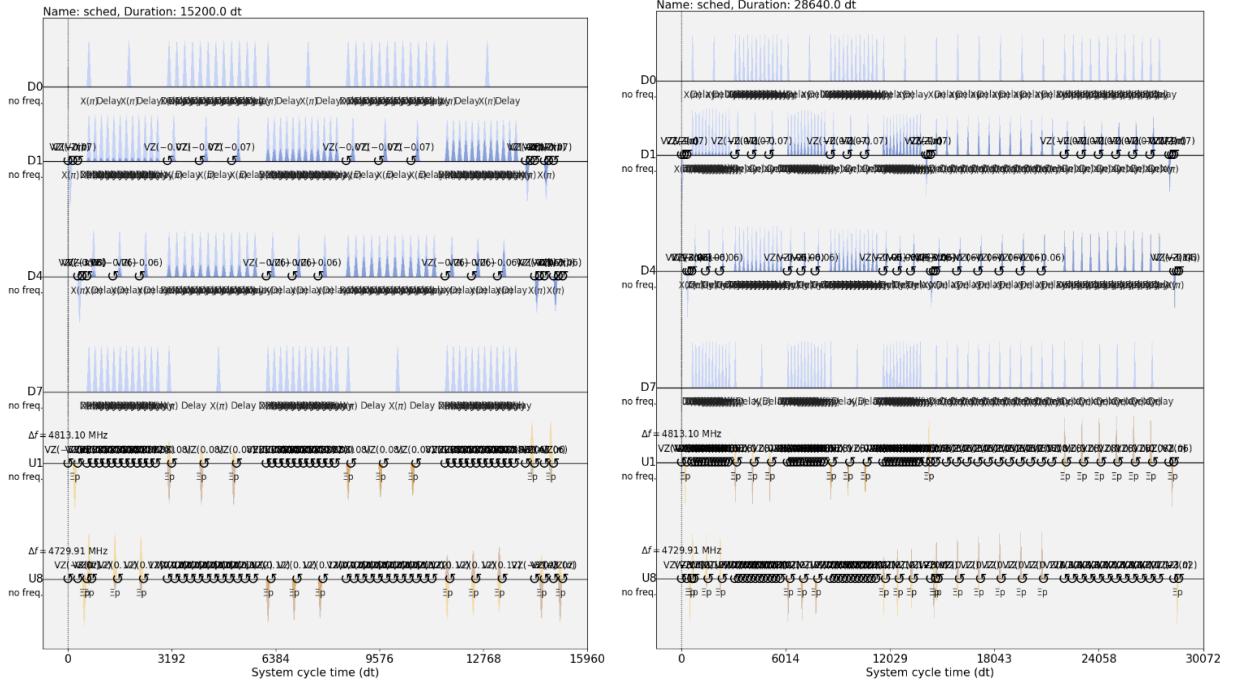


図 5.2.14: no CR CCCX で「ZZ 対策あり+local Rz 対策あり+電荷分散対策なし」(左) と「ZZ 対策あり+local Rz 対策あり+電荷分散対策あり」(右) のパルスシーケンス。電荷分散対策は 4.3 節で述べたようにシーケンスを延長するものなので、スケジュール時間は(左)が 3.38[μs]、(右)が 6.36[μs] である。

一方で dressed frame による $U_d H_{\text{stat}} U_d^\dagger = \tilde{H}_{\text{stat}}^{(d)}$ では、隣の結合している量子ビットが $|0\rangle$ のときに H_{stat} によるエラーがなくなるという性質から、 $|00\rangle$ 、 $|01\rangle$ 、 $|10\rangle$ にはエラーがなくなるように U_d をとるため、 $\tilde{H}_{\text{stat}}^{(d)}$ による時間発展は $|11\rangle$ に加わる位相エラーの周波数 ν を用いて次のような形をしているはずである：

$$e^{-i\tilde{H}_{\text{stat}}^{(d)}t} = \begin{bmatrix} 1 & & & \\ & 1 & & \\ & & 1 & \\ & & & e^{-i\nu t} \end{bmatrix} \quad (5.2.11)$$

そこでこの行列の形になるような $\tilde{H}_{\text{stat}}^{(d)}$ を、 $\tilde{H}_{\text{stat}}^{(q)}$ から逆算してみる。その結果、

$$\tilde{H}_{\text{stat}}^{(d)} = \tilde{H}_{\text{stat}}^{(q)} - \Delta(\sigma_z \otimes I + I \otimes \sigma_z) \quad (5.2.12)$$

という関係であれば、

$$e^{-i\tilde{H}_{\text{stat}}^{(d)}t} = e^{-i[\tilde{H}_{\text{stat}}^{(q)} - \Delta(\sigma_z \otimes I + I \otimes \sigma_z)]t} \quad (5.2.13)$$

$$= e^{-i\tilde{H}_{\text{stat}}^{(q)}t} e^{-i\Delta(\sigma_z \otimes I)t} e^{-i\Delta(I \otimes \sigma_z)t} \quad (5.2.14)$$

$$= \begin{bmatrix} e^{-i\Delta t} & & & \\ & e^{i\Delta t} & & \\ & & e^{i\Delta t} & \\ & & & e^{-i\Delta t} \end{bmatrix} \begin{bmatrix} e^{i\Delta t} & & & \\ & e^{i\Delta t} & & \\ & & e^{-i\Delta t} & \\ & & & e^{-i\Delta t} \end{bmatrix} \begin{bmatrix} e^{i\Delta t} & & & \\ & e^{-i\Delta t} & & \\ & & e^{i\Delta t} & \\ & & & e^{-i\Delta t} \end{bmatrix} \quad (5.2.15)$$

$$= e^{i\Delta t} \begin{bmatrix} 1 & & & \\ & 1 & & \\ & & 1 & \\ & & & e^{-i4\Delta t} \end{bmatrix} \quad (5.2.16)$$

