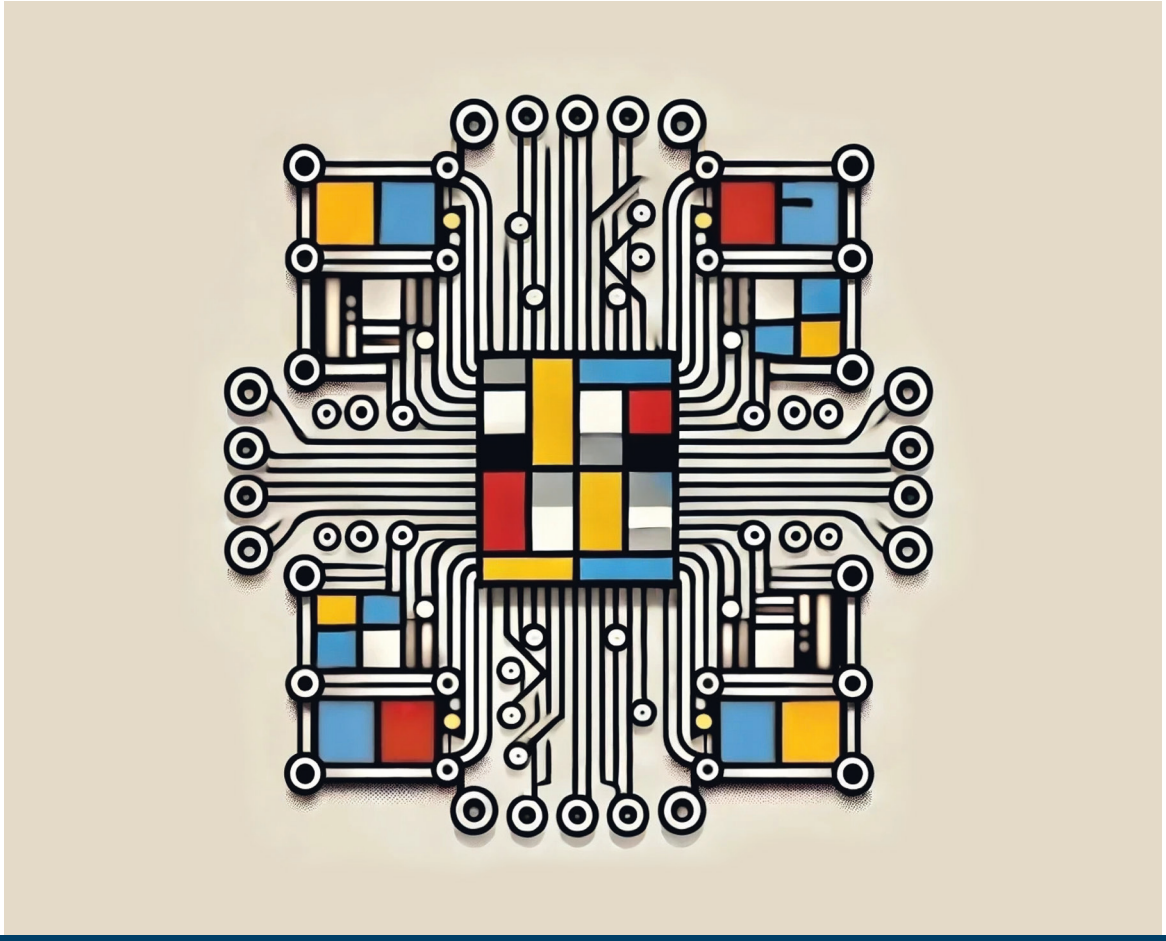


Practical Methods for Efficient Analytical Control in Superconducting Qubits

Boxi Li

Schlüsseltechnologien / Key Technologies
Band / Volume 290
ISBN 978-3-95806-807-0

Mitglied der Helmholtz-Gemeinschaft

JÜLICH
Forschungszentrum

Schriften des Forschungszentrums Jülich
Reihe Schlüsseltechnologien / Key Technologies Band / Volume 290

Forschungszentrum Jülich GmbH
Peter Grünberg Institut (PGI)
Quantum Control (PGI-8)

Practical Methods for Efficient Analytical Control in Superconducting Qubits

Boxi Li

Schriften des Forschungszentrums Jülich
Reihe Schlüsseltechnologien / Key Technologies Band / Volume 290

ISSN 1866-1807 ISBN 978-3-95806-807-0

Bibliografische Information der Deutschen Nationalbibliothek.
Die Deutsche Nationalbibliothek verzeichnet diese Publikation in der
Deutschen Nationalbibliografie; detaillierte Bibliografische Daten
sind im Internet über http://dnb.d-nb.de abrufbar.

Herausgeber und Vertrieb:	Forschungszentrum Jülich GmbH Zentralbibliothek, Verlag 52425 Jülich Tel.: +49 2461 61-5368 Fax: +49 2461 61-6103 zb-publikation@fz-juelich.de www.fz-juelich.de/zb
Umschlaggestaltung:	Grafische Medien, Forschungszentrum Jülich GmbH
Druck:	Grafische Medien, Forschungszentrum Jülich GmbH
Copyright:	Forschungszentrum Jülich 2025

Schriften des Forschungszentrums Jülich
Reihe Schlüsseltechnologien / Key Technologies, Band / Volume 290

D 38 (Diss. Köln, Univ., 2024)

ISSN 1866-1807
ISBN 978-3-95806-807-0

Vollständig frei verfügbar über das Publikationsportal des Forschungszentrums Jülich (JuSER)
unter www.fz-juelich.de/zb/openaccess.

This is an Open Access publication distributed under the terms of the Creative Commons Attribution License 4.0,
which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

But in this dark forest,
there's a stupid child called humanity,
who has built a bonfire and is standing beside it shouting,
"Here I am! Here I am!"

— *The Three-Body Problem: Dark Forest*

Abstract

Quantum technology is at the forefront of revolutionizing information processing by exploiting the principles of quantum mechanics to perform operations infeasible for its classical counterparts. As this field shifts from pure scientific exploration to practical application, developing advanced quantum control techniques becomes critical for precise and reliable quantum system manipulation. This thesis focuses on analytical quantum control techniques to enhance the performance of superconducting qubits, a leading architecture in quantum information processing. Due to their simplicity and efficiency, the model-based analytical methods discussed are particularly advantageous for experimental integration.

The thesis covers three aspects of quantum control: system modelling, control scheme design, and performance benchmarking. It starts by discussing the efficient modelling of quantum systems, aiming to reduce the dimension of the model while keeping the essential features of the dynamics. Here, to build more accurate and efficient models, the traditional perturbative approach is generalized by adopting the recursive structure and the exact diagonalization of a two-by-two matrix via Givens rotation. Building upon these modelling methods, the thesis addresses the dynamic control errors in quantum operations, including leakage, crosstalk, and other control errors in superconducting qubits. Based on the Derivative Removal by Adiabatic Gate (DRAG) framework, several applications are studied for two-qubit gates, multi-level qudit, and inter-qubit crosstalk. The key insight is to use the recursive formulation, which allows the integration of multiple DRAG corrections to address different errors simultaneously while maintaining simplicity and practicality for experimental calibration. Lastly, to validate the performance of control methods, the thesis introduces a new simulation tool for quantum circuits at the pulse level, based on the widely used software package Quantum Toolbox in Python (QuTiP). This tool incorporates realistic control errors and dissipation, aiding in the design, testing, and practical implementation of quantum control strategies in real-world settings.

Contents

Abstract

1 Introduction **1**

2 Quantum control **5**
2.1 Introduction to quantum control 5
2.1.1 Formulation of the quantum control problem 6
2.1.2 Open- and closed-loop quantum control 7
2.2 Numerical quantum control algorithms 8
2.2.1 GRadient Ascent Pulse Engineering algorithm 8
2.2.2 Chopped RAndom Basis algorithm 9
2.2.3 Gradient Optimization of Analytic conTrols algorithm . . . 10
2.3 Analytical quantum control for diabatic error 11
2.3.1 Coherent error and decoherence 11
2.3.2 Adiabatic theorem 13
2.3.3 Shortcuts to adiabaticity 13
2.3.4 Derivative Removal by Adiabatic Gate 15
2.4 The concept of recursion 16

3 Nonperturbative analytical diagonalization **19**
3.1 Introduction 20
3.2 Mathematical methods 22
3.2.1 Non-perturbative Analytical Diagonalization 22
3.2.2 Recursive Schrieffer-Wolff perturbation method 26
3.2.3 Block diagonalization 29

3.2.4 Comparison between different methods . . . 30
3.3 Physical applications . . . 32
3.3.1 Effective ZZ entanglement from non-dispersive interactions 32
3.3.2 ZZ coupling suppression in the quasi-dispersive regime . . . 35
3.3.3 The cross-resonance coupling strength . . . 41
3.4 Conclusion and outlook . . . 42
3.5 Appendix . . . 44
3.5.1 The error bound for truncating the BCH expansion . . . 44
3.5.2 Efficiency comparison between RSWT and SWT . . . 45
3.5.3 RSWT results for the ZZ interaction strength . . . 45
3.5.4 Effect of higher-order perturbation on the zero points of ZZ interaction . . . 49

4 Revisiting the single-Transmon DRAG pulse 51
4.1 Introduction to DRAG . . . 51
4.2 Notations . . . 53
4.3 First-order DRAG correction . . . 53
4.4 Detuning and phase ramping . . . 55
4.5 The Y-only DRAG correction . . . 56
4.6 An alternative derivation of the first-derivative DRAG . . . 56
4.7 High-order DRAG correction . . . 57

5 Experimental error suppression in Cross-Resonance gates via DRAG 61
5.1 Introduction . . . 62
5.2 Multi-derivative pulse shaping . . . 64
5.3 Application to control-qubit errors . . . 65
5.4 Application to multi-qubit operator errors . . . 70
5.5 Benchmarking the improved CR gate . . . 73
5.6 Discussion . . . 76
5.7 Methods . . . 77
5.7.1 Derivation of the recursive DRAG pulse . . . 77
5.7.2 Numerical simulation of the CR gate . . . 80
5.8 Appendix . . . 81
5.8.1 Partial suppression of transition errors . . . 81
5.8.2 Robustness of the recursive pulse . . . 81
5.8.3 Amplifying the transition error . . . 82
5.8.4 Data of the used Transmon qubits . . . 84
5.8.5 Additional data on the transition error suppression . . . 86
5.8.6 Calibration of the CNOT gate . . . 87

6 Leakage suppression in transmon qudits with DRAG **95**
6.1 Introduction . . . 96
6.2 Qudit Model for universal quantum gates . . . 97
6.2.1 Native gate set for superconducting qudit . . . 97
6.2.2 The transmon Hamiltonian . . . 97
6.2.3 Four-level effective model . . . 101
6.3 Recursive DRAG pulse for qudit gates . . . 102
6.3.1 Single-derivative DRAG and its limitation . . . 103
6.3.2 General DRAG correction for a n-photon transition . . . 103
6.3.3 First-order (linearized) solution for qudits . . . 106
6.3.4 Second-order solution for qudit . . . 107
6.3.5 Performance benchmarking . . . 108
6.4 Error beyond the targeted two-level subspace . . . 110
6.4.1 Phase error beyond the two target levels. . . . 110
6.4.2 Leakage on $|k+2\rangle \leftrightarrow |k+3\rangle$ . . . 111
6.4.3 Three-photo leakage $|k-1\rangle \leftrightarrow |k+2\rangle$ . . . 111
6.5 Conclusion and discussion . . . 112
6.6 Appendix . . . 112
6.6.1 Universality of the ladder transition . . . 112
6.6.2 Transmon circuit in the charge representation . . . 114
6.6.3 Derivation of the Leakage manifold . . . 116
6.6.4 Derivation of recursive DRAG pulse . . . 117

7 Quantum crosstalk suppression with DRAG **123**
7.1 Introduction . . . 123
7.2 Model of quantum crosstalk . . . 124
7.3 First-derivative DRAG pulse . . . 125
7.4 Second-derivative DRAG pulse . . . 126
7.5 Numerical simulation . . . 128
7.6 Analytical expression for the gate error . . . 130
7.7 Discussion . . . 132

8 Pulse-level quantum circuit simulation with QuTiP **135**
8.1 Introduction . . . 136
8.2 Software information . . . 137
8.3 Quantum circuits and open quantum dynamics . . . 139
8.3.1 Quantum circuits and gate-level simulation . . . 139
8.3.2 Continuous time evolution and pulse-level description . . . 140
8.4 Pulse-level quantum-circuit simulation framework . . . 143
8.4.1 Processor . . . 144
8.4.2 Model . . . 146
8.4.3 Compiler . . . 150
8.4.4 Scheduler . . . 151
8.4.5 Optimal control . . . 153
8.4.6 Noise . . . 154

8.4.7 Pulse . . . 157
8.4.8 Adding custom hardware models . . . 158
8.5 Importing and exporting circuits in QASM format . . . 159
8.6 Conclusion . . . 160
8.7 Data availability . . . 161
8.8 Appendix . . . 162
8.8.1 Simulating the Deutsch-Jozsa algorithm . . . 162
8.8.2 Compiling and simulating a 10-qubit Quantum Fourier Transform (QFT) . . . 163
8.8.3 Customizing the physical model and noise . . . 166

9 Summary and Outlook **173**
9.1 Summary and conclusions . . . 173
9.2 Future work . . . 175

References **179**

Acknowledgement **203**

List of Publications **205**

Curriculum vitae **207**

1 Introduction

Quantum technology, harnessing the peculiar principles of quantum mechanics, stands as a frontier in modern scientific and engineering fields, promising revolutionary advancements across various domains, from secure communication [1] and high-precision sensing [2] to quantum computing [3]. The late 20th century witnessed foundational experiments that demonstrated quantum entanglement and teleportation, underscoring the counterintuitive nature of quantum information processing. In those studies, the information is encoded in the quantum degrees of freedom of well-isolated systems, which exhibit unique properties governed by quantum theory. These early quantum processors, though primitive, revealed potential capabilities beyond the reach of classical physics.