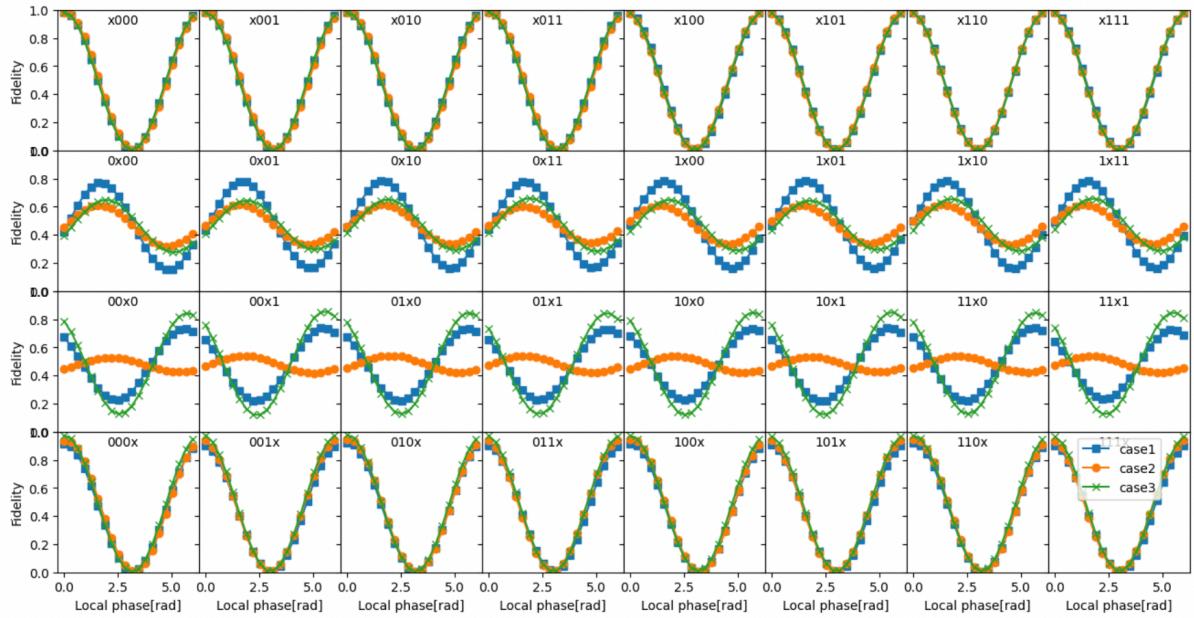


図 5.2.15: no CR CCCX で「ZZ 対策あり+local Rz 対策あり+電荷分散対策なし」の実験を行い、8通りの初期状態での $Rz(\phi)$ に対する忠実度のプロット。縦軸は忠実度、横軸は ϕ である。case1 が青、case2 がオレンジ、case3 が緑のプロットに対応している。

$$\sim \begin{bmatrix} 1 & & & \\ & 1 & & \\ & & 1 & \\ & & & e^{-i4\Delta t} \end{bmatrix} \quad (5.2.17)$$

となるため、 $\nu = 4\Delta$ で式 (5.2.11) の形を満たす。したがって、dressed frame という回転系による $\tilde{H}_{stat}^{(d)}$ には、ZZ 相互作用 $(\sigma_z \otimes \sigma_z)$ 、local Rz $(\sigma_z \otimes I, I \otimes \sigma_z)$ が含まれる事がわかる。ここで図 5.2.1 の実験であった状況のように、ZZ 対策設計として 1 つ目の量子ビットに X を $2n$ 回等間隔に実装した場合、 $\tilde{H}_{stat}^{(d)}$ 内の $(\sigma_z \otimes \sigma_z)$ と $(\sigma_z \otimes I)$ という生成子がキャンセルされ、ZZ 対策設計を施していないトランズモンのみに local Rz $(I \otimes \sigma_z)$ が生じる。

ここから、上の考察を IBMQuantum の実験結果に照らし合わせていく。dressed frame を採用するということは、他トランズモンが全て $|0\rangle$ のときの H_{stat} によるエラーがなくなる性質があると述べたが、実際に IBMQuantum で行なった実験を見ると、図 5.2.3、5.2.11 共に、ZZ 対策設計なしのプロット（黒）は QPT を行う量子ビットと結合している全ての量子ビットの状態が $|0\rangle$ のとき、最も忠実度が高い。（それぞれ $|x0\dots\rangle, |0x0\rangle, |\cdot 0x0\rangle, |\cdot \cdot 0x\rangle$ の忠実度が最も高かった。ドットに量子ビットは QPT 対称の量子ビット x と結合していないので、この状態は関係ない。）また図 5.2.3 の ZZ 相互作用を適用したプロット（橙）を見ると、ZZ 対策設計を施していない量子ビットのみに local Rz による位相エラーが見られている。よって dressed frame により「ZZ 対策設計を施していないトランズモンのみに local Rz $(I \otimes \sigma_z)$ が生じる」という上の考察と合致する。以上から、IBMQuantum では dressed frame を採用しているため local Rz が生じたという考察をした。

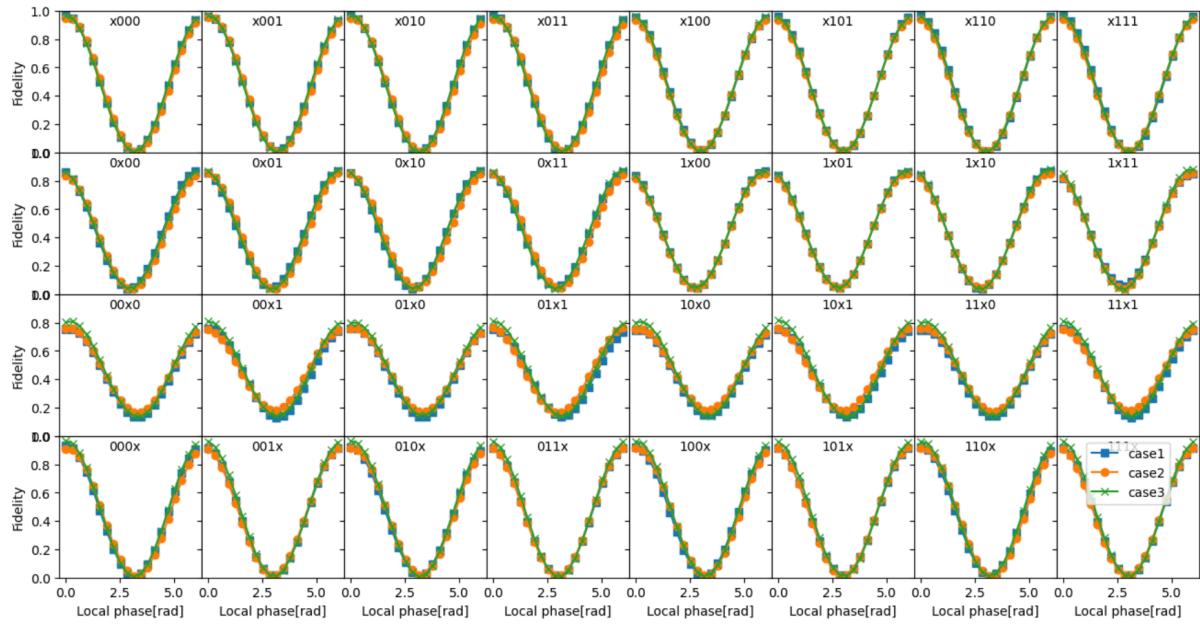


図 5.2.16: no CR CCCX で「ZZ 対策あり+local Rz 対策あり+電荷分散対策あり」の実験を行い、8通りの初期状態での $Rz(\phi)$ に対する忠実度のプロット。縦軸は忠実度、横軸は ϕ である。case1 が青、case2 がオレンジ、case3 が緑のプロットに対応している。

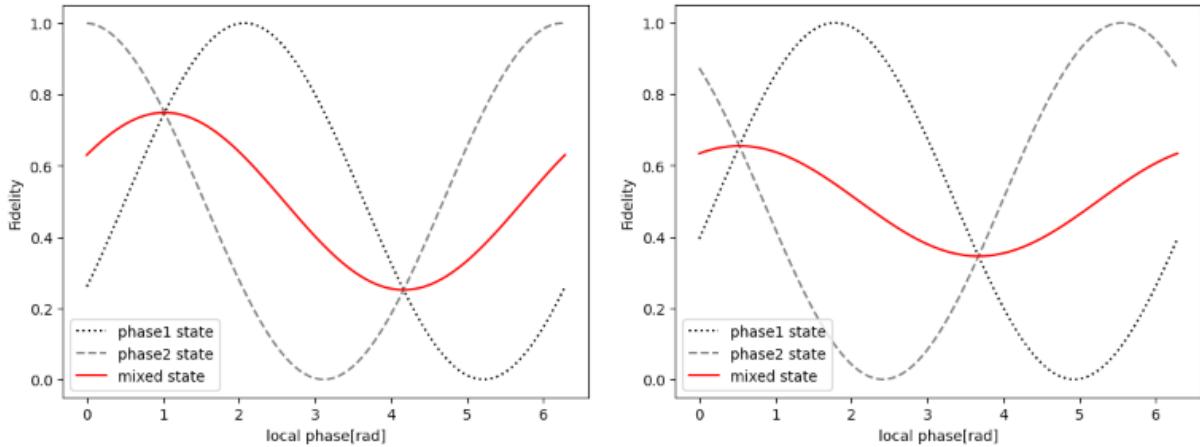


図 5.2.17: 図 5.2.15 で見られる sin 波の振幅減衰を説明する図。黒点線はそれぞれ $|2\rangle$ が ω_1 、 ω_2 のときの純粋状態の忠実度であり、赤線はそれらの混合状態の忠実度である。左右の図で赤線の振幅や位相が変化しているように、混合状態の忠実度は $\phi = \phi_1$ 、 $\phi = \phi_2$ に依存している。

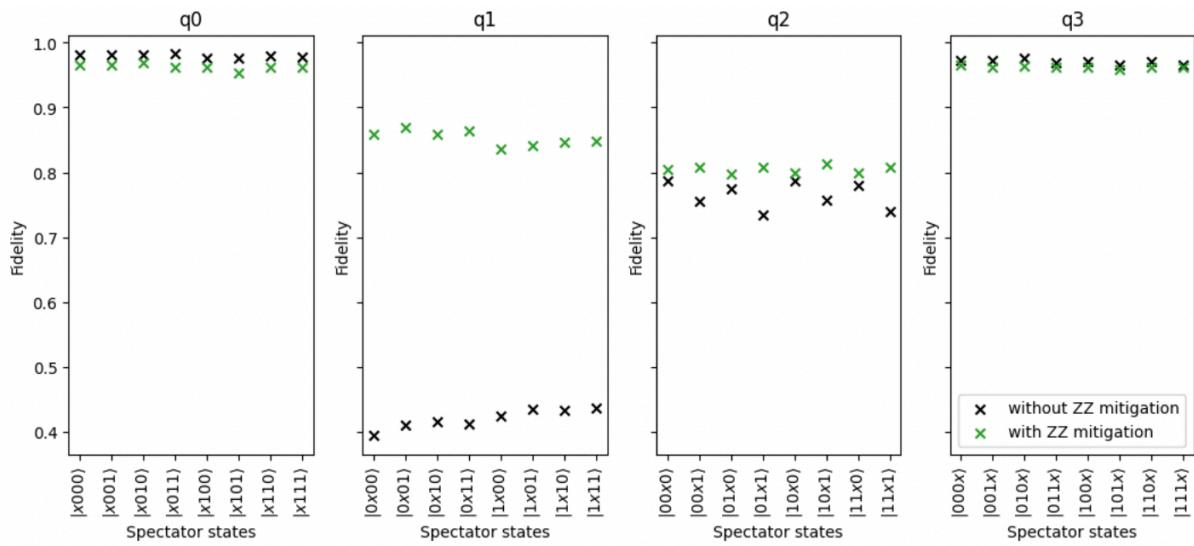


図 5.2.18: case3 の条件で電荷分散対策ありとなしのパルススケジュールを実装した q_0-q_3 において、 $|000\rangle - |111\rangle$ の 8 通りの初期状態でのアイデンティティに対する忠実度。黒が対策なし、緑が対策ありを表している。

第6章 結論

6.1 まとめ

本研究では、量子アルゴリズムで頻繁に使用される重要な多制御ビットゲートの精度向上を目指し、主要なエラー源である CNOT の使用数を削減するために、量子トリットを用いることを検討した。具体的には、量子ビットを使用した場合、 n 制御ビットゲートを実装するには $6n - 5$ 個の CNOT が必要だが、量子トリットを用いると $2n - 1$ 個で実装できることが示唆されていた。そして、量子トリットを用いて n 制御ビットゲートをハードウェアで実装する際に新たなエラー源となると予想されるベリー位相、ZZ 相互作用、電荷分散に対処するため、新しい対策方法をパルスレベルで設計した。ベリー位相に対しては、随伴行列となる量子トリットゲートのパルスシーケンスを確立し、また Virtual Z Gate という技術を活用してベリー位相を解消した。ZZ 相互作用に対しては、動的デカップリングなどを利用して位相エラーが自動的にキャンセルされる仕組みを構築した。電荷分散に対しては、補正用位相を活用して位相エラーを解消した。これらのエラー対策設計を組み込んだ新しい n 制御ビットゲートのパルスシーケンスを提案した。

そしてこれらのエラー対策設計が機能するかを確認するため、qudit-sim と IBMQuantum を用いて実証した。実証には、新規で考案した対策設計が全て組み込まれている n 制御ビットゲートの中で、最小の系である CCCX を使用した。

qudit-sim では、ベリー位相と ZZ 相互作用への対策設計の実証を行なった。ベリー位相対策設計は、導入前と導入後で忠実度が 0.1% しか変化せず、ゲートの精度を下げることなくベリー位相を取り除けることが確認できた。また ZZ 相互作用対策設計を用いた CCCX のゲート忠実度は、対策前と比較して 28% 向上し、0.9028 を達成した。

IBMQuantum では、QPT による忠実度測定を短時間で可能にするため、CCCX から CR を抜いたゲートシーケンスを利用して、電荷分散への対策設計の実証を行なった。実機で電荷分散対策が有効であることを示すためには、ZZ 相互作用の位相エラーが含まれていないことが前提となっていたため、まずは実機で ZZ 相互作用の対策設計が機能しているかを簡単にテストした。テストの結果、ZZ 相互作用対策は機能はしていたが、演算時間に比例する位相エラー (local Rz) が新たに発見された。そこで local Rz に対して、対策設計を開発した。最後に local Rz と ZZ 相互作用を対策して、主なエラー要因を電荷分散に絞った上で、電荷分散への対策を施したシーケンスの忠実度を測定し、位相エラーが解消されていることが確認できた。

以上の結果からベリー位相、ZZ 相互作用、電荷分散という 3 種類のエラーを対策する設計が有効であることが実証できた。

6.2 今後の展望

本研究では、対称としていた3種類のエラーを解消する設計が開発できたが、量子トリットを用いた n 制御ビットゲートの実用性を示すために、検証や改善をする必要がある点が残っている。

まずは実機において、CCCX を含め量子トリットを使用した場合の n 制御ビットゲートの忠実度を、量子ビットを使用した場合の忠実度と比較することである。この結果は量子トリットを用いることに優位性があるか否かが明確になるので、実験するべきであったが時間が足りず間に合わなかったことに加え、実機の占有時間が限られているため、QPTによる忠実度測定も困難であった。IBMQQuantumにおいて、デフォルトの CCCX の実装方法では $15.3[\mu\text{s}]$ かかるのに対し、本研究で作成した量子トリットを使用した CCCX の実装方法では CNOT 数を大幅に削減したことにより、 $6.36[\mu\text{s}]$ で実装できることがわかった。qudit-sim ではデコヒーレンスの効果が入っていないため、実機で実装することでこの演算時間短縮による精度差も含めて比較したい。