Over the past decades, significant progress has been made in transitioning quantum technology from theoretical constructs and experimental curiosity to practical applications. This evolution is marked by profound developments in material science, experimental quantum engineering, and theoretical physics, paving the way for the exploitation of quantum phenomena in real-world applications.

The quantum bit, or qubit, is the most commonly used quantum information element. Unlike a classical bit, a qubit can exist in a superposition of multiple states. Among the various physical systems suitable for implementing qubits, the superconducting architecture has emerged as one of the leading technologies [4–6]. The key element is non-linear superconducting circuits operating at cryogenic temperatures, where resistance vanishes and quantum effects dominate. Superconducting qubits are promising due to their flexibility, manufacturability, and the advanced state of associated technologies derived from the semiconductor industry. However, this architecture also faces significant challenges concerning decoherence, which arise from their interaction with the environment and material defects. These interactions can disrupt the delicate state of qubits, leading to information loss and computation errors.

As the demand for higher-quality information processing increases, precise control over qubits' dynamics becomes increasingly crucial. This is particularly challenging due to the existence of ancillary levels in real quantum systems. The development of quantum control methods in this subfield focuses on the high-quality and fast manipulation of quantum states. For synthetic qubits, such as superconducting qubits, the need for precise control is further heightened by system parameter drift [7], which can alter the behaviour of a quantum system over time, requiring more frequent recalibration.

These challenges underscore the importance of developing advanced analytical control methods. In this context, "analytical" does not imply a mere focus on elegant expressions or refined mathematical models. Instead, the goal is to identify the most efficient strategies for engineering and calibrating systems in practice, ultimately enhancing the quality of quantum processors. This involves exploiting knowledge of the system model to understand and dissect the dynamics of quantum systems. As a result, control schemes are provided as a compact set of analytical expressions, such as control pulse shapes, with minimal free parameters to be determined in experiments. These methods are often adaptable and are easily fine-tuned for practical experimental setups.

In this thesis, we explore analytical quantum control methods, advancing the state-of-the-art by developing new strategies to efficiently model the system and counteract control errors in system dynamics. While the methods presented are general, we discuss specific applications tailored for superconducting qubits, demonstrating their practical relevance and potential to push the boundaries of current quantum technologies.

Organization of the Thesis The thesis starts with a general overview of quantum control theories and methods in Chapter 2, with an emphasis on those applicable to superconducting qubits. This includes a discussion on both numerical and analytical control strategies. Although we mainly investigate the application to superconducting Transmon qubits in this thesis, the control schemes are general. As such, we do not include an exhaustive review of superconducting qubits but provide the simplified physical model in each chapter when necessary.. For a more comprehensive background on superconducting qubits, readers are referred to the many review articles that have been continually updated over the past decade [4–6].

We then discuss in Chapter 3 the modelling of the quantum system and introduce Nonperturbative Analytical Diagonalization (NPAD) for deriving effective Hamiltonians. This chapter includes a detailed examination of diagonalization and block-diagonalization methods and their application to typical problems in superconducting qubits. By providing nonperturbative, closed-form solutions for effective Hamiltonians, NPAD allows for efficient control over quantum state manipulations, drastically reducing the size of the Hilbert space and the number of free parameters in the model. This method proves particularly useful in modelling and simulating the behaviour of superconducting qubits, where the Hamiltonian describing those non-linear oscillators is often sparse.

Building upon the modelling techniques, the following chapters focus on suppressing various control errors via the Derivative Removal by Adiabatic Gate (DRAG) method. Chapter 4 revisits its well-developed application on single-Transmon gates. It introduces the fundamental tools for analyzing leakage errors and deriving effective DRAG frames. In Chapter 5, we extend the applicability of DRAG to the Cross-Resonance gate, which is a widely used two-qubit gate on fix-frequency superconducting qubits. Due to the off-resonant microwave drives, DRAG is applied to eliminate multiple transition error sources and errors in the multi-qubit dynamics. In addition to analytical and numerical investigations, we also experimentally implement the drive scheme on IBM cloud hardware, demonstrating its efficacy in real-world quantum computing scenarios. The next chapter, Chapter 6, exploits the higher levels of the non-linear oscillators, using multiple levels as information units, i.e., qudits. The involvement of multiple levels also introduces new leakage channels, posing challenges for analytical control methods. Finally, in Chapter 7, we study the error across multiple quantum elements, i.e., crosstalk among qubits. This chapter presents methods for suppressing unwanted excitations of neighbouring qubits during single-qubit operations.

The last contribution of this thesis is presented in Chapter 8, which introduces a tool for characterizing and benchmarking quantum control schemes by simulating the most widely used quantum computing model, quantum circuit, at the level of time evolution using the Lindblad master equation. By compiling the quantum circuit into time-dependent Hamiltonian models, this framework incorporates realistic control and dissipation errors that closely reflect the experimental setup. The software package is open-sourced and is continually being updated and developed.

2 Quantum control

2.1 Introduction to quantum control

Quantum control theory represents a dynamic and expanding field of research. Since the establishment of quantum mechanics, the manipulation of quantum phenomena has been a driving force behind many groundbreaking discoveries. One of the main goals of quantum control theory is to establish a theoretical framework and develop a series of systematic methods for the active manipulation and control of quantum systems [8]. Achieving this objective is far from straightforward, due to the unique and complex properties of quantum dynamics, such as entanglement, which have no counterparts in classical physics [9].

Historically, the earliest aims of quantum control were centred on manipulating the quantum phenomena at the atomic and molecular scales. The goal was to break and form chemical bonds in polyatomic molecules, with the goal of creating a chemical product unattainable by conventional means [10, 11]. Quantum control strategies exploited the delicate interplay of constructive and destructive interference among the microscopic quantum degrees of freedom.

As the field matured, the focus of quantum control expanded to include a deeper understanding of the internal degrees of freedom of atoms and molecules. This broader perspective was significantly advanced by developments in Nuclear Magnetic Resonance (NMR) techniques that emerged from the need for precise spectroscopic analysis of complex molecular structures [12]. Here, the development and refinement of laser technology provided a significant boost to quantum control efforts. These advancements enabled the coherent control of quantum states through precise temporal and spatial modulation of laser fields, leading to innovative experiments that have continually pushed the boundaries of achievability [13].

The emergence of quantum information science as a distinct field opened new objectives for quantum control. Theoretical proposals for quantum computing [3],

quantum cryptography [14], and quantum teleportation [15] during the 1990s and the progress in experimental demonstration in the following decades highlighted the need for precise and scalable control of quantum systems. This period witnessed the convergence of quantum control with quantum information processing, focusing on developing techniques for high-fidelity quantum gate operations, measurement and state preparation on various physical architectures.

In quantum information processing, the focus is often on a relatively small physical system characterized by a few active dynamical degrees of freedom, well isolated from environmental perturbations and dissipative couplings. One common approach to achieving this isolation is to operate the experimental apparatus at very low temperatures and close to a vacuum environment. Quantum control in this field becomes a subset of quantum engineering with the aim of meeting the stringent requirements for implementing quantum information registers [16]. In this context, quantum control tasks involve not only optimizing control functions that drive the desired dynamics but also building accurate models that capture the essential behaviours of the system.

For solid-state systems studied in this thesis, one generally finds an intriguing combination of characterization and control issues. Whereas accurate models can often be constructed for atomic systems, modelling solid-state systems typically requires a more phenomenological approach [9]. In particular, it is seldom possible to derive comprehensive models for the many-body dynamics and the residual environmental couplings for syntactic superconducting quantum circuits. Here, simplified models are crucial for analysing dynamics and devising effective control strategies, especially in complex systems like superconducting quantum circuits.

While this thesis touches on many aspects of quantum control, there is one subfield that we do not cover: the controllability of quantum systems. Controllability describes the capability to steer a quantum system towards a specific target state [17]. In this thesis, our attention is directed towards the more practical aspect of quantum control, particularly for well-understood quantum systems where controllability is already established. For those systems, however, identifying the most efficient way to direct the system evolution and outpace decoherence presents a separate challenge that we aim to address.

2.1.1 Formulation of the quantum control problem

Throughout this thesis, we use the finite-dimensional linear models of quantum control systems [8]. The state $|\psi(t)\rangle$ of a closed quantum system evolves according to the Schrodinger equation

$$i\hbar\frac{\mathrm{d}}{\mathrm{d}t}|\psi(t)\rangle = \hat{H}|\psi(t)\rangle \tag{2.1}$$

where $\hat{H}$ is the time-independent Hamiltonian. We set $\hbar = 1$ for the ease of notation. For simplicity, we consider here only finite-dimensional quantum systems, which is a valid approximation in many practical situations. The control of the

quantum system is realized by a set of control functions $c_k(t) \in \mathbb{R}$, which is a set of time-dependent functions that drive the system to undergo controlled evolution

$$\hat{H} = \hat{H}_0 + \sum_k c_k(t)\hat{H}_k. \tag{2.2}$$

Following the quantum control terminology, $\hat{H}_0$ is referred to as the drift Hamiltonian and $\hat{H}_k$ the control Hamiltonians. The time-dependent function $c_k(t)$ represents the control functions, which are often also referred to as the control pulses in this thesis.

The task for quantum control is to find a set of control functions that drives the initial state $|\psi_0\rangle$ into a predefined target state $|\psi_f\rangle$. More generally, especially in the context of quantum computing, the desired evolution is predefined on a set of basic states. Therefore, the target is a unitary governed by the control Hamiltonian

$$i\frac{\mathrm{d}}{\mathrm{d}t}\hat{U} = \hat{H}\hat{U} \tag{2.3}$$

with $\hat{U}(t=0) = \hat{\mathbb{I}}$ the identity. The solution is given by $\hat{U} = \hat{\mathcal{T}} \exp\left(-\int_0^T i\hat{H}(t)\mathrm{d}t\right)$, where $\hat{\mathcal{T}}$ is the time-ordering operator.

2.1.2 Open- and closed-loop quantum control

One of the most significant differences between classical and quantum control lies in the measurement postulate. In quantum systems, any attempt to acquire information from a quantum system inevitably affects or even destroys the quantum state. Therefore, many practical control techniques are classified as open-loop quantum control, where predefined control functions are applied to the quantum system without using any real-time feedback of the quantum state. In this approach, control functions are typically determined by either analytically studying the model of the system or numerically simulating the time evolution. Hence, open-loop control can encounter difficulties in large quantum systems due to the exponential growth of the Hilbert space and may be sensitive to the accuracy of the model parameters.

A natural solution to this problem is to explore closed-loop control strategies. Two forms of closed-loop control are proposed for quantum systems [8]: closed-loop learning control and quantum feedback control. In closed-loop learning control, each cycle adopts a control strategy and applies it to the quantum system. The result is observed and a new control is proposed based on a learning algorithm. In quantum feedback control, however, information is continuously extracted from the target system during the whole process of control [18–20].

In this thesis, we do not discuss quantum feedback control as the real-time feedback is still limited by the current state of experimental technology. Instead, we focus primarily on the open-loop control strategy but put significant emphasis on the practical application of closed-loop control, often referred to in quantum

engineering as calibration. This is achieved by designing effective models of the system and building analytical pulse shapes parameterized by a small number of free parameters. These pulse Ansatzes capture the dominant dynamics while the additional parameters can be fine-tuned to optimize performance against specific types of errors. This control strategy motivated by the underlying physical model is particularly relevant for practical implementation as experimental sampling and characterization are often expensive and subject to noise. In fact, in practice, each parameter is often calibrated individually by measuring the corresponding error and the learning algorithm, in this case, is simply a one-dimensional optimization.

2.2 Numerical quantum control algorithms

For many practical quantum control problems, obtaining an analytical solution proves to be impractical. As a result, various numerical quantum control algorithms have been proposed over the last few decades to solve the control problem. Given the challenging nature of time-evolving quantum systems governed by the Schrödinger equation, the primary objective is often to efficiently extract valuable information from the dynamics. This may involve computing gradients or simplifying the parameterization of control pulses. Below, we provide a concise overview of some well-known quantum control algorithms. For a more comprehensive review of numerical control algorithms, readers are referred to dedicated literature [21–23].

2.2.1 GRadient Ascent Pulse Engineering algorithm

The GRAPE algorithm is one of the earliest-proposed and widest-used algorithms, originally designed for the NMR system [24]. The advantage, as the name suggested, is to make use of the analytical gradient information of the time evolution at each discretized time step to update the pulse function.

For simplicity, we use the problem of state preparation, which can be naturally generalized to unitary operation by providing the identity as an initial state. We consider preparing the state $|\psi_f\rangle$ from the initial state $|\psi_0\rangle$ under the control Hamiltonian equation (2.2). Usually $|\psi_0\rangle$ is the ground state of the drift Hamiltonian $\hat{H}_0$ and the target $|\psi_f\rangle$ has some useful quantum properties such as a Bell state or a GHZ state. The task is to find a set of time-dependent control functions $c_k(t)$ defined for $t \in [0, T]$ such that the state reaches the target $|\psi_f\rangle$ by the end of the evolution. With a loss function defined by the quality of the final state, such as the overlap fidelity $\mathcal{F} = |\langle\psi_f|\psi(T)\rangle|^2$.

One important strategy of the GRAPE algorithm is to break down evolution into small time steps and consider a piecewise constant pulse. The total time is chopped into N intervals of length T/N

$$c_k(t) = c_k(j) \quad for \quad (j-1)\,\delta t < t < j\,\delta t. \tag{2.4}$$

The total evolution can then be broken down similarly into N time steps:

$$\hat{U}(T) = \hat{U}_N \hat{U}_{N-1} \cdots \hat{U}_2 \hat{U}_1 \tag{2.5}$$

with $\hat{U}_j$ defined by the time evolution under the constant Hamiltonian $\hat{H}(j) = \hat{H}_0 + \sum_k c_k(j)\hat{H}_k$ for the j-th time interval δt. This means that, in total, we have $N \cdot K$ parameters, i.e., the product of the total number of the time steps and the number of control functions.

To obtain the gradient information, we need to isolate the step that involves only one parameter. We rewrite the fidelity as the overlap of the forward and backwards propagated states:

$$\mathcal{F} = \left|\left\langle \psi_f \middle| \hat{U}(T)\psi_0 \right\rangle\right|^2 = \left|\left\langle \hat{U}_{m+1}^\dagger \cdots \hat{U}_N^\dagger \psi_f \middle| \hat{U}_m \cdots \hat{U}_1 \psi_0 \right\rangle\right|^2 . \tag{2.6}$$

This allows us to focus on one specific time step m and compute the gradient with respect to $c_k(m)$. Since $\hat{U}_m$ is the only one that depends on $c_k(m)$, this simplifies to the derivative of an exponential

$$\frac{\partial}{\partial c_k(m)}\hat{U}_m = \frac{\partial}{\partial c_k(m)} \exp\left[-i\delta t\left(\hat{H}_0 + \sum_k c_k(m)\hat{H}_k\right)\right] \approx -i\delta t \hat{H}_k \hat{U}_m. \tag{2.7}$$

The last approximation is valid if the time step δt is sufficiently small. The intuition is that for small time steps, the sum can be tailored into products of many exponential functions independent on $c_k(m)$ and $\exp\left[-i\delta t c_k(m)\hat{H}_k\right]$, neglecting all the high order commutators. The rigorous derivation requires Lie group and Lie algebra theory, which is covered in more specific literatures [21, 25].

Following the above derivation, with a given set of discretized control functions $c_k(j)$, the gradient-based update is given as

$$c_k(j) \to c_k(j) + i\epsilon\delta_t \left\langle \hat{U}_{m+1}^\dagger \cdots \hat{U}_N^\dagger \psi_f \right| \hat{H}_k \left| \hat{U}_m \cdots \hat{U}_1 \psi_0 \right\rangle , \tag{2.8}$$

where ϵ is the updating step size. Compared to gradient-independent optimizations, this analytical form of the gradient provides a much more stable and fast convergence and allows for optimization in a much larger parameter space.

Numerical implementation of the GRAPE algorithm is available in several software packages such as in QuOCS [26], QuTiP [27], and `QuantumControl.jl` [28]. A similar algorithm, the Krotov algorithm, has also been implemented [28, 29], which follows the same idea of discretization and analytical gradient calculation but performs the updates in a different way.

2.2.2 Chopped RAndom Basis algorithm

Due to discrepancies between the model and quantum system, in reality, closed-loop optimization is often unavoidable. Although the piecewise constant pulse used in the GRAPE algorithm enables the calculation of the analytical gradient at each time step for the optimization, the number of parameters grows significantly as the increase of time steps and the number of controls. In addition, a real experimental pulse generator often prefers smooth pulses rather than larger jumps in amplitude. Those practical reasons make GRAPE less suited for closed-loop optimization with fine time steps.

The CRAB algorithm, on the other hand, uses a different Ansatz to represent the control functions to reduce the parameter space [30–32]. Instead of piecewise constant Ansatz, the CRAB algorithm makes use of the trigonometric bases

$$c_k(t) = \mathcal{I}_0 \left[1 + \sum_j A_{k,j} \sin(\omega_j t) + B_{k,j} \cos(\omega_j t) \right] \tag{2.9}$$

with $\mathcal{I}_0$ a normalization constant to keep the initial and final control pulse value fixed. In some formulation, $\mathcal{I}_0$ is replaced by a time-dependent shape function to ensure the boundary condition of the pulse. The frequencies ω_k are chosen uniformly from $[0, \omega_{\max}]$. Based on the Ansatz, the two sets of parameters, $\{A_k\}$ and $\{B_k\}$ are optimized for the best fidelity. In this case, the gradient can no longer be calculated analytically since the time-dependent Hamiltonian is often nonintegrable. One either needs to estimate the gradient from finite sampling or refer to gradient-independent optimization such as the Nelder-Mead algorithm [33].

Extending on the CRAB methods, in particular to expanding the search space while maintaining the number of parameters, dressed CRAB (dCRAB) was proposed [34]. It introduces super iterations, which add new base functions with different ω_k and optimizes those new amplitude parameters on top of the existing solution. This adaptive method allows the algorithm to explore new directions while keeping the number of free parameters under control. The CRAB algorithms have also been implemented in several packages such as QuOCS [26] and QuTiP [27].

2.2.3 Gradient Optimization of Analytic conTrols algorithm

The development of the two algorithms above highlights two general considerations for control algorithms: convergence speed and efficient parameterization. The GOAT algorithm tries to combine those two aspects by ensuring the precise computation of gradients while maintaining an efficient parameterization of the drive pulse [35].

Achieving a stable and efficient gradient computation for dynamics with continuous pulse representations is challenging because of the non-integrability of quantum dynamics. Often, an analytical solution of the unitary evolution is unattainable and numerical solutions of ordinary differential equations (ODEs) are required. The GOAT algorithm addresses this challenge by employing a methodology similar to the automatic differentiation techniques seen in Neural-ODE [36]. Analogous to solving a Schrödinger evolution problem via numerical ODEs, the derivative of a state or unitary with respect to control parameters can also be "evolved" concurrently with the system dynamics [37]

$$\partial_t \begin{pmatrix} \hat{U} \\ \partial_\alpha \hat{U} \end{pmatrix} = -i \begin{pmatrix} \hat{H} & 0 \\ \partial_\alpha \hat{H} & \hat{H} \end{pmatrix} \begin{pmatrix} \hat{U} \\ \partial_\alpha \hat{U} \end{pmatrix} \tag{2.10}$$

where α represents the parameters to be optimized. The computational cost of the numerical evolution scales linearly with the number of parameters. At each optimization step, an ODE solver is called to compute the evolution $\hat{U}$ and the

gradient $\partial_\alpha \hat{U}$, which is then used by a gradient-based optimizer to update the parameter α.

2.3 Analytical quantum control for diabatic error

In the last section, we introduced various numerical algorithms, which are often required for very complicated quantum dynamics. Instead of studying the preparation of the ground state of some large many-body Hamiltonians, in this thesis, we aim to push a relatively simple physical system to its best performance, for quantum information processing. Therefore, the emphasis of this thesis is on the analytical methods for designing quantum control schemes.

These model-based solutions are favoured for their compactness and convenience in experimental calibration. The main physical system investigated in this thesis is the superconducting qubits architecture, which benefits from well-established fabrication and engineering techniques for the microelectronics industry [4–6]. Nonetheless, this system is not without its challenges. Significant issues such as fabrication inhomogeneity [38, 39] and parameter drift [7] may affect its performance. Fabrication inhomogeneity implies that each qubit may differ slightly and the drive scheme for each of them needs to be calibrated separately. Parameter drift refers to the tendency of system parameters to deviate over time from their last measured values due to the change of environment such as magnetic field. To bring the best performance, the machine usually needs to be calibrated on a daily basis. Here, an analytical approach proves particularly advantageous because it offers a concise Ansatz with a limited number of parameters to be calibrated. These parameters typically correspond to directly measurable physical quantities, making them ideal for frequent adjustment and fine-tuning.

In the rest of this chapter, we briefly introduce the control error studied in this thesis and the analytical methods previously devised to compensate for those errors.

2.3.1 Coherent error and decoherence

The error in quantum control can be broadly divided into two main categories: coherent error (unitary error) and decoherence (dissipative) error. Coherent error describes errors in the unitary evolution $\hat{U}$ that deviate from the ideal dynamics $\hat{U}_\mathrm{I}$ in the target subspace. Unlike stochastic errors induced by environmental interactions, coherent errors are deterministic and reproducible. This means they can be replicated by repeating the same drive scheme and are less susceptible to statistical effects from the random environment. However, they can still pose significant challenges for error correction codes, as they introduce non-Markovian errors and errors beyond the qubit model [40]. As such, coherent errors are often targets for open-loop quantum control techniques.

Among the most well-known examples of coherent error in the superconducting qubits architecture is the leakage error. In the commonly used Transmon qubits [41],

the lowest two levels of a nonlinear oscillator are used as a qubit. However, due to the finite energy separation between the computational subspace and higher-energy ancillary levels, there exists a probability of the system transitioning into these non-computational states during the operation. This type of leakage, along with associated phase errors, substantially limits the fidelity of quantum operations. Such errors are prominently addressed using analytical control techniques like the DRAG method, which is detailed later in this chapter. This thesis aims to extend and generalize these methods to tackle more complex and practically relevant scenarios.

On the other hand, decoherence error, another critical challenge in quantum computing, arises from disturbances induced by the random, uncontrollable environment surrounding the quantum system. It is characterized by the non-unitary dynamics and the loss of quantum information, resulting in the deterioration of the quantum state towards a classical state. This process is typically described using statistical terms, such as spectral density and Lindblad operators with associated jump rates, considering the environment often has infinite degrees of freedom. Common assumptions include that the interaction between the quantum system and its environment is small compared to the system's own dynamics, known as the Born approximation. And that the environment is so large that its statistical properties are not changed by the weak coupling, known as the Markov approximation [42]. Under these approximations, the environment's memory effects on the system dynamics are negligible, allowing for a simplification in the mathematical treatment.

Within this framework, the quantum state is represented by a density matrix $\hat{\rho}$, which includes the statistical properties in its definition. The evolution of $\hat{\rho}$ over time can be effectively described by various forms of master equations, which govern the open-system dynamics [43]. From the perspective of quantum information, the evolution operator is no longer described by a simple unitary propagator, but by quantum channels. The description of quantum channels inherently includes the loss of information, a process generally irreversible under natural system dynamics, unless additional quantum resources are used, such as quantum state purification [44] or error correction [45].

In this thesis, with the exception of Chapter 8, our primary focus is on modelling and control methods aimed at suppressing coherent errors in quantum systems. This involves providing precise and clear Hamiltonian descriptions of the system and designing drive schemes for high-fidelity unitary time evolution. However, it is worth noting that the methods proposed in this work also extend their utility to overcoming the challenges presented by decoherence. For instance, the simplest characterization of decoherence, in quantum engineering terminology, is represented by the coherence time T_1 and T_2 of qubits. With finite coherence time, a prolonged quantum operation will inevitably lead to a degradation of the quality. By introducing an advanced drive scheme that reduces operation time, quantum control based on the schrödinger dynamics can also contribute to surpassing the limitations imposed by qubits' coherence.

2.3.2 Adiabatic theorem

The adiabatic theorem is a cornerstone of quantum simulation [46] and adiabatic quantum computing [47]. Although it may not be the central element in contemporary quantum information processing techniques with superconducting qubits, the adiabatic theorem still plays a crucial role, particularly in understanding the effects of ancillary energy levels on qubits. According to the theorem, a quantum system that starts in an eigenstate of its Hamiltonian will remain in a corresponding eigenstate of the evolving Hamiltonian if the Hamiltonian changes sufficiently slowly. This implies that the system can evolve from one state to another without inducing transitions between different energy levels, maintaining its quantum coherence throughout the process.

The simplest example is an interpolation between two constant Hamiltonians $\hat{H}_0$ and $\hat{H}_1$,

$$\hat{H}(t) = (1 - \lambda(t))\hat{H}_0 + \lambda(t)\hat{H}_1, \tag{2.11}$$

where $\lambda(0) = 0$ and $\lambda(t_f) = 1$. The system initialized in the ground state of $\hat{H}_0$, will end up in the ground state of $\hat{H}_1$ at the end of the time evolution if the adiabatic condition is satisfied. Typically, the initial state is easy to prepare (e.g., the all-zero state), while the final ground state of $\hat{H}_1$ is hard. This forms the essence of adiabatic quantum computing [47].

We define the eigenstates of the Hamiltonian $\hat{H}(t)$ at time step t as $|\psi_j(t)\rangle$ and $|\psi_0(t)\rangle$ as the ground state. They are referred to as the instantaneous eigenstates. The adiabatic condition can be mathematically expressed as [47]

$$\max_{t\in[0,t_f]} \frac{\left\langle \psi_j(t) \middle| \dot{\psi}_k(t) \right\rangle}{|E_j - E_k|} \ll 1 \tag{2.12}$$

for $j \neq k$ and $E_j(t)$ the eigenvalue of $|\psi_j(t)\rangle$. The overdot $\left|\dot{\psi}_k(t)\right\rangle$ denotes the time derivative of the eigenstates. This condition implies that the rate of change in the excited instantaneous states, particularly for non-ground states, must be small relative to the energy gaps to the ground state, ensuring minimal overlap.

2.3.3 Shortcuts to adiabaticity

Achieving true adiabatic evolution in quantum systems typically requires the process to be conducted slowly to satisfy the adiabatic condition. However, this slow evolution is often impractical in realistic quantum systems, where decoherence and other imperfections limit the operational timescales. The concept of Shortcuts to Adiabaticity (STA) offers a solution to accelerate adiabatic processes without sacrificing the fidelity of maintaining the ground state population. The name STA was first coined by Chen et al. in Ref. [48, 49] but the fundamental principle traces back to much earlier works [50, 51].