次に local Rz 対策を n 制御ビットゲートへ組み込むことである。本研究で開発した local Rz 対策は、電荷分散対策設計を評価するための応急措置であったため、本来 CR を実装する領域に別のゲートを適用してしまっている。local Rz は ZZ 相互作用と同程度の位相エラーを起こすことから、 n 制御ビットゲートの主なエラー要因となるため、CR を抜いた n 制御ビットゲートではなく、CR も含めた n 制御ビットゲートに対策設計を組み込む必要がある。

最後に CNOT_{trit} をハードウェアで実装する際の制約を解消することである。今回 5.1 節で CNOT_{trit}を作成する中で、作成ができないトランズモンペアがあることが判明した。理由は 2 個のトランズモンの量子ビット周波数や非調和性といったハードウェアパラメータの関係性によって、図 5.1.4 で c となる交点を持たないペアがあったからであった。CCCX では CNOT_{trit} を実装できるトランズモンペアが 2 連続であればよかったが、 n 制御ビットゲートでは $n - 1$ 連続で必要なので、 n が大きくなると実機で実装できない可能性が高くなる。汎用性を高めるためには、 $\nu_{zx} = -\nu_{\zeta x}$ という条件に依存しない CNOT_{trit} の実装方法を開発する必要があると考える。

これらの問題を解決して、量子トリットによる n 制御ビットゲートの実用性を高めることが今後の展望である。

第7章 謝辞

この論文の完成にあたり、多くの方々からのご支援とご協力をいただきました。心より感謝申し上げます。指導教員である寺師弘二准教授には、研究テーマの相談から始まり、さまざまな場面で迷った時には適切な方向性を示していただき、充実した修士生活を送ることができました。研究発表の機会には、スライドや発表への指摘もしていただき、大きく成長することができました。大変お世話になりました。飯山悠太郎助教には、GPUの使用方法やコードの作成、具体的なエラーへの対処など、コーディングに関する相談や技術的なパルス技術の質問に答えていただき、最終的に修論で成果を上げることができました。的確なアドバイスに感謝いたします。先輩の張元豪さんには、研究で詰まった際に幾度も相談に乗っていただき、その際の方針の提案や問題点の指摘は研究を進める上で欠かせないものでした。心から感謝しています。ミーティングで数多くの議論と意見交換をしていただいた方々にも感謝申し上げます。皆様の熱心な参加と示唆に富んだコメントは、私の研究に新たな視点をもたらしました。研究室の友人たちにも多くの感謝を捧げたいと思います。共に過ごした時間は私にとって宝物であり、皆の協力がなければこの論文は完成しなかったでしょう。お互いに刺激を受け、助け合いながら研究に取り組むことができたことに感謝しています。最後に、私の家族に心からの感謝を捧げます。彼らの絶え間ないサポートと信じる心が、私を支えてくれました。彼らのおかげで、私はこの研究を続ける力を得ることができました。以上の皆様に心より感謝申し上げます。本論文は多くの方々のご支援のもと、完成することができました。今後も精進し、研究者としての一歩を進んでいけるよう努力してまいります。どうもありがとうございました。

関連図書

- [1] Michael A Nielsen and Isaac L Chuang. *Quantum computation and quantum information*. Cambridge university press, 2010.
- [2] Peter W. Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM Review*, 41(2):303–332, 1999.
- [3] Lov K. Grover. A fast quantum mechanical algorithm for database search. In *Proceedings of the Twenty-Eighth Annual ACM Symposium on Theory of Computing*, STOC ’96, page 212–219, New York, NY, USA, 1996. Association for Computing Machinery.
- [4] Peter W Shor. Algorithms for quantum computation: discrete logarithms and factoring. In *Proceedings 35th annual symposium on foundations of computer science*, pages 124–134. Ieee, 1994.
- [5] Frank Arute, Kunal Arya, Ryan Babbush, Dave Bacon, Joseph C Bardin, Rami Barends, Rupak Biswas, Sergio Boixo, Fernando GSL Brandao, David A Buell, et al. Quantum supremacy using a programmable superconducting processor. *Nature*, 574(7779):505–510, 2019.
- [6] Philip Krantz, Morten Kjaergaard, Fei Yan, Terry P Orlando, Simon Gustavsson, and William D Oliver. A quantum engineer’s guide to superconducting qubits. *Applied physics reviews*, 6(2), 2019.
- [7] Fulvio Flamini, Nicolo Spagnolo, and Fabio Sciarrino. Photonic quantum information processing: a review. *Reports on Progress in Physics*, 82(1):016001, 2018.
- [8] Colin D Bruzewicz, John Chiaverini, Robert McConnell, and Jeremy M Sage. Trapped-ion quantum computing: Progress and challenges. *Applied Physics Reviews*, 6(2), 2019.
- [9] John Preskill. Quantum computing in the nisq era and beyond. *Quantum*, 2:79, 2018.
- [10] Andre He, Benjamin Nachman, Wibe A de Jong, and Christian W Bauer. Zero-noise extrapolation for quantum-gate error mitigation with identity insertions. *Physical Review A*, 102(1):012426, 2020.
- [11] Kristan Temme, Sergey Bravyi, and Jay M Gambetta. Error mitigation for short-depth quantum circuits. *Physical review letters*, 119(18):180509, 2017.

- [12] Seyon Sivarajah, Silas Dilkes, Alexander Cowtan, Will Simmons, Alec Edgington, and Ross Duncan. $t \rightarrow \text{ket}_t$: a retargetable compiler for nisq devices. *Quantum Science and Technology*, 6(1):014003, 2020.
- [13] Toshiaki Inada, Wonho Jang, Yutaro Iiyama, Koji Terashi, Ryu Sawada, Junichi Tanaka, and Shoji Asai. Measurement-free ultrafast quantum error correction by using multi-controlled gates in higher-dimensional state space, 2021.
- [14] Dmitri Maslov. Advantages of using relative-phase toffoli gates with an application to multiple control toffoli optimization. *Phys. Rev. A*, 93:022311, Feb 2016.
- [15] Casey Duckering, Jonathan M. Baker, Andrew Litteken, and Frederic T. Chong. Orchestrated trios: Compiling for efficient communication in quantum programs with 3-qubit gates. In *Proceedings of the 26th ACM International Conference on Architectural Support for Programming Languages and Operating Systems*, ASPLOS '21, page 375–385, New York, NY, USA, 2021. Association for Computing Machinery.
- [16] IBMQuantum. <https://quantum.ibm.com/>.
- [17] Pranav Gokhale, Jonathan M Baker, Casey Duckering, Natalie C Brown, Kenneth R Brown, and Frederic T Chong. Asymptotic improvements to quantum circuits via qutrits. In *Proceedings of the 46th International Symposium on Computer Architecture*, pages 554–566, 2019.
- [18] Yushi Wang and Marek Perkowski. Improved complexity of quantum oracles for ternary grover algorithm for graph coloring. In *2011 41st IEEE International Symposium on Multiple-Valued Logic*, pages 294–301, 2011.
- [19] Ji Chu, Xiaoyu He, Yuxuan Zhou, Jiahao Yuan, Libo Zhang, Qihao Guo, Yongju Hai, Zhikun Han, Chang-Kang Hu, Wenhui Huang, Hao Jia, Dawei Jiao, Sai Li, Yang Liu, Zhongchu Ni, Lifu Nie, Xianchuang Pan, Jiawei Qiu, Weiwei Wei, Wuerkaixi Nuerbolati, Zusheng Yang, Jiajian Zhang, Zhida Zhang, Wanjing Zou, Yuanzhen Chen, Xiaowei Deng, Xiuhao Deng, Ling Hu, Jian Li, Song Liu, Yao Lu, Jingjing Niu, Dian Tan, Yuan Xu, Tongxing Yan, Youpeng Zhong, Fei Yan, Xiaoming Sun, and Dapeng Yu. Scalable algorithm simplification using quantum AND logic. *Nature Physics*, 19(1):126–131, nov 2022.
- [20] Yuchen Wang, Zixuan Hu, Barry C Sanders, and Sabre Kais. Qudits and high-dimensional quantum computing. *Frontiers in Physics*, 8:589504, 2020.
- [21] AS Nikolaeva, EO Kiktenko, and AK Fedorov. Decomposing the generalized toffoli gate with qutrits. *Physical Review A*, 105(3):032621, 2022.
- [22] Pranav Gokhale, Ali Javadi-Abhari, Nathan Earnest, Yunong Shi, and Frederic T Chong. Optimized quantum compilation for near-term algorithms with openpulse. In *2020 53rd Annual*