The general approach is to find a particular effective frame, where the diabatic errors, i.e., errors that cause transitions between different quantum states, can be compensated for by additional control Hamiltonians. This effective frame is often

chosen as the instantaneous eigenstates of the system, although other frames might be used depending on which configuration allows for more effective cancellation of diabatic errors [52]. In the lab frame, the quantum state still evolves out of the ground state but carefully designed STA ensures that it returns to the ground state by the end of the evolution. Often the boundary conditions of the pulse shapes need to be satisfied to guarantee that by the end of the evolution, the lab frame and the effective frame are equivalent.

Though collectively termed as STA, it involves several distinct methodologies, including Counter-Diabatic (CD) driving, invariant-based inverse engineering and the fast-forward approach [53]. In the following, we give a brief introduction to the CD approach, which is closely related to the control method used in this thesis.

The initial formulation of the CD approach was introduced by Demirplak and Rice in Ref. [50] and Berry in Ref. [51]. In the effective frame formed by the instantaneous eigenstates $|\psi_j(t)\rangle$, a solution is constructed as [51]

$$\hat{H}_{\mathrm{CD}} = i\sum_j \Big(|\dot{\psi}_j(t)\rangle \langle\psi_j(t)| - \langle\psi_j(t)|\dot{\psi}_j(t)\rangle |\psi_j(t)\rangle \langle\psi_j(t)| \Big) \tag{2.13}$$

While this formulation might seem straightforward, its application is not trivial due to several challenges. First, the control Hamiltonian must be available in the lab frame. As the portion of $\hat{H}_1$ increases, the instantaneous eigenstates $\psi_j(t)$ could be highly entangled states, making the practical realization of $\hat{H}_{\mathrm{CD}}$ challenging. Partial implementation of $\hat{H}_{\mathrm{CD}}$ does not always improve the result due to the complexity in quantum dynamics. Second, the full-time-dependent spectrum information is needed to design the CD Hamiltonian, which is impractical in many problems.

An alternative way to derive the CD Hamiltonian is to diagonalize the time-dependent Hamiltonian.

$$\hat{H}_{\mathrm{eff}} = \hat{V}(t)\hat{H}_0(t)\hat{V}^\dagger(t) + i\dot{\hat{V}}(t)\hat{V}^\dagger(t), \tag{2.14}$$

where $\hat{V}(t)$ diagonalize the Hamiltonian $\hat{H}_0(t)$ for each time t. It is then straightforward to see that in the lab frame, a correction Hamiltonian can be derived

$$\hat{H}_{\mathrm{CD}} = -i\hat{V}^\dagger(t)\dot{\hat{V}}(t). \tag{2.15}$$

This solution is overkill as it ensures that all the eigenstates stay intact, not just the ground state, but it will become handy when we go beyond the adiabatic evolution of the ground state.

Despite its elegance, the CD approach typically finds applications only in simpler quantum systems, where analytical solutions are feasible. Common applications are two-level systems [48] or three-level Λ systems with certain symmetry [48, 54, 55], where a simple analytical diagonalization can be found. Going beyond a three-level system involving exploring additional properties of the system, such as a special Lie algebra subspace [56] or the Quantum Rabi model with harmonic oscillators [57]. A few attempts have been made to generalize it to many-body dynamics by variational optimization [58, 59] or special symmetry consideration [60–62].

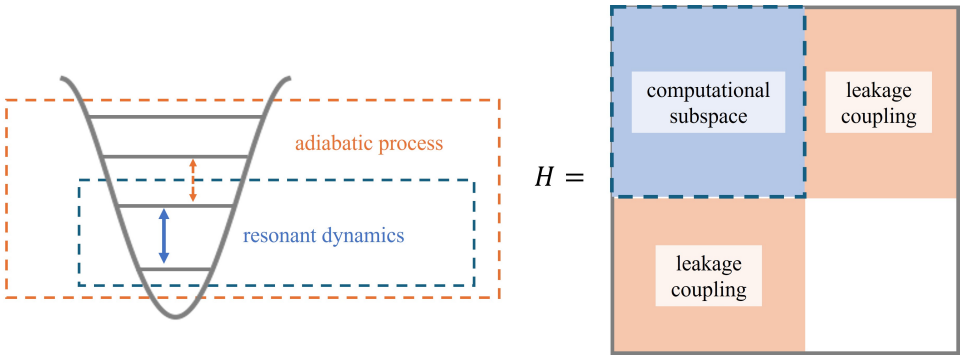


Figure 2.1: Illustration of the DRAG method concept. The left side shows a basic use case discussed in the proposal [64], where the lowest two levels are treated as a qubit and are controlled resonantly. In this scenario, the two-level subspace remains adiabatic relative to the auxiliary higher levels. The right side depicts the block diagonalization of the Hamiltonian, illustrating the mathematical approach underlying the method.

Another challenge with STA is the vast number of possible choices for the effective frame, each imposing different requirements on the control Hamiltonians. The time-dependent Hamiltonian can encode complex dynamics, the imperfect dynamics in one frame may appear clean and simple in another. Consequently, different but equivalent solutions can exist [52]. This diversity offers considerable flexibility in designing STA protocols, but it also makes it difficult to identify the optimal configuration [63].

2.3.4 Derivative Removal by Adiabatic Gate

In contrast to STA methods, which mostly address state preparation challenges, the Derivative Removal by Adiabatic Gate (DRAG) method is particularly tailored for reducing leakage to ancillary levels during quantum operations. Introduced independently from STA by Motzoi et al. in 2009, DRAG was designed to manage single-qubit operations in nonlinear harmonic oscillators like Transmon qubits [64]. It addresses the challenges of leakage errors introduced by transitions to higher non-computational levels, which are only finitely detuned from the qubit frequency.

In this setup, the lowest two (qubit) levels take the role of a (degenerated) ground state in an STA problem. When driving the desired transition between the qubit levels, the drive should be adiabatic enough with respect to the ancillary levels to prevent leakage, as shown in Figure 2.1. Similar to the adiabatic theorem, if the operation time is infinitely long, the population should stay fully in the qubit subspace. The DRAG method is then adopted to speed up the operation by compensating for the leakage couplings and the effects of Stark shifts. While the principles of DRAG bear resemblances to CD driving, DRAG specifically addresses the challenges posed by degenerated subspaces and inherently on-resonant dynamics

typical in practical quantum systems . In addition to suppressing leakage errors, the phase also needs to be corrected to implement the desired quantum operation.

The essence of the DRAG method is to replace the diagonalization operator $\hat{V}(t)$ in equation (2.14) by a block diagonalization (see also Chapter 3), which preserves the desired on-resonant dynamics, while treats the leakage to the non-computational levels as non-adiabatic errors, as illustrated in Figure 2.1. Often, the available amount of control terms (with fixed pulse shapes) is limited, compared to the number of leakage couplings. The DRAG method opens new degrees of freedom in the time domain by introducing derivatives of the original pulse shape, following equation (2.15).

Unlike many scenarios where CD driving is applicable, the Hamiltonian for a system with mixed diabatic and adiabatic dynamics does not readily yield integrable solutions. However, when the leakage coupling is relatively small compared to the separation of the ancillary energy level, a perturbative approach can be used to derive an effective Hamiltonian. By targeting the dominant error source, very efficient DRAG pulse Ansatzes can be designed with only a few free parameters to be calibrated in experiments.

Although the DRAG methods have been theoretically studied under many different models [64–69], here we focus on its application on more realistic problems, where multiple errors often present at the same time. We will revisit the mathematical details of DRAG in Chapter 4, and then expand its application to two-qubit gates (Chapter 5), multi-level qudit systems (Chapter 6) and quantum crosstalk (Chapter 7). These DRAG formulations provide efficient pulse Ansatz for practical implementation, which can also be combined with classical frequency engineering [70] and optimal control algorithm [71] to gain further improvement.

2.4 The concept of recursion

Apart from the quantum control techniques introduced in the sections above, we provide a brief description of recursion. While recursion is a general method rather than a quantum-control-specific technique, it plays a central role in several methods introduced in this thesis.

Recursion is a fundamental concept in mathematics and computer science, characterized by defining a function or process in terms of itself. Typically, recursion involves solving a problem by breaking it down into smaller, more manageable sub-problems of the same type, until reaching a base case that can be solved directly. Each recursive step consists of only simple processes, which do not solve the problem but transform it into a simpler problem of the same type. It is widely used in programming and algorithm design, often providing elegant solutions for complex problems that might otherwise be challenging to solve directly. Although recursion and iteration are often equivalent to each other, recursion sometimes provides a particularly elegant perspective to the problem.

The simplest teaching example is the factorial function: $\mathrm{factorial(n)} = \prod_{\mathrm{k}=1}^{\mathrm{n}} \mathrm{k}$

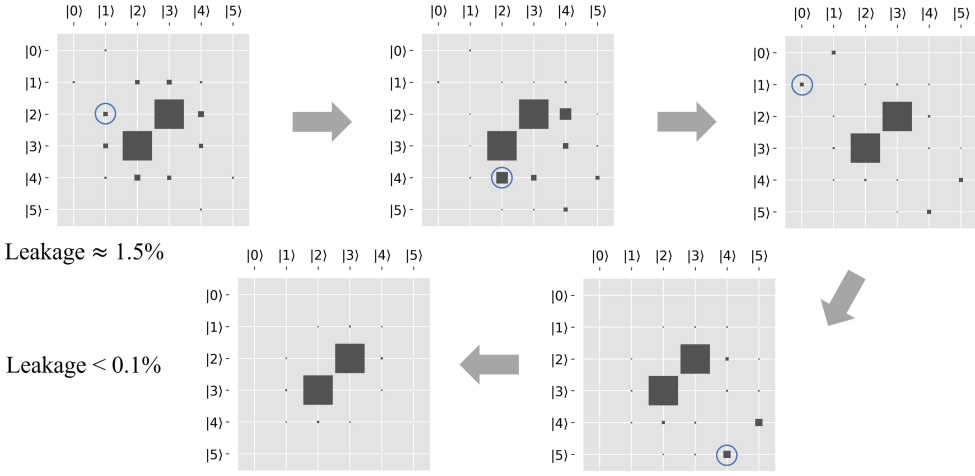


Figure 2.2: Illustration of recursively suppressing leakage errors. The plot shows abstract representations of unitary propagator matrices, focusing on the off-diagonal elements while omitting the diagonal ones. The size of each block corresponds to the absolute square of the matrix element. The two largest blocks represent the desired transitions, while the smaller blocks indicate leakage. In each recursive step, a specific leakage coupling is targeted (circled) and a DRAG correction is applied, effectively suppressing the targeted leakage.

with factorial(0) = 1. The recursion step simplifies the question by the relation

$$\text{factorial(n)} = \text{n} \times \text{factorial(n} - 1),$$

with the recursion terminating when $n = 0$ at factorial(0) = 1. Another example is the Fibonacci sequence, where the analysis of the recursive relation can even reveal an analytical expression.

Instead of trying to tackle the big problem, recursion only aims at making a small step. This way of thinking is also ubiquitous in the field of quantum physics, usually for defining efficient algorithms for a structured problem, such as renormalization group theory [72], quantum Fourier transform [3] and quantum error correction [45]. In each case, recursion enables the transformation of a complex problem into a series of simpler problems within the same category, leading to the final solution.

In this thesis, recursion emerges as a central concept, particularly within the frameworks of recursive diagonalization methods and applying DRAG correction against multiple error terms. The elementary step here is the simplest scenario in quantum physics, a two-level system, for which an analytical solution often exists. Thus, we build recursion by isolating and addressing a single two-level subsystem at each step. This targeted approach allows for systematic iteration through all dominant terms, gradually refining the system until the desired solution is achieved. This is illustrated in Figure 2.2.

For time-independent problems (see Chapter 3), it involves addressing one off-diagonal term via a Givens rotation at each step. Following the Jacobi iteration, the matrix progressively converges to the desired form. Moreover, in the perturbative limit, it is equivalent to a recursive version of the well-known Schrieffer Wolff diagonalization. In this case, one recursion step removes the leading order off-diagonal coupling. The remaining coupling is quadratically smaller at every iteration, much faster than the linear suppression by solving the BCH expansion for the generator at each order.

The problem becomes more complex with time-dependent dynamics (see Chapters 5 and 6), where finding a converging recursive solution can be challenging or sometimes infeasible. Nonetheless, the sparse and structured nature of many quantum Hamiltonians studied allows for the derivation of analytical solutions through approximations, with very few recursion steps. The recursive formula can sometimes provide a surprisingly compact and efficient solution, avoiding unnecessary expansions.

3

Nonperturbative analytical diagonalization

Deriving effective Hamiltonian models plays an essential role in quantum theory, with particular emphasis in recent years on control and engineering problems. In this chapter, we present two symbolic methods for computing effective Hamiltonian models: the Non-perturbative Analytical Diagonalization (NPAD) and the Recursive Schrieffer-Wolff Transformation (RSWT). NPAD makes use of the Jacobi iteration and works without the assumptions of perturbation theory while retaining convergence, allowing it to treat a very wide range of models. In the perturbation regime, it reduces to RSWT, which takes advantage of an in-built recursive structure where remarkably the number of terms increases only linearly with perturbation order, exponentially decreasing the number of terms compared to the ubiquitous Schrieffer-Wolff method. In this regime, NPAD further gives an exponential reduction in terms, i.e. superexponential compared to Schrieffer-Wolff, relevant to high precision expansions. Both methods consist of algebraic expressions and can be easily automated for symbolic computation. To demonstrate the application of the methods, we study the ZZ and cross-resonance interactions of superconducting qubits systems. We investigate both suppressing and engineering the coupling in near-resonant and quasi-dispersive regimes. With the proposed methods, the coupling strength in the effective Hamiltonians can be estimated with high precision comparable to numerical results.

This chapter has been published, with minor changes, as Boxi Li, Tommaso Calarco, and Felix Motzoi, *Nonperturbative Analytical Diagonalization of Hamiltonians with Application to Circuit QED*, PRX Quantum 3.3 (2022): 030313. [73]. The thesis author conducted most of the analysis, composed all the figures and wrote the manuscript with input from the advisors.

3.1 Introduction

Deriving effective models is of fundamental importance in the study of complex quantum systems. Often, in an effective model, one decouples the system of interest from the ancillary space, as shown in Figure 3.1. The dynamics are then studied within the effective subspace, which is usually much easier than in the original Hilbert space, and provides fundamental information such as conserved symmetries, entanglement formation, orbital hybridization, computational eigenstates, spectroscopic transitions, effective lattice models, etc. In terms of the Hamiltonian operator, an effective compression of the Hilbert space can be achieved by diagonalization or block diagonalization.

When the coupling between the system and ancillary space is small compared to the dynamics within the subspace, the effective model is often derived by a perturbative expansion. In the field of quantum mechanics, a ubiquitous expansion method that enables reduced state space dimension is the Schrieffer-Wolff Transformation (SWT) [74, 75], also known in various sub-fields as adiabatic elimination [76], Thomas-Fermi or Born-Oppenheimer approximation [77, 78], and quasi-degenerate perturbation theory [79]. Finding uses throughout quantum physics, SWT can be found in atomic physics [76], superconducting qubits [80, 81], condensed matter [75], semiconductor physics [82], to name a few.

The SWT method is however limited to regimes where a clear energy hierarchy can be found and therefore fails to converge for a wide variety of physical examples. In particular, for infinite-dimensional systems such as coupled harmonic and anharmonic systems (e.g., in superconducting quantum processors), the abundance of both engineered and spurious resonances motivates the use of other techniques. Moreover, even when perturbation theory is applicable, the number of terms in the expansions grows exponentially as the perturbation level and therefore is not practically usable in many instances.

In this article, we introduce a new symbolic algorithm, Non-Perturbative Analytical Diagonalization (NPAD), that allows the computation of closed-from, parametric effective Hamiltonians in a finite-dimensional Hilbert space with a guarantee for convergence. The method makes use of the Jacobi iteration and recursively applies Givens rotations to remove all unwanted couplings. In the perturbative limit, it reduces via BCH expansion to a variant of SWT, which we refer to as the Recursive Schrieffer-Wolff Transformation (RSWT). For this method, the number of commutators grows only linearly with respect to the perturbation order, in contrast to the exponential growth in the traditional approach. Both methods can be used in low-order expansions to provide compact analytical expressions of effective Hamiltonians; or, alternatively, higher-order expansions that allow for fast parametric design [83] and tuning [84] of effective Hamiltonian models (and, e.g., subsequent automatic differentiation). As illustrated in Figure 3.1, with the two methods, one can tune the system for engineered decoupling or enhanced controlled coupling.

The key insight of our work is that the iteration step in forming the effective model can be applied recursively, i.e. after each step the transformed Hamiltonian

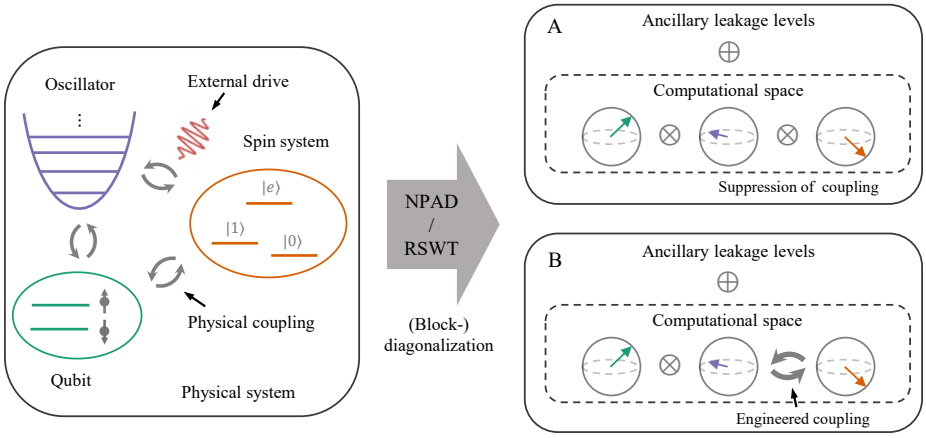


Figure 3.1: Illustration of generating an effective Hamiltonian model from a given physical model. The left-hand side shows the physical system composed of several different quantum subsystems and possible coupling among them. External controls may also exist and drive the system dynamics. The methods introduced in this article (NPAD and RSWT) can be used to compute the effective model (right-hand side) where undesired interactions are effectively removed (block A) and engineered couplings are enhanced (block B). The dynamics can then be studied in the computational subspace.

is viewed as a new starting point and determines the next step. Moreover, each step can act on a chosen single state-to-state coupling at a time, thereby providing an exact elimination of the term. In this regard, this can be understood as a generalization of the well-known numerical Jacobi iteration used for diagonalization of real symmetric matrices [85], which has also found use for Hermitian operators [86, 87]. Similar ideas have also been widely used in the orbital localization problem [88].

As demonstrations of the practical utility of the methods, we study superconducting qubits, which are especially relevant for robust parametric design methods, not only because they are prone to spurious resonances [89–91], but because they can be readily fabricated across a very wide range of energy scales [92, 93].

We investigate both the near-resonant regime and in the quasi-dispersive regime, focusing on the ZZ and cross-resonance interaction. In the near-resonant regime, we consider the two-excitation manifold and compute accurate approximations of the ZZ interaction strength applicable to the full parameter regime for gate implementation [94–97]. In the second scenario, we study the suppression of ZZ interactions [83, 98–113] in the traditional setup of resonator mediated coupling without direct qubit-qubit interaction. The result shows that the ZZ interaction can be suppressed without resorting to additional coupling in a regime where the qubit-resonator detuning is comparable to the qubit anharmonicity, described by an

equation of a circle. Extending the applications to block diagonalization, we then compute the coupling strength of a microwave-activated cross-resonant interaction. We show that, with only 4 Givens rotations, we can diagonalize the drive and achieve accurate estimation in the regime where the perturbation method fails.

This chapter is organized as follows: In Section 3.2, we present the mathematical methods, NPAD and RSWT, for diagonalization and obtaining effective Hamiltonian models. We also briefly discuss generalizing the two methods to block diagonalization in Section 3.2.3. Next, in Section 3.3, we demonstrate the applications to superconducting systems. We study the ZZ interaction for generating entanglement in the near-resonant regime (Section 3.3.1), and in the (quasi-) dispersive regime for suppressing cross-talk noise (Section 3.3.2). The computation of the cross-resonance coupling strength is presented in Section 3.3.3. We conclude and give an outlook of other possible applications in Section 3.4.

3.2 Mathematical methods

3.2.1 Non-perturbative Analytical Diagonalization

In this subsection, we introduce the NPAD for symbolic diagonalization of Hermitian matrices and discuss how it can be applied to obtain effective models.

In this algorithm, a Givens rotation is defined in each iteration to remove one specifically targeted off-diagonal term. By iteratively applying the rotations, the transformed matrix converges to the diagonal form. The rotation keeps the energy structure when the off-diagonal coupling is small while always exactly removing the coupling even when it is comparable to or larger than the energy gap. Compared to the Jacobi method used in numerical diagonalization [85–87], we truncate the iteration at a much earlier stage. As each iteration consists only of a few algebraic expressions, the algorithm produces a closed-form, parametric expression of the transformed matrix.

We start from a two-by-two Hermitian matrix and define a complex Givens rotation that diagonalizes it. Then, we generalize the rotation to higher-dimensional matrices, discuss the convergence of the iteration, and how to use it as a symbolic algorithm. In Section 3.3.1, we show a concrete application where we apply NPAD with only two rotations to approximate the energy spectrum of a near-resonant quantum system which can not be studied perturbatively.

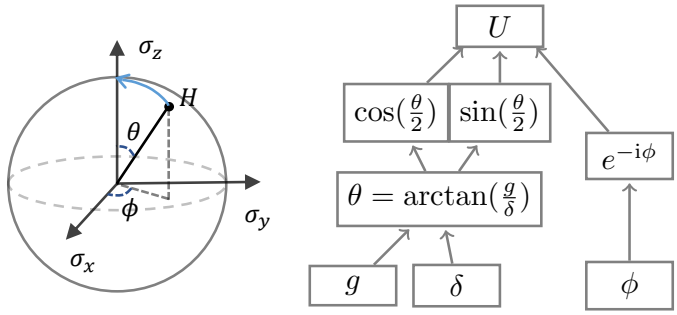
Givens rotations

We consider a two-by-two Hermitian matrix

$$\hat{H} = \begin{pmatrix} \varepsilon + \delta & g\mathrm{e}^{-i\phi} \\ g\mathrm{e}^{i\phi} & \varepsilon - \delta \end{pmatrix}, \tag{3.1}$$

where g, ϕ, ε and δ are real numbers. The matrix can be decomposed in the Pauli basis as

$$\hat{H} = \epsilon\hat{I} + \delta\hat{\sigma}_z + g\left(\cos(\phi)\hat{\sigma}_x + \sin(\phi)\hat{\sigma}_y\right) \tag{3.2}$$



(a) (b)

Figure 3.2: **(a):** The Givens rotation illustrated on a Bloch sphere. A Hermitian matrix defined in equation (3.1) is denoted as a point on the surface of a Bloch sphere with the radius $\sqrt{\delta^2 + g^2}$. This is different from the Bloch sphere representation of a quantum state, where the radius is always smaller than or equal to 1. The coordinates correspond to the coefficients in the representation in the Pauli basis. The Givens rotation $\hat{U}$ that diagonalizes the matrix can be viewed as a rotation denoted by the blue arrow (for $\delta \geq 0$). **(b):** The computational graph of the Givens rotation $\hat{U}$, defining the main mathematical steps in the symbolic algorithm 1. The inputs g, δ and ϕ can be directly extracted from the Hamiltonian.

which can be illustrated in a Bloch sphere with the radius $\sqrt{\delta^2 + g^2}$ (omitting the identity) as shown in Figure 3.2a. Without loss of generality, we assume that $g \geq 0$ and absorb the sign into the complex phase.

The diagonalization can be understood as a rotation on the Bloch sphere to the North or South pole. In particular, if $\delta \geq 0$, it is rotated to the North pole, and otherwise to the South pole, avoiding unnecessarily flipping the energy level during the diagonalization. This rotation is performed around the axis $\hat{n} = \cos(\phi)\hat{\sigma}_y - \sin(\phi)\hat{\sigma}_x$ with the angle $\theta = \arctan\left(\frac{g}{\delta}\right)$. As an illustration, for $\delta \geq 0$, the rotation is denoted by a blue arrow in Figure 3.2a.

The unitary transformation that diagonalizes the matrix is given by

$$\hat{U} = \exp\left[\hat{S}\right] = \exp\left[\frac{i}{2}\theta\hat{n}\right] = \begin{pmatrix} \cos\left(\frac{\theta}{2}\right) & \mathrm{e}^{-i\phi}\sin\left(\frac{\theta}{2}\right) \\ -\mathrm{e}^{i\phi}\sin\left(\frac{\theta}{2}\right) & \cos\left(\frac{\theta}{2}\right) \end{pmatrix}, \tag{3.3}$$

where $\hat{S} = \frac{i}{2}\theta\hat{n}$ is referred to as the generator of the rotation. The transformation satisfies $\Lambda = \hat{U}\hat{H}\hat{U}^\dagger$ with Λ the diagonalized matrix. We refer to $\hat{U}$ as a Givens rotation [114]. Notice that in most literature, the Givens rotation is defined with $\phi = 0$. Here we use this more general (Hermitian) definition as it shares many common properties.

The computation of the unitary consists only of elementary mathematical functions, as illustrated in Figure 3.2b. This is critical for it to be used as a building block for a symbolic algorithm. As we will see later, by concatenating this building

block, a parameterized expression can be generated for an arbitrary Hermitian matrix.

Simplified formulation

In practice, the inverse trigonometric function in the expression of θ is often avoided by using the trigonometric identities

$$\tan(\theta) = \frac{2t}{1-t^2} \tag{3.4}$$

with $t = \tan\left(\frac{\theta}{2}\right)$. We then rewrite equation (3.4) as

$$t^2 + 2t/\kappa - 1 = 0 \tag{3.5}$$

with $\kappa = g/\delta$. We choose the root with smaller norm for the convenience that the rotation will not flip the two energy levels [1]:

$$t = \frac{\sqrt{\kappa^2+1}-1}{\kappa}. \tag{3.6}$$

In this way, the parameters $\cos\left(\frac{\theta}{2}\right)$ and $\sin\left(\frac{\theta}{2}\right)$ in the Givens rotation can be calculated directly from g and δ using algebraic expressions. It is also evident in equation (3.6) that the rotation angle is bounded by $|\theta| \leq \pi/2$.

The iterative method

We now apply the Givens rotation to remove the (j,k)-th entry of a general Hermitian matrix $\hat{H}$. The parameters are chosen to be consistent with equation (3.1), i.e., $\delta_{jk} = (H_{j,j} - H_{k,k})/2$ and $g_{jk}\mathrm{e}^{-i\phi_{jk}} = H_{j,k}$. For simplicity, we use the notation $c_{jk} = \cos\left(\frac{\theta_{jk}}{2}\right)$, $s_{jk} = \sin\left(\frac{\theta_{jk}}{2}\right)$, and $t_{jk} = s_{jk}/c_{jk}$. We write the Givens rotation $\hat{U}_{jk}$ as

$$\hat{U}_{jk} = \begin{pmatrix} 1 & & & & & & \\ & \ddots & & & & & \\ & & c_{jk} & \cdots & \mathrm{e}^{-i\phi_{jk}}s_{jk} & & \\ & & \vdots & \ddots & \vdots & & \\ & & -\mathrm{e}^{i\phi_{jk}}s_{jk} & \cdots & c_{jk} & & \\ & & & & & \ddots & \\ & & & & & & 1 \end{pmatrix} \tag{3.7}$$

[1] In numerical implementation, it is often written as $t = \frac{\mathrm{sgn}(\kappa)}{|1/\kappa| + \sqrt{1/\kappa^2+1}}$ for numerical stability when $\kappa \to 0$.

where the diagonal elements are all 1 except for two entries (j, j) and (k, k). All other entries not explicitly defined are 0.