IEEE/ACM International Symposium on Microarchitecture (MICRO), pages 186–200. IEEE, 2020.

- [23] qiskit.pulse. <https://docs.quantum.ibm.com/api/qiskit/pulse>.
- [24] Thomas Alexander, Naoki Kanazawa, Daniel J Egger, Lauren Capelluto, Christopher J Wood, Ali Javadi-Abhari, and David C McKay. Qiskit pulse: programming quantum computers through the cloud with pulses. *Quantum Science and Technology*, 5(4):044006, 2020.
- [25] Shelly Garion, Naoki Kanazawa, Haggai Landa, David C McKay, Sarah Sheldon, Andrew W Cross, and Christopher J Wood. Experimental implementation of non-clifford interleaved randomized benchmarking with a controlled-s gate. *Physical Review Research*, 3(1):013204, 2021.
- [26] Nathan Earnest, Caroline Tornow, and Daniel J Egger. Pulse-efficient circuit transpilation for quantum applications on cross-resonance-based hardware. *Physical Review Research*, 3(4):043088, 2021.
- [27] Wonho Jang, Yutaro Iiyama, Naoki Kanazawa, Toshiaki Inada, Ryu Sawada, Tamiya Onodera, and Koji Terashi. Stable toffoli gate on fixed-frequency superconducting qutrits. In *2022 IEEE International Conference on Quantum Computing and Engineering (QCE)*, pages 869–872, 2022.
- [28] Andrei B Klimov, Luis L Sánchez-Soto, Hubert de Guise, and Gunnar Björk. Quantum phases of a qutrit. *Journal of Physics A: Mathematical and General*, 37(13):4097, mar 2004.
- [29] Yang Liu, Gui Lu Long, and Yang Sun. Analytic constructions of general n-qubit controlled gates. *arXiv preprint arXiv:0708.3274*, 2007.
- [30] Dmitri Maslov. Advantages of using relative-phase toffoli gates with an application to multiple control toffoli optimization. *Physical Review A*, 93(2):022311, 2016.
- [31] Adriano Barenco, Charles H Bennett, Richard Cleve, David P DiVincenzo, Norman Margolus, Peter Shor, Tycho Sleator, John A Smolin, and Harald Weinfurter. Elementary gates for quantum computation. *Physical review A*, 52(5):3457, 1995.
- [32] Alexis Morvan, VV Ramasesh, MS Blok, JM Kreikebaum, K O ’ Brien, L Chen, BK Mitchell, RK Naik, DI Santiago, and I Siddiqi. Qutrit randomized benchmarking. *Physical review letters*, 126(21):210504, 2021.
- [33] Yao-Min Di and Hai-Rui Wei. Elementary gates for ternary quantum logic circuit. *arXiv preprint arXiv:1105.5485*, 2011.
- [34] 野口 篤史 長田 有登, 山崎 歴舟. 量子技術序論, March 2021. https://www.sqeic.u-tokyo.ac.jp/qed/QEd_textbook.pdf.
- [35] qudit-sim. <https://qudit-sim.readthedocs.io>.

- [36] Jens Koch, M Yu Terri, Jay Gambetta, Andrew A Houck, David I Schuster, Johannes Majer, Alexandre Blais, Michel H Devoret, Steven M Girvin, and Robert J Schoelkopf. Charge-insensitive qubit design derived from the cooper pair box. *Physical Review A*, 76(4):042319, 2007.
- [37] David C McKay, Christopher J Wood, Sarah Sheldon, Jerry M Chow, and Jay M Gambetta. Efficient z gates for quantum computing. *Physical Review A*, 96(2):022330, 2017.
- [38] Easwar Magesan and Jay M Gambetta. Effective hamiltonian models of the cross-resonance gate. *Physical Review A*, 101(5):052308, 2020.
- [39] Sarah Sheldon, Easwar Magesan, Jerry M Chow, and Jay M Gambetta. Procedure for systematically tuning up cross-talk in the cross-resonance gate. *Physical Review A*, 93(6):060302, 2016.
- [40] Petar Jurcevic, Ali Javadi-Abhari, Lev S Bishop, Isaac Lauer, Daniela F Bogorin, Markus Brink, Lauren Capelluto, Oktay Günlük, Toshinari Itoko, Naoki Kanazawa, et al. Demonstration of quantum volume 64 on a superconducting quantum computing system. *Quantum Science and Technology*, 6(2):025020, 2021.
- [41] Abhinav Kandala, Ken X Wei, Srikanth Srinivasan, Easwar Magesan, S Carnevale, GA Keefe, D Klaus, O Dial, and DC McKay. Demonstration of a high-fidelity cnot gate for fixed-frequency transmons with engineered z z suppression. *Physical Review Letters*, 127(13):130501, 2021.
- [42] KX Wei, E Magesan, I Lauer, S Srinivasan, DF Bogorin, S Carnevale, GA Keefe, Y Kim, D Klaus, W Landers, et al. Quantum crosstalk cancellation for fast entangling gates and improved multi-qubit performance. *arXiv preprint arXiv:2106.00675*, 2021.
- [43] Yushi Wang and Marek Perkowski. Improved complexity of quantum oracles for ternary grover algorithm for graph coloring. In *2011 41st IEEE International Symposium on Multiple-Valued Logic*, pages 294–301. IEEE, 2011.
- [44] Di Xiao, Ming-Che Chang, and Qian Niu. Berry phase effects on electronic properties. *Reviews of modern physics*, 82(3):1959, 2010.
- [45] Machiel S Blok, Vinay V Ramasesh, Thomas Schuster, Kevin O ’ Brien, John-Mark Kreikebaum, Dar Dahlen, Alexis Morvan, Beni Yoshida, Norman Y Yao, and Irfan Siddiqi. Quantum information scrambling on a superconducting qutrit processor. *Physical Review X*, 11(2):021010, 2021.
- [46] Jaseung Ku, Xuexin Xu, Markus Brink, David C. McKay, Jared B. Hertzberg, Mohammad H. Ansari, and B. L. T. Plourde. Suppression of unwanted zz interactions in a hybrid two-qubit system. *Phys. Rev. Lett.*, 125:200504, Nov 2020.
- [47] Moein Malekakhlagh, Easwar Magesan, and David C McKay. First-principles analysis of cross-resonance gate operation. *Physical Review A*, 102(4):042605, 2020.