Applying this unitary transformation with $\hat{H}' = \hat{U}_{jk}\hat{H}\hat{U}_{jk}^{\dagger}$ eliminates the off-diagonal entry $H_{j,k}$, i.e., $|H'_{j,k}| = |H'_{k,j}| = g'_{jk} = 0$. It renormalizes the energies such that

$$\delta'_{jk} = \delta_{jk} + t_{jk}g_{jk} \tag{3.8}$$

However, this will also mix other entries on the j, k-th rows and columns, given by

$$H'_{h,j} = c_{jk}H_{h,j} + \mathrm{e}^{i\phi_{jk}}s_{jk}H_{h,k} \tag{3.9}$$

$$H'_{h,k} = c_{jk}H_{h,k} - \mathrm{e}^{-i\phi_{jk}}s_{jk}H_{h,j} \tag{3.10}$$

with $h \neq j, k$.

One can diagonalize the matrix by applying the rotation $\hat{U}_{jk}$ with the corresponding parameters iteratively on the largest remaining non-zero off-diagonal entry, which is referred to as the Jacobi iteration [85]. That is, we can iteratively solve for the eigenenergies by picking the next largest off-diagonal element, e.g., $H'_{j',k'} = g'_{j'k'}\mathrm{e}^{-i\phi'_{j'k'}}$, and applying another Givens rotation, as summarized in Algorithm 1.

Algorithm 1: Non-Perturbative Analytical Diagonalization (NPAD)

input : a Hermitian matrix $\hat{H}_0$
output : an effective model $\hat{H}'$
$\hat{H} \leftarrow \hat{H}_0$;
while $\|\hat{H} - \mathrm{diag}(\hat{H})\| >$ tolerance **do**

1. find the target coupling $H_{j,k}$;
2. define δ_{jk}, g_{jk} and ϕ_{jk} such that $\delta_{jk} = (H_{j,j} - H_{k,k})/2$ and $g_{jk}\mathrm{e}^{-i\phi_{jk}} = H_{j,k}$;
3. $\theta_{jk} \leftarrow \arctan\left(\frac{g_{jk}}{\delta_{jk}}\right)$;
4. $c_{jk} \leftarrow \cos\left(\frac{\theta_{jk}}{2}\right)$, $s_{jk} \leftarrow \sin\left(\frac{\theta_{jk}}{2}\right)$;
5. define $\hat{U}$ according to equation (3.7);
6. $\hat{H} \leftarrow \hat{U}\hat{H}\hat{U}^{\dagger}$;

end
$\hat{H}' \leftarrow \hat{H}$

In practice, the above definition of the Jacobi iteration can be relaxed. For instance, the next target does not always have to be the largest element. In fact, the order of the rotations does not affect the convergence, as long as all terms are

covered in the iteration rules (e.g., cyclic iterations on all off-diagonal entries) [86]. However, performing the rotation first on large elements usually increases the convergence rate. This can be seen by studying the norm of all off-diagonal terms $\|\hat{H}\|_F = \sum_{m\neq n} |H_{m,n}|^2$. Since we have $|H'_{h,j}|^2 + |H'_{h,k}|^2 = |H_{h,j}|^2 + |H_{h,k}|^2$ for $h \neq j, k$ and $\hat{H}'_{j,k} = 0$, each Givens rotation reduces the norm of all off-diagonal terms:

$$\|\hat{H}'\|_F = \|\hat{H}\|_F - 2|H_{j,k}|^2. \tag{3.11}$$

If (j, k) is chosen so that $|H_{j,k}|^2$ is larger than the average norm among the off-diagonal terms, one obtains [86]

$$\|\hat{H}'\|_F = (1 - \frac{2}{N(N-1)})\|\hat{H}\|_F \tag{3.12}$$

where $N(N-1)$ is the total number of off-diagonal terms. Therefore, the algorithm converges exponentially. Moreover, if the off-diagonal terms are much smaller than the energy gap, the convergence becomes even faster, i.e., exponentially fast with a quadratic convergence rate [87]. This leads to an efficient variant of the Schrieffer-Wolff-like methods, as described in Section 3.2.2.

From the above analysis, we also see that the Givens rotation does not have to exactly zero the target coupling. Instead, it only needs to reduce the total norm. Therefore, if the structure of the Hamiltonian is known, rotations can be grouped such that all rotations within one group are constructed from the same Hamiltonian and then applied recursively. We will also explore this possibility in concrete examples later in the article.

As a machine-precision, numerical diagonalization algorithm, the Jacobi iteration is slower than the QR method for dense matrices. However, in many problems in quantum engineering, the Hamiltonian is often sparse and it is known in advance which interaction needs to be removed. It is not always necessary to compute the fully diagonalized matrix but only to transform it into a frame where the target subspace is sufficiently decoupled from the leakage levels. Therefore, an iterative method where each step is targeted at one off-diagonal entry is of particular interest.

As a symbolic method, we can truncate the Jacobi iteration at a very early stage to obtain closed-formed parametric expressions. It will also correctly calculate the renormalized energy and other couplings while keeping the energy structure in the perturbative limit, as will be discussed in Section 3.2.2.

3.2.2 Recursive Schrieffer-Wolff perturbation method

In the previous subsection, we introduced NPAD which produces a closed-form, parametric expression of an approximately diagonalized matrix. Here, we show that in the perturbative limit, where the coupling is much smaller than the bare energy difference, the Jacobi iteration reduces to a Schrieffer-Wolff-like transformation. Interestingly, the recursive nature of the Jacobi iteration is preserved in this limit. Instead of looking for one generator that diagonalizes the full matrix as in the traditional Schrieffer-Wolff transformation (SWT), an iteration is constructed

such that every time only the leading-order coupling is removed. We refer to it as recursive Schrieffer-Wolff transformation (RSWT) because of the recursive expression it produces. We also show that RSWT demonstrates an exponential improvement in complexity compared to SWT for perturbation beyond the leading order. In Section 3.3.2, we demonstrate an application of RSWT in estimating the ZZ interaction between two Transmon qubits in a dispersive regime.

Givens rotation in the perturbative limit

In the perturbative limit, compared to $\hat{U}_{jk}$ in equation (3.7), it is more convenient to specify the generator defined in equation (3.3). For the Givens rotation $\hat{U}_{jk}$ the corresponding generator $\hat{S}'$ has two non-zero entries

$$S'_{j,k} = -{S'_{k,j}}^* = H_{j,k}/(H_{j,j} - H_{k,k}), \tag{3.13}$$

all other entries being 0. In addition, assuming that we only aim at removing the leading-order off-diagonal terms, we define a generator

$$\hat{S} = \sum_{p\in\mathcal{P}} \hat{S}'_p \tag{3.14}$$

where the sum over $\mathcal{P}$ denotes all pairs of non-zero off-diagonal entries in $\hat{H}$. The assumption of perturbation indicates that $\|\hat{S}\|_F \ll 1$. In this case, the unitary generated by $\hat{S}$ still eliminates all the leading-order coupling because

$$\exp\big(\hat{S}\big) = \exp\left(\sum_{p\in\mathcal{P}} \hat{S}'_p\right) = \prod_{p\in\mathcal{P}} \mathrm{e}^{\hat{S}'_p} + \mathcal{O}\left(\|\hat{S}\|_F^2\right). \tag{3.15}$$

This generator $\hat{S}$ is identical to the generator of the leading-order SWT. One can verify that $[\hat{S}, \hat{D}] = -\hat{V}$ where $\hat{D}$ and $\hat{V}$ are the diagonal and off-diagonal parts of $\hat{H}$. By expanding the transformation $\mathrm{e}^{\hat{S}}\hat{H}\mathrm{e}^{-\hat{S}}$ using the BCH formula

$$\hat{H}' = \mathrm{e}^{\hat{S}}\hat{H}\mathrm{e}^{-\hat{S}} = \hat{H} + [\hat{S}, \hat{H}] + \frac{1}{2!}[\hat{S}, [\hat{S}, \hat{H}]] + \cdots \tag{3.16}$$

and truncating the series at $\mathcal{O}(\|\hat{S}\|_F^2)$, one obtains the leading-order SWT.

The difference between RSWT and SWT appears when one considers higher-order perturbation. In SWT, one expands the transformed Hamiltonian $\hat{H}'$ and the generator $\hat{S}$ perturbatively as a function of a small parameter and collects terms of the same order on both sides of equation (3.16). However, here, the generator is predefined and it only eliminates the leading-order coupling. Similar to the Jacobi iteration, we treat the transformed Hamiltonian $\hat{H}'$ as a new Hermitian matrix and perform another round of leading-order transformation as the next iteration. This results in a recursive expression for $\hat{H}'$, which is still a closed-form expression. The remaining off-diagonal terms can always be removed by the next iteration if the truncation level of the BCH formula is high enough to guarantee sufficient accuracy. We present the iteration of RSWT in detail in the next subsection and show that it simplifies the calculation for perturbation beyond the leading order.

The RSWT iterations

In the following, we outline the iterative procedure of the RSWT. We denote the initial matrix $\hat{H}$ as step zero, with the notation $\hat{D}_0 = \hat{D}$, $\hat{V}_0 = \lambda\hat{V}$ and $\hat{H}_0 = \hat{H} = \hat{D}_0 + \hat{V}_0$. The parameter λ is the dimensionless small parameter used to track the perturbation order. Assume that we want to compute the perturbation to the eigenenergy up to the order λ^K. We refer to this as the λ^K-perturbation. Given the Hamiltonian of iteration n, $\hat{H}_n$, we can compute the next iteration $\hat{H}_{n+1}$ as follows.

We first define a generator $\hat{S}_{n+1}$ according to equation (3.14) such that

$$[\hat{S}_{n+1}, \hat{D}_n] = -\hat{V}_n,$$

where $\hat{D}_n$ and $\hat{V}_n$ are the diagonal and the off-diagonal part of $\hat{H}_n$. As the energy gap $\hat{D}_n$ always stays at $\mathcal{O}(\lambda^0)$ under the assumption of small perturbation, $\hat{S}_{n+1}$ is of the same order as $\hat{V}_n$. Notice that $\hat{S}_{n+1}$ is generated from the perturbed matrix in the previous iteration, $\hat{H}_n$, in contrast to the unperturbed matrix as in SWT.

Then, the next level of perturbation is computed with

$$\hat{H}_{n+1} = \sum_{t=0}^{m} \frac{1}{t!}\mathcal{C}_t(\hat{S}_{n+1}, \hat{D}_n) + \sum_{t=0}^{m-1} \frac{1}{t!}\mathcal{C}_t(\hat{S}_{n+1}, \hat{V}_n) \tag{3.17}$$

where $\mathcal{C}$ is the nested commutator defined by

$$\mathcal{C}_{t+1}(\hat{A}, \hat{B}) = [\hat{A}, \mathcal{C}_t(\hat{A}, \hat{B})] \tag{3.18}$$

and $\mathcal{C}_0(\hat{A}, \hat{B}) = \hat{B}$. The truncation level m of the BCH expansion will be defined explicitly later. Because $[\hat{S}_{n+1}, \hat{D}_n] = -\hat{V}_n$ by construction, we have for all n and t

$$\mathcal{C}_{t+1}(\hat{S}_{n+1}, \hat{D}_n) = -\mathcal{C}_t(\hat{S}_{n+1}, \hat{V}_n). \tag{3.19}$$

Therefore, plugging in equation (3.19) into equation (3.17) simplifies it to

$$\hat{H}_{n+1} = \hat{D}_n + \sum_{t=1}^{m-1} \frac{t}{(t+1)!}\mathcal{C}_t(\hat{S}_{n+1}, \hat{V}_n). \tag{3.20}$$

Notice that t starts from 1 in the sum, which means that all coupling terms at the same order of $\hat{V}_n$ are removed and the order of the remaining coupling, $\hat{V}_{n+1}$, is squared. This iteration is applied until the desired order is reached, as summarized in Algorithm 2.

To ensure that the truncation of the BCH is accurate up to the order $\mathcal{O}(\lambda^K)$, for the nth iteration, we need to choose the truncation $m = \lfloor \frac{K}{2^n} \rfloor$, which ensures that $\hat{H}_{n+1} = \mathrm{e}^{(\hat{S}_{n+1})}\hat{H}_n\mathrm{e}^{(-\hat{S}_{n+1})} + \mathcal{O}(\lambda^{K+1})$. This maximal level m is halved every time the iteration step increases because the remaining coupling is quadratically smaller. This means that, in contrast to SWT, the first iteration has the largest

Algorithm 2: Recursive Schrieffer-Wolff Transformation (RSWT)

input : a Hermitian matrix $\hat{H}_0$
output : $\hat{H}'$ including correction to the eigenenergy up to λ^K
$n_{\max} \leftarrow \lfloor \log_2(K) \rfloor$;
for $n \leftarrow 0;\ n < n_{\max};\ n \leftarrow n+1$ **do**

1. $\hat{D}_n \leftarrow \mathrm{diag}(\hat{H}_n)$; $\hat{V}_n \leftarrow \hat{H}_n - \hat{D}_n$;
2. initialize a zero matrix $\hat{S}_{n+1}$;
   for j,k with $V_{n,j,k} \neq 0$ **do**
   $S_{n+1,j,k} \leftarrow V_{n,j,k}/(D_{n,j,j} - D_{n,k,k})$;
   end
3. $m \leftarrow \lfloor \frac{K}{2^n} \rfloor$;
   $\hat{H}_{n+1} \leftarrow \hat{D}_n + \sum_{t=1}^{m-1} \frac{t}{(t+1)!} \mathcal{C}_t(\hat{S}_{n+1}, \hat{V}_n)$;

end
$\hat{H}' \leftarrow \hat{H}_{n_{\max}}$

number of terms in RSWT. In Section 3.5.1, we show that, if $\|\hat{S}_{n+1}\| < \frac{1}{2}$, the error of the truncation in equation (3.20) is bounded by

$$\left\|\hat{H}_{n+1} - \hat{H}^{\infty}_{n+1}\right\| \leq \frac{2^m}{m!}\left\|\hat{S}_{n+1}\right\|^m \left\|\hat{V}_n\right\| \tag{3.21}$$

where the $\hat{H}^{\infty}_{n+1}$ is equation (3.20) in the limit $m \to \infty$.

3.2.3 Block diagonalization

Both the NPAD and the RSWT methods introduced in the previous sections can be designed to only target a selected set of off-diagonal terms and, hence, used for block-diagonalization. This is especially useful to engineer transversal coupling in a subsystem and leave the remaining levels as intact as possible. Here, we briefly discuss these generalizations. Notice that it is always possible to first diagonalize the matrix and then reconstruct the block diagonalized form that satisfies certain conditions, for instance as in Ref. [115]. In the following, we discuss only methods that do not diagonalize the matrix first.

In NPAD, by construction, each rotation removes one off-diagonal element. With Givens rotations only applied to the inter-block elements, an iteration for block diagonalization can be defined. The norm of all off-diagonal entries, $\|\hat{H}\|_F$, is still monotonously decreasing according to equation (3.11). Hence, a limit exists and its convergence is also the convergence of the block diagonalization. However, the convergence is not always monotonous with respect to the norm of all inter-block terms. This is because a Givens rotation may rotate a large intra-block term into

an inter-block entry. Therefore, the algorithm may not always converge faster than the full diagonalization would. Nevertheless, if the dominant coupling terms in the Hamiltonian are known, the Jacobi iteration can be designed to target those to realize an efficient block-diagonalization. In Section 3.3.3, we show an example of this in computing the cross-resonance coupling strength through NPAD.

For perturbation, RSWT can be applied as a block diagonalization method under the constraint that both the inter-block and the intra-block coupling are much smaller than the inter-block energy gap. This can be achieved by slightly modifying the RSWT iterations: We first separate the diagonal, the intra-block and the inter-block terms: $\hat{H}_n = \hat{D}_n + \hat{V}_n^{\text{intra}} + \hat{V}_n^{\text{inter}}$. Next, in Algorithm 2 we only define $\hat{S}$ for those non-zero entries in $\hat{V}_n^{\text{inter}}$, i.e. the couplings we wish to remove. And in the last step, we replace equation (3.20) with

$$\begin{aligned}\hat{H}_{n+1} = \hat{D}_n &+ \sum_{t=1}^{m-1} \frac{t}{(t+1)!} \mathcal{C}_t(\hat{S}_{n+1}, \hat{V}_n^{\text{inter}}) \\ &+ \sum_{t=0}^{m} \frac{1}{t!} \mathcal{C}_t(\hat{S}_{n+1}, \hat{V}_n^{\text{intra}}).\end{aligned} \tag{3.22}$$

In this definition, the leading interblock coupling is of the order $\mathcal{O}([\hat{S}_{n+1}, \hat{V}_n^{\text{intra}}])$. As we do not remove the intra-block coupling, we still get $\hat{V}_n^{\text{intra}} = \mathcal{O}(\lambda)$. Therefore, the remaining coupling is $\mathcal{O}(\lambda \hat{V}_n^{\text{intra}})$, i.e. the perturbation order is increased by one, instead of being squared as in the case of full diagonalization. Therefore, the exponential reduction of the number of commutators does not always apply in the case of block diagonalization. However, notice that the small parameter λ here is defined as the (largest) ratio between the inter-block couplings and gaps, which is usually much smaller than those within the block. Hence, if carefully designed, the convergence can still surpass the full diagonalization in the first few perturbative orders.

3.2.4 Comparison between different methods

To help understand the proposed methods, we here discuss the difference between them and the traditional methods. We first compare RSWT with traditional SWT and then NPAD with the perturbation methods.

For RSWT, with the same target accuracy, e.g., $\mathcal{O}(\lambda^K)$, it should provide the same expression as from SWT, up to the error $\mathcal{O}(\lambda^{K+1})$. However, compared to the SWT, RSWT requires a much smaller number of iterations and commutators. To reach $\mathcal{O}(\lambda^K)$, SWT needs $K-1$ iterations, while RSWT only needs $\lfloor \log_2(K) \rfloor$ because of the quadratic convergence rate. More importantly, the total number of commutators grows only linearly for RSWT, compared to the exponentially fast growth for SWT [80].

Intuitively, this is because RSWT uses the recursive structure and avoids unnecessary expansions of the intermediate results. Mathematically, this can be

seen from the following two aspects: First, in RSWT, each iteration improves the perturbation level from λ^k to λ^{2k}, instead of λ^{k+1}. Hence, the number of iterations increases only logarithmically with respect to the perturbation order, as seen in the definition of $n_{\max}$ in Algorithm 2. This is because we always treat the transformed matrix as a new one and remove the leading-order coupling. It is consistent with the quadratic convergence rate of the Jacobi iteration with small off-diagonal terms. Second, in RSWT, the generator $\hat{S}_n$ is only used at the current iteration. Hence, there are no mixed terms such as $[\hat{S}_2, [\hat{S}_1, \hat{V}_0]]$, in contrast to SWT.

The total number of commutators required to reach level λ^K is shown in Table 3.1, where we have taken into consideration that if $\mathcal{C}_t(\hat{A}, \hat{B})$ is known, computing $\mathcal{C}_{t+1}(\hat{A}, \hat{B})$ only requires one additional commutator. The detailed calculation is presented in Section 3.5.2.

The NPAD method, on the other hand, uses non-linear rotations to replace the linear perturbative expansion. More concretely, in the Jacobi iteration, by targeting only one coupling in each recursive iteration, the unitary transformation can be analytically expressed as a Givens rotation, thus avoiding the BCH expansion in equation (3.16). Therefore, it efficiently and accurately captures the non-perturbative interactions in the system.

To compare it with the perturbation methods, we estimate the number of operations required for NPAD in the perturbative regime. Assume we construct the Jacobi iteration from the G coupling terms used in generating an $\hat{S}$ in RSWT. Applying those unitaries is, to the leading order, the same as applying one RSWT iteration. A single Givens rotation on a Hamiltonian takes $\mathcal{O}(N)$ operations, where N is the matrix size. Thus, the cost for computing the effective Hamiltonian after G rotations is the same as computing one commutator $[\hat{S}, \cdot]$, up to a constant factor. Because the Givens rotation avoids the BCH expansion, there are no nested commutators and the total number of operations is $\mathcal{O}(n_{\max} NG)$ with $n_{\max}$ the number of iterations in Algorithm 2. Hence the number of terms scales logarithmically with respect to K instead of linearly as for RSWT, i.e., a super-exponential reduction compared to SWT (Table 3.1). However, the non-linear expressions provided by NPAD are usually harder to simplify and evaluate by hand compared to the rational expressions obtained from perturbation.

From the above discussion, one can see that it is also straightforward to combine NPAD with perturbation. Instead of fully diagonalizing the matrix, the Jacobi iteration can be designed to remove only the dominant couplings and combined with perturbation methods to obtain simplified analytical expressions. In fact, this is often used implicitly in the analysis when, e.g., a strongly coupled two-level system is perturbatively interacting with another quantum system. The Jacobi iteration suggests that this can be generalized systematically to more complicated scenarios.

K	2	3	4	5	6	7	8
SWT	1	4	11	26	57	120	247
RSWT	1	2	4	5	7	8	11
NPAD	1	1	2	2	2	2	3

Table 3.1: The number of terms in the evaluation for different methods to reach the λ^K-perturbation. The number denotes the total number of commutators in SWT and RSWT, or the total number of sweeps over all couplings for NPAD. This describes both the "algebraic complexity" (i.e. complexity of the output algebraic expressions) and the computational (time-cost) complexity. The complexity is reduced from exponential to linear and eventually to logarithmic. However, notice that although the computational complexity for one commutator and for one Jacobi sweep scales the same in terms of the number of couplings to be removed (see the main text), the Givens rotation in NPAD consists of non-linear algebraic expressions which are individually more expensive to compute.

3.3 Physical applications

In this section, we use the methods introduced in Section 3.2 to study the ZZ interaction in two different parameter regimes. In a two-qubit system, the ZZ interaction strength is defined by

$$\zeta = E_{11} - E_{10} - E_{01} + E_{00} \tag{3.23}$$

where E_{jk} denotes the eigenenergy of the two-qubit states $|jk\rangle$. The Hamiltonian interaction term is written as $\zeta\sigma_{z_1}\sigma_{z_2}$, acting on the two qubits. Typically, in superconducting systems, it arises from the interaction of the $|11\rangle$ state with the non-computational state in the physical qubits, and can both be used as a resource for entangling gates [94–97] or viewed as cross-talk noise that needs to be suppressed [98–113] .

3.3.1 Effective ZZ entanglement from non-dispersive interactions

In this first application, we apply the NPAD method described in Section 3.2.1 to study a model consisting of two directly coupled qubits in the near-resonant regime, where the ZZ interaction can be used to construct a control-Z (CZ) gate (see Figure 3.1 block B) [94–97]. We show that, with two rotations, NPAD provides an improvement on the estimation of the interaction strength for at least one order of magnitude, compared to approximating the system as only a single avoided crossing between the strongly interacting levels, as is standard in the literature. In addition, if one of the non-computational bases is comparably further detuned than the other, the correction takes the form of a Kerr nonlinearity, with a renormalized coupling strength accounting for the near-resonant dynamics.

We consider the Hamiltonian of two superconducting qubits that are directly coupled under the rotating-wave [67, 116] and Duffing [117] approximations:

$$\hat{H} = \sum_{q\in\{1,2\}} \omega_q \hat{b}_q^\dagger \hat{b}_q + \frac{\alpha_q}{2} \hat{b}_q^\dagger \hat{b}_q^\dagger \hat{b}_q \hat{b}_q + g(\hat{b}_1 \hat{b}_2^\dagger + \hat{b}_1^\dagger \hat{b}_2) \tag{3.24}$$

where $\hat{b}_q$, ω_q, α_q are the annihilation operator, the qubit bare frequency and the anharmonicity, respectively. The parameter g denotes the coupling strength. In this Hamiltonian, the sum of the eigenenergies is always a constant $E_{10} + E_{01} = \omega_1 + \omega_2$ because of the symmetry. Hence, the ZZ interaction comes solely from the interaction between the state $|11\rangle$ and the non-computational basis $|20\rangle$ and $|02\rangle$. If the frequency is tuned so that the state $|11\rangle$ is close to one of the non-computational states, the coupling will shift the eigenenergy, leading to a large ZZ interaction (Figure 3.3a).

For simplicity, we consider the Hilbert subspace consisting of $|20\rangle$, $|11\rangle$, $|02\rangle$ and write the following Hamiltonian

$$\hat{H} = \begin{pmatrix} \delta & g_1 & 0 \\ g_1 & -\delta & g_2 \\ 0 & g_2 & -\Delta \end{pmatrix}. \tag{3.25}$$

The parameters in the diagonal elements are given by $\delta = (\omega_1 - \omega_2 + \alpha_1)/2$ and $\Delta = 3(\omega_1 - \omega_2)/2 - \alpha_2 + \alpha_1/2$. To keep the result general, we use two different coupling strengths g_1 and g_2, although according to equation (3.24) they both equal $\sqrt{2}g$. Without loss of generality, we assume the state $|02\rangle$ is comparably further detuned from the other two, i.e. $\Delta > g_j, \delta$. If in contrast $|20\rangle$ is further detuned, one can exchange the $|02\rangle$ and $|20\rangle$ in the matrix and redefine δ and Δ accordingly. Notice that this Hamiltonian is different from a Λ system [76], where coupling exists only between far-detuned levels.

To implement the CZ gate, one tunes the qubit frequency ω_1 so that the states $|11\rangle$ and $|20\rangle$ are swept from a far-detuned to a near-resonant regime. Hence, the perturbative expansion diverges and cannot be used. A naive approach is to neglect the far-detuned state $|02\rangle$ and approximate the interaction as a single avoided crossing. In this case, ζ is approximated by

$$\zeta_{\text{2-level}} \approx \delta - \delta\sqrt{1 + \frac{g_1^2}{\delta^2}}. \tag{3.26}$$

However, the interaction g_2 results in an error that, in the experimentally studied parameter regimes, can be as large as 10%, as shown in Figure 3.3b.

In the following, we show that with only two Givens rotations, one can obtain an analytical approximation, with the error reduced by one order of magnitude. The correction can be understood as a Kerr non-linearity with a renormalized coupling strength.

To get an accurate estimation of the ZZ interaction ζ, we need to calculate the eigenenergy of $|11\rangle$ by eliminating its coupling with the other two states. Therefore,

we will make two rotations sequentially on the entry $(0,1)$ and $(1,2)$, given by

$$\hat{H}^{(2)} = \hat{U}_2 \hat{H}^{(1)} \hat{U}_2^\dagger = \hat{U}_2 \hat{U}_1 \hat{H} \hat{U}_1^\dagger \hat{U}_2^\dagger, \tag{3.27}$$

where $\hat{U}_1$ and $\hat{U}_2$ are Givens rotations (equation (3.3)) constructed for eliminating the entries $(0,1)$ and $(1,2)$. Because the matrix is real symmetric, the phase ϕ in equation (3.3) is 0.