- [48] Jens Koch, M Yu Terri, Jay Gambetta, Andrew A Houck, David I Schuster, Johannes Majer, Alexandre Blais, Michel H Devoret, Steven M Girvin, and Robert J Schoelkopf. Charge-insensitive qubit design derived from the cooper pair box. *Physical Review A*, 76(4):042319, 2007.
- [49] Vinay Tripathi, Huo Chen, Mostafa Khezri, Ka-Wa Yip, E.M. Levenson-Falk, and Daniel A. Lidar. Suppression of crosstalk in superconducting qubits using dynamical decoupling. *Phys. Rev. Appl.*, 18:024068, Aug 2022.
- [50] Qiskit Textbook. Calibrating qubits with qiskit pulse, October 2021. <https://qiskit.org/textbook/ch-quantum-hardware/calibrating-qubits-pulse.html>.

付録A 付録

A.1 Qutrit echo cross-resonance CNOT の $ZX(\pi/3)\zeta X(-\pi/3)$ の計算

$$(X_+ \otimes I) \text{CR}^+(\pi/3) (X_+ \otimes I) \text{CR}^+(\pi/3) (X_+ \otimes I) \text{CR}^-(\pi/3) = ZX(\pi/3)\zeta X(-\pi/3) \quad (\text{A.1.1})$$

の計算を示す。式 (2.2.37) の結果から、

$$ZX(\pm\pi/3) \zeta X(\mp\pi/3) = \begin{bmatrix} Rx_{01}(\pm\pi/3) & 0 & 0 \\ 0 & Rx_{01}(\mp 2\pi/3) & 0 \\ 0 & 0 & Rx_{01}(\pm\pi/3) \end{bmatrix} \quad (\text{A.1.2})$$

である。これより、式 (2.2.32) の行列表現から、

$$\begin{aligned} \text{CR}^\pm(\pi/3) &= [Rz(\phi_0)R\zeta(\phi_1) \otimes I][ZX(\pm\pi/3) \zeta X(\mp\pi/3)] \\ &= \begin{bmatrix} e^{-i\frac{\phi_0}{2}} Rx_{01}(\pm\pi/3) & 0 & 0 \\ 0 & e^{i\frac{\phi_0-\phi_1}{2}} Rx_{01}(\mp 2\pi/3) & 0 \\ 0 & 0 & e^{i\frac{\phi_1}{2}} Rx_{01}(\pm\pi/3) \end{bmatrix} \end{aligned} \quad (\text{A.1.3})$$

となる。また $(X_+ \otimes I) \text{CR}^\pm(\pi/3)$ は

$$\begin{aligned} (X_+ \otimes I) \text{CR}^\pm(\pi/3) &= \begin{bmatrix} 0 & 0 & -ie^{i\frac{\phi_1}{2}} Rx_{01}(\pm\pi/3) \\ e^{-i\frac{\phi_0}{2}} Rx_{01}(\pm\pi/3) & 0 & 0 \\ 0 & e^{i\frac{\phi_0-\phi_1}{2}} Rx_{01}(\mp 2\pi/3) & 0 \end{bmatrix} \end{aligned} \quad (\text{A.1.4})$$

である。よって式 (A.1.1) の行列計算は、

$$\begin{aligned}
& (X_+ \otimes I) \text{CR}^-(\pi/3) (X_+ \otimes I) \text{CR}^-(\pi/3) (X_+ \otimes I) \text{CR}^+(\pi/3) \\
&= (X_+ \otimes I) \text{CR}^-(\pi/3) (X_+ \otimes I) \text{CR}^-(\pi/3) \\
&\quad \begin{bmatrix} 0 & 0 & -ie^{i\frac{\phi_1}{2}} Rx_{01}(\pi/3) \\ e^{-i\frac{\phi_0}{2}} Rx_{01}(\pi/3) & 0 & 0 \\ 0 & e^{i\frac{\phi_0-\phi_1}{2}} Rx_{01}(-2\pi/3) & 0 \end{bmatrix} \\
&= (X_+ \otimes I) \text{CR}^-(\pi/3) \\
&\quad \begin{bmatrix} 0 & e^{-i\frac{-\phi_0}{2}} Rx_{01}(-\pi) & 0 \\ 0 & 0 & e^{-i\frac{\phi_0-\phi_1}{2}} Rx_{01}(0) \\ e^{-i\frac{\phi_1}{2}} Rx_{01}(\pi) & 0 & 0 \end{bmatrix} \tag{A.1.5} \\
&= \begin{bmatrix} -iRx_{01}(2\pi/3) & 0 & 0 \\ 0 & -iRx_{01}(-4\pi/3) & 0 \\ 0 & 0 & -iRx_{01}(2\pi/3) \end{bmatrix} \\
&= ZX(\pi/3) \zeta X(-\pi/3)
\end{aligned}$$

となる。

A.2 量子トリット-量子トリット間の ZZ 相互作用の解消計算

$$(X_+ \otimes I) e^{-i(\tilde{H}_{\text{stat}})\alpha} (X_+ \otimes I) e^{-i(\tilde{H}_{\text{stat}})\alpha} (X_+ \otimes I) e^{-i(\tilde{H}_{\text{stat}})\alpha} \sim I \otimes I \tag{A.2.1}$$

の計算を示す。ここで

$$\tilde{H}_{\text{stat}} = \nu_{33} (\lambda_3 \otimes \lambda_3) + \nu_{38} (\lambda_3 \otimes \lambda_8) + \nu_{83} (\lambda_8 \otimes \lambda_3) + \nu_{88} (\lambda_8 \otimes \lambda_8) \tag{A.2.2}$$

と表現していたが、このゲルマン行列の表現だと計算が複雑になるため、 z と ζ に基底変換した次のようなハミルトニアンで考える：

$$\tilde{H}_{\text{stat}} = \nu_{zz} (z \otimes z) + \nu_{z\zeta} (z \otimes \zeta) + \nu_{\zeta z} (\zeta \otimes z) + \nu_{\zeta\zeta} (\zeta \otimes \zeta) \tag{A.2.3}$$

このハミルトニアンによる α 分の時間発展は、次のように算出できる：

$$\begin{aligned}
& e^{-i\tilde{H}_{\text{stat}}\alpha} \\
&= e^{-i\nu_{zz} (z \otimes z)\alpha} e^{-i\nu_{z\zeta} (z \otimes \zeta)\alpha} e^{-i\nu_{\zeta z} (\zeta \otimes z)\alpha} e^{-i\nu_{\zeta\zeta} (\zeta \otimes \zeta)\alpha} \\
&= \begin{bmatrix} Rz(\nu_{zz}\alpha) & & & \\ & Rz(-\nu_{zz}\alpha) & & \\ & & I & \end{bmatrix} \begin{bmatrix} R\zeta(\nu_{z\zeta}\alpha) & & & \\ & R\zeta(-\nu_{z\zeta}\alpha) & & \\ & & I & \end{bmatrix} \begin{bmatrix} I & & & \\ & Rz(\nu_{\zeta z}\alpha) & & \\ & & Rz(-\nu_{\zeta z}\alpha) & \\ & & & I \end{bmatrix} \begin{bmatrix} I & & & \\ & R\zeta(\nu_{\zeta\zeta}\alpha) & & \\ & & R\zeta(-\nu_{\zeta\zeta}\alpha) & \\ & & & I \end{bmatrix} \\
&= \begin{bmatrix} Rz(\nu_{zz}\alpha)R\zeta(\nu_{z\zeta}\alpha) & & & \\ & Rz(-\nu_{zz}\alpha)R\zeta(-\nu_{z\zeta}\alpha)Rz(\nu_{\zeta z}\alpha)R\zeta(\nu_{\zeta z}\alpha) & & \\ & & Rz(-\nu_{\zeta z}\alpha)R\zeta(-\nu_{\zeta z}\alpha) & \\ & & & Rz(\nu_{\zeta\zeta}\alpha)R\zeta(\nu_{\zeta\zeta}\alpha) \end{bmatrix} \\
&= \begin{bmatrix} Rz(\nu_{zz}\alpha)R\zeta(\nu_{z\zeta}\alpha) & & & \\ & Rz((\nu_{\zeta z} - \nu_{zz})\alpha)R\zeta((\nu_{\zeta\zeta} - \nu_{z\zeta})\alpha) & & \\ & & Rz(-\nu_{\zeta z}\alpha)R\zeta(-\nu_{\zeta\zeta}\alpha) & \end{bmatrix} \tag{A.2.4}
\end{aligned}$$

また $(X_+ \otimes I)e^{-i\tilde{H}_{\text{stat}}\alpha}$ は先程の結果を利用して、次のようになる：

$$(X_+ \otimes I)e^{-i\tilde{H}_{\text{stat}}\alpha} = \begin{bmatrix} & -iRz(-\nu_{\zeta z}\alpha)R\zeta(-\nu_{\zeta \zeta}\alpha) \\ Rz(\nu_{zz}\alpha)R\zeta(\nu_{z\zeta}\alpha) & \\ & Rz((\nu_{\zeta z} - \nu_{zz})\alpha)R\zeta((\nu_{\zeta \zeta} - \nu_{z\zeta})\alpha) \end{bmatrix} \quad (\text{A.2.5})$$

これより式 (A.2.1) の左辺は次のようになる：

$$\begin{aligned} & (X_+ \otimes I)e^{-i\tilde{H}_{\text{stat}}\alpha}(X_+ \otimes I)e^{-i\tilde{H}_{\text{stat}}\alpha}(X_+ \otimes I)e^{-i\tilde{H}_{\text{stat}}\alpha} \\ &= (X_+ \otimes I)e^{-i\tilde{H}_{\text{stat}}\alpha}(X_+ \otimes I)e^{-i\tilde{H}_{\text{stat}}\alpha} \\ &\quad \begin{bmatrix} & -iRz(-\nu_{\zeta z}\alpha)R\zeta(-\nu_{\zeta \zeta}\alpha) \\ Rz(\nu_{zz}\alpha)R\zeta(\nu_{z\zeta}\alpha) & \\ & Rz((\nu_{\zeta z} - \nu_{zz})\alpha)R\zeta((\nu_{\zeta \zeta} - \nu_{z\zeta})\alpha) \end{bmatrix} \\ &= (X_+ \otimes I)e^{-i\tilde{H}_{\text{stat}}\alpha} \\ &\quad \begin{bmatrix} & -iRz(-\nu_{zz}\alpha)R\zeta(-\nu_{z\zeta}\alpha) \\ & -iRz((\nu_{zz} - \nu_{\zeta z})\alpha)R\zeta((\nu_{z\zeta} - \nu_{\zeta \zeta})\alpha) \\ Rz(\nu_{\zeta z}\alpha)R\zeta(\nu_{\zeta \zeta}\alpha) \end{bmatrix} \\ &= \begin{bmatrix} -iI & & \\ & -iI & \\ & & -iI \end{bmatrix} \\ &\sim I \otimes I \end{aligned} \quad (\text{A.2.6})$$

以上から、式 (A.2.1) が成立する。

A.3 補正用位相の算出

n 制御ビットゲートの補正用位相が式 (4.3.8) と算出される過程を説明する。まず式 (4.3.9) から等間隔で X_+ を 3 回実装している間は電荷分散による位相が蓄積しないことがわかっている。あとは CNOT_{bit}、CNOT_{trit} の制御トリットと標的トリットで、電荷分散による位相エラーがどのように発生するかが理解できれば、式 (4.3.8) が算出できる。

電荷分散は式 (4.3.1) で説明した通り、 $|1\rangle - |2\rangle$ 遷移周波数に対し、時間 t 分のドライブ周波数の非共鳴効果 δ により位相エラーを引き起こし、 X_{12} のような $|2\rangle$ の状態を遷移させるゲートを適用すると、位相エラーが生じてしまう $e^{-i\zeta\delta t}$ の形で表面化する、というものであった。

ここで CR_{bit}、CR_{trit} の有効ハミルトニアンの形を見ると、

$$\tilde{H}_{\text{CRbit}}^{\pm} = \nu_{zI} \frac{\sigma_z I}{2} \mp \nu_{Ix} \frac{I \sigma_x}{2} \pm \nu_{zx} \frac{\sigma_z \sigma_x}{2} \quad (\text{A.3.1})$$

$$\tilde{H}_{\text{CRtrit}}^{\pm} = \nu_{zI} \frac{zI}{2} \pm \nu_{zx} \frac{zx}{2} + \nu_{\zeta I} \frac{\zeta I}{2} \pm \nu_{\zeta x} \frac{\zeta x}{2} \quad (\text{A.3.2})$$

であり、制御トリットには I, z, ζ という生成子しか存在せず、標的トリットには I, x という生成子しか存在しない。したがって電荷分散より蓄積した位相エラーが CR によって表面化することなく、CNOT 中の位相エラーが生じるゲートは CNOT_{trit} シーケンス中の制御トリットに存在する X_{12} のみである。

つぎに CNOT_{trit} シーケンス中の制御トリットの X_{12} で発生する位相エラーの合計を求める。そのためにはまず CR_{trit} と位相エラーとの交換関係を求める。 $z_0 = z_3 = \text{diag}(1, -1, 0) = z$ 、 $z_1 = \text{diag}(0, 1, -1) = \zeta$ 、 $z_2 = z_{-1} = \text{diag}(-1, 0, 1) = \lambda$ を用いて、交換関係は、

$$\begin{aligned} & [e^{-i(z_n I)\delta t}, e^{-iH_{\text{CR}}^{\pm} t}] \\ &= [e^{-i(z_n I)\delta t}, e^{-i(\nu_{zI} \frac{zI}{2})t}] + [e^{-i(z_n I)\delta t}, e^{\mp i(\nu_{zx} \frac{zx}{2})t}] + [e^{-i(z_n I)\delta t}, e^{-i(\nu_{\zeta I} \frac{\zeta I}{2})t}] + [e^{-i(z_n I)\delta t}, e^{\mp i(\nu_{\zeta x} \frac{\zeta x}{2})t}] \\ &= 0 \end{aligned} \quad (\text{A.3.3})$$

と計算できる。つまり電荷分散による位相エラーと CR_{trit} は可換である。この結果を利用すれば、CNOT_{trit} の制御トリットで発生する位相エラーの合計は、CR のドライブ時間を α とおけば、次のように計算できる：

$$\begin{aligned} & (X_+^{(3\alpha)} \otimes I) \text{CR}^-(\pi/3) (X_+^{(2\alpha)} \otimes I) \text{CR}^-(\pi/3) (X_+^{(\alpha)} \otimes I) \text{CR}^+(\pi/3) \\ &= (X_+ e^{-i\zeta(3\alpha)} \otimes I) \text{CR}^-(\pi/3) (X_+ e^{-i\zeta(2\alpha)} \otimes I) \text{CR}^-(\pi/3) (X_+ e^{-i\zeta(\alpha)} \otimes I) \text{CR}^+(\pi/3) \quad (\because \text{式 (4.3.1)}) \\ &= e^{-i\lambda(3\alpha)} e^{-iz(2\alpha)} e^{-i\zeta(\alpha)} (X_+ \otimes I) \text{CR}^-(\pi/3) (X_+ \otimes I) \text{CR}^-(\pi/3) (X_+ \otimes I) \text{CR}^+(\pi/3) \\ &\quad (\because \text{式 (4.3.3), (A.3.3)}) \\ &= e^{iI\alpha} \text{CNOT}_{\text{trit}} \\ &= \text{CNOT}_{\text{trit}} \end{aligned} \quad (\text{A.3.4})$$

つまり CNOT_{trit} 中で蓄積する位相エラーは 0 である。以上を踏まえると、式 (4.3.9)(A.3.4) は局所的に電荷分散による位相エラーを解消できていることがわかる。式 (4.3.9)(A.3.4) はどちらも ZZ 相互作用への対策設計であるため、「ZZ 相互作用への対策設計を実装中の量子トリットでは、電荷分散による位相エラーが蓄積しない」と言い変えられる。

この結果を踏まえて図 4.2.4 を見ると、一番最初の X_- が実装されてから最後の X_-^\dagger が実装される時間の中で ZZ 対策設計が実装されていない時間のみが、実際に電荷分散による位相エラーが蓄積している時間であり、補正用位相のための必要な遅延 τ である。一番最初の X_- が実装されてから最後の X_-^\dagger が実装される時間 D_{mcx} は、図 4.2.4 の通りゲートシーケンスに含まれる全ての CNOT の実装時間の合計である。そこから ZZ 対策設計が組み込まれている CNOT_{trit} 時間分を引けば、各量子トリットに必要な τ が求まる。その結果、 (q_i, q_j) CNOT_{trit} のゲート実装時間を $d(q_i, q_j)$ 、 q_i に必要な遅延を τ_i と表記して、結果は次のように表現できる：

$$\begin{aligned} n = 2m + 1 \text{ のとき } & \left\{ \begin{array}{l} \tau_{2j+1} (0 \leq j < \frac{n-1}{2}) = 2 \left(\sum_{i=1}^{m-1} d(q_{2i}, q_{2i+1}) \right) + d(q_{2m}, q_{2m+1}) \\ \tau_{2j} (0 < j < \frac{n-1}{2}) = 2 \left(\sum_{i=1}^{m-1} d(q_{2i+1}, q_{2i+2}) \right) \end{array} \right. \\ n = 2m \text{ のとき } & \left\{ \begin{array}{l} \tau_{2j+1} (0 \leq j < \frac{n}{2}) = 2 \left(\sum_{i=1}^{m-1} d(q_{2i}, q_{2i+1}) \right) \\ \tau_{2j} (0 < j < \frac{n}{2}) = 2 \left(\sum_{i=1}^{m-1} d(q_{2i+1}, q_{2i+2}) \right) + d(q_{2m-1}, q_{2m}) \end{array} \right. \end{aligned} \quad (\text{A.3.5})$$

A.4 dressed frame

dressed frame を数式を使って説明を行う。この説明は文献 [35] を参考にしている。

qudit frame では \tilde{H}_{int} による ZZ 相互作用エラーが大きく、パルスをドライブしない時間であっても必ず時間発展が生じてしまう。そこで H_{int} を考慮に入れた回転系を定義するため、回転系へのユニタリ変換 U を定義し直す。

まず H_{stat} をユニタリー演算子 V で対角化する：

$$H_{stat} = VEV^\dagger \quad (\text{A.4.1})$$

ここで対角行列 E は対角行列 δ を用いて、 $E = H_0 + \delta$ と表現できる。この δ が qudit frame の有効ハミルトニアンにおいて、トランズモン結合により生じていた ZZ 相互作用である。

この E は H_{int} も含めた系の固有エネルギーであり、 n 個のトランズモン系の全てのエネルギー固有状態 $|l_0\rangle |l_1\rangle \cdots |l_n\rangle$ に対して、 $\langle l_0| \langle l_1| \cdots \langle l_n| E |l_0\rangle |l_1\rangle \cdots |l_n\rangle$ と固有エネルギーが求まる。したがってこの固有エネルギーを $E(l_0, l_1, \dots, l_n)$ と表記したとき、他トランズモン状態が $(l_0, l_1, \dots, l_{i-1}, l_{i+1}, \dots, l_n)$ のときの i 番目の $|n\rangle - |n+1\rangle$ 遷移周波数 $\omega_n^{(i)}$ は、次のようになる：

$$\omega_n^{(i)} = E(l_0, l_1, \dots, l_{i-1}, l_n = 1, l_{i+1}, \dots, l_n) - E(l_0, l_1, \dots, l_{i-1}, l_i = 0, l_{i+1}, \dots, l_n) \quad (\text{A.4.2})$$

このように任意の状態に応じた回転系にユニタリ変換できるような U を選べば、他トランズモン状態に依存した非共鳴効果はなくなり、 H_{stat} による位相エラーは解消される。

しかし実機で回転系を定める際は、各トランズモンに回転系へ変換するための周波数 $\omega_n^{(i)}$ を一つに定めてしまう。なぜなら、実際の物理系では途中で他のトランズモン状態を把握して動的に回転系への周波数を変化させることができないからである。そのためトランズモン毎に回転系としての遷移周波数 $\omega_n^{(i)}$ を一つ選択する。この周波数の選択方法として、「他トランズモンの状態が全て $|0\rangle$ である」ような遷移周波数を選んだものが、dressed frame であり、

$$D = \sum_{j=1}^n \sum_l (\langle l_j| \otimes \langle 0|^{\otimes n-1} E |l_j\rangle \otimes |0\rangle^{\otimes n-1}) |l_j\rangle \langle l_j| \quad (\text{A.4.3})$$

を用いてユニタリ変換 $U_d(t) = e^{iDt}$ と表される。