The first transformed Hamiltonian, $\hat{H}^{(1)} = \hat{U}_1 \hat{H} \hat{U}_1^\dagger$, takes the form

$$\hat{H}^{(1)} = \begin{pmatrix} E_2 & 0 & g_2 s_{01} \\ 0 & -E_2 & c_{01} g_2 \\ g_2 s_{01} & c_{01} g_2 & -\Delta \end{pmatrix} \tag{3.28}$$

where $E_2 = \delta \sqrt{1 + \frac{g_1^2}{\delta^2}}$ is the eigenenergy for diagonalizing the two-level system of $|20\rangle$ and $|11\rangle$, consistent with equation (3.26). The notations used are the same as in Section 3.2.1. In this frame, the coupling between $|11\rangle$ and $|02\rangle$ is reduced to $c_{01} g_2$, where c_{01} is given by the non-linear expression

$$c_{01} = \frac{1}{\sqrt{\left(\frac{E_2 - \delta}{g_1}\right)^2 + 1}}. \tag{3.29}$$

This non-linearity is crucial for the accurate estimation of the eigenenergy.

The second rotation further removes this renormalized coupling $c_{01} g_2$, giving

$$\hat{H}^{(2)} = \begin{pmatrix} E_2 & g_2 s_{01} s_{12} & c_{12} g_2 s_{01} \\ g_2 s_{01} s_{12} & -E_2 + g_2 c_{01} t_{12} & 0 \\ c_{12} g_2 s_{01} & 0 & -\Delta - g_2 c_{01} t_{12} \end{pmatrix}. \tag{3.30}$$

Including the new correction, $g_2 c_{01} t_{12}$, the eigenenergy of state $|11\rangle$ reads

$$H_{1,1}^{(2)} = -E_2 + \frac{\Delta - E_2}{2} \left(\sqrt{1 + \left(\frac{2 c_{01} g_2}{\Delta - E_2} \right)^2} - 1 \right). \tag{3.31}$$

In Figure 3.3b, we plot the error of the estimated interaction strength $\zeta = H'_{1,1} + \delta$ using typical parameters of superconducting hardware, compared to the numerical diagonalization $\tilde{\zeta}$. An improvement of at least one order of magnitude is observed compared to traditional methods.

Following the assumptions that $\Delta \gg \delta, g_j$, equation (3.31) simplifies to

$$\boxed{H_{1,1}^{(2)} \approx -E_2 + \frac{c_{01}^2 g_2^2}{\Delta - E_2}}. \tag{3.32}$$

We see that the correction takes the form of a Kerr non-linearity [118], but with a renormalized coupling strength $c_{01} g_2$. This non-linear factor c_{01} accounts for

the dynamics between $|20\rangle$ and $|11\rangle$ in the near-resonant regime. The same effect can be observed in higher levels where similar three-level subspaces exist. This approximation is plotted as a dashed curve in Figure 3.3b.

The error of this estimation comes both from the expansion of the square root in equation (3.31) as well as from the remaining coupling in $\hat{H}^{(2)}$. The former can be approximated by the next order expansion

$$\epsilon_1 \approx \frac{c_{01}^4 g_2^4}{(\Delta - E_2)^3}. \tag{3.33}$$

For the latter, we consider the remaining coupling in $\hat{H}^{(2)}$ between $|20\rangle$ and $|11\rangle$, which reads $g_2 s_{01} s_{12}$. In the limit $\Delta \gg \delta, g_j$, we have $s_{12} \leq \frac{\theta_{12}}{2} \leq \frac{g_2}{\Delta - E_2} \ll 1$, indicating that this coupling is much smaller than the energy difference. Hence, further correction can be estimated by

$$\epsilon_2 \approx \frac{\left(H_{0,1}^{(2)}\right)^2}{|H_{0,0}^{(2)} - H_{1,1}^{(2)}|} \leq \frac{(g_2 s_{01} s_{12})^2}{g_1} \leq \frac{g_2^4 s_{01}^2}{g_1 (\Delta - E_2)^2}. \tag{3.34}$$

The contribution of the other remaining coupling between $|20\rangle$ and $|02\rangle$ is much smaller due to the large energy gap. Since ϵ_2 is one order smaller than the ϵ_1, ϵ_1 will be the dominant error. We plot the region below this error in Figure 3.3b as a shaded background.

For the more general cases without assuming $\Delta \gg \delta, g_j$, it is hard to provide an error estimation due to the non-linearity. However, the result in Figure 3.3b indicates that equation (3.31) still shows a good performance in other parameter regimes commonly used in superconducting hardware, with an error smaller than 3%. We also observe that an improvement for another order of magnitude can be achieved by introducing a third rotation again on the entry $(0, 1)$.

3.3.2 ZZ coupling suppression in the quasi-dispersive regime

In this second example, we use the two methods to investigate the suppression of ZZ cross-talk with the qubit-resonator-qubit setup in the dispersive cQED regime, which corresponds to Figure 3.1 block A. We demonstrate that in the traditional setup without direct inter-qubit coupling, the ZZ interaction defined in equation (3.23) can still be zeroed in a quasi-dispersive regime by engineering the two parameters of qubit-resonator detuning. The zero points are described by an equation of a circle in the λ^4-perturbation. To accurately capture the interaction strength in the quasi-dispersive regime, we also compute with RSWT the λ^6-perturbation and show that the NPAD method with only 8 Givens rotations provides an expression with similar accuracy.

We consider a Hamiltonian of two superconducting qubits connected by a resonator:

$$\hat{H} = \sum_{q \in \{1,2\}} \omega_q \hat{b}_q^\dagger \hat{b}_q + \frac{\alpha_q}{2} \hat{b}_q^\dagger \hat{b}_q^\dagger \hat{b}_q \hat{b}_q + g_q (\hat{b}_q \hat{a}^\dagger + \hat{b}_q^\dagger \hat{a}) + \omega_r \hat{a}^\dagger \hat{a}. \tag{3.35}$$

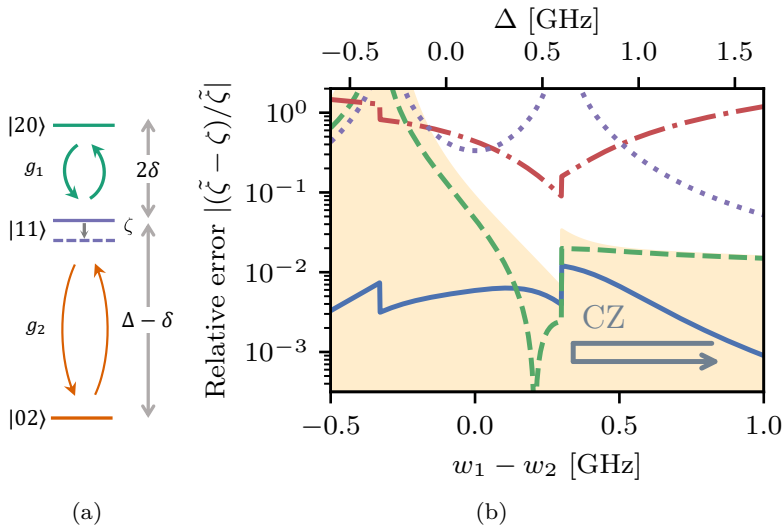


Figure 3.3: **(a):** Interaction and energy level diagram of the two-excitation manifold in the unperturbed Hamiltonian given by equation (3.25). The solid lines represent the bare qubit states, while the arrow and the dashed purple line denote the Stark shift and the eigenenergy of the perturbed $|11\rangle$ state. **(b):** Performance of ZZ interaction estimation using NPAD. We plot the relative difference between the estimated ζ and the value obtained by numerical diagonalization $\tilde{\zeta}$. The estimations are computed with 2 rotations (solid, equation (3.31)), hybrid method with the additional assumption $\Delta \gg \delta, g_j$ (dashed, equation (3.32)), by assuming only a 2-level system (dash-dot, equation (3.26)), and with a leading-order perturbation (dotted). The shaded area covers the region below the error estimation given by equation (3.33). The grey arrow denotes a typical path to generate a CZ gate through ZZ interaction by changing the qubit-qubit detuning. The two jumps are located at $\omega_1 = \omega_2 + \alpha_2$ and $\omega_1 + \alpha_1 = \omega_2$, i.e., the points where the bare energy level swaps. This changes the direction of the Givens rotation. The parameters used are $g_1 = g_2 = \sqrt{2} \cdot 0.1$ GHz and $\alpha_1 = \alpha_2 = -0.3$ GHz.

Due to the finite detuning between the resonator and the qubits, a static ZZ interaction exists even if there is no additional control operation on the system. In order to implement high-quality quantum operations, this interaction needs to be sufficiently suppressed.

Several approaches have been developed to suppress the ZZ interaction. One way is to add a direct capacitive coupling channel in parallel with the resonator [100–108, 119]. By engineering the parameters, the two interaction channels cancel each other. The interaction can either be turned on through a tunable coupler or through the cross-resonant control scheme. The second approach is to choose a hybrid qubit system with opposite anharmonicity, which allows parameter engineering to suppress the ZZ interaction. One implementation is using a transmon and a capacitively shunt flux qubit (CSFQ) [98, 99]. Other methods include using additional off-resonant drive [111–113] and different types of qubits have also been proposed [110].

Most of the above works are based on the strong dispersive regime, where the resonator is only weakly coupled with the qubits. In this regime, the ZZ interaction strength ζ is only determined by the effective interaction with the two non-computational qubit states, $|20\rangle$ and $|02\rangle$ [80]

$$\zeta_{\text{disp}} = -\frac{2J_{20,11}^2}{\Delta_1 - \Delta_2 + \alpha_1} + \frac{2J_{02,11}^2}{\Delta_1 - \Delta_2 - \alpha_2} \tag{3.36}$$

where $\Delta_q = \omega_q - \omega_c$ is the qubit-resonator frequency detuning, α_q the anharmonicity and $J_{jk,j'k'}$ the effective coupling strength between the physical qubit state $|jk\rangle$ and $|j'k'\rangle$. They are obtained by performing a leading order SWT and effectively decoupling the resonator from the two qubits. In this regime, it is impossible to achieve zero ZZ interaction unless the two anharmonicities α_q adopt different signs.

However, equation (3.36) is only valid when ignoring the higher level of the resonator. If we reduce the qubit detuning Δ_q so that it becomes comparable with the anharmonicity α_q, the second excited state of the resonator comes into the picture and can be used to suppress the ZZ interaction, also known as the QUASIDISQ regime [83]. We identify this regime as the quasi-dispersive regime because g/Δ_q is manufactured larger than 0.1, e.g. in superconducting qubits with weak anharmonicity such as Transmons, though we show the same analysis can also hold for stronger anharmonicities. As a result, the calculation of ζ_{disp} cannot be treated by only the leading-order SWT. In particular, we will see that, in the straddling regime, where $|\Delta_1 - \Delta_2| < \alpha$, the interaction with the second excited resonator state leads to a λ^4-perturbative correction that can be used to suppress the ZZ interaction.

In the following, we first use the λ^4-perturbation to qualitatively understand the energy landscape and then investigate the higher-order corrections. For the λ^4-perturbation, using RWST, we only need 2 iterations and evaluate 4 commutators instead of 3 iterations and 11 commutators, as for traditional perturbation (Table 3.1).

In fact, the traditional approach that first approximates the system as an effective qubit-qubit direct interaction and then applies another perturbation to

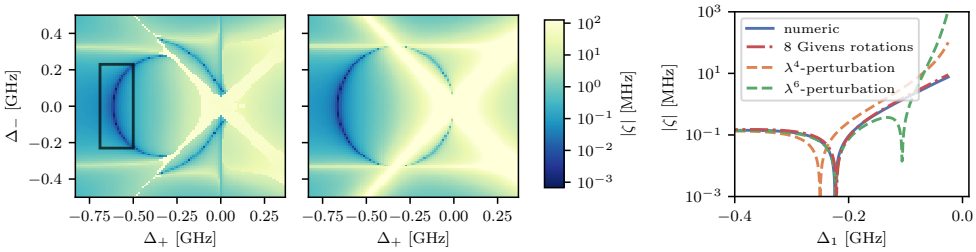


Figure 3.4: **(a)**: The landscape of the ZZ interaction strength $|\zeta|$ as a function of $\Delta_+ = \Delta_1 + \Delta_2$ and $\Delta_- = \Delta_1 - \Delta_2$. **Left**: numerical diagonalization of the so-called QUASIDISQ regime [83]; **Right**: The λ^4-perturbative approximation. In the perturbative approximation, the zero points are described by a circle with a diameter of $2|\alpha|$. The particularly interesting regime is the left part of the circle and away from the resonant line, where the perturbation theory can still be applied, which is marked by the grey rectangle. In the numerical result, the circle is distorted due to the resonant lines and the left half of the circle shrinks because of the higher-order perturbative correction. **(b)**: The numerical result compared to the perturbative correction up to $\lambda^6 = (g/\Delta)^6$ and the Jacobi iteration with 8 two-by-two Givens rotations. Parameters used: $g_1 = g_2 = 0.05$ GHz, $\alpha_1 = \alpha_2 = \alpha = -0.33$ GHz and $\Delta_- = 0.4|\alpha|$.

obtain the ZZ strength is also a two-step recursion [80]. However, for simplicity, it neglects the resonator states in the second perturbation. As detailed in Section 3.5.3, adding the resonator states, we obtain a better estimation for the quasi-dispersive regime. The result is consistent with the diagrammatic techniques used in [98, 120].

To illustrate the energy landscape, we write the interaction strength as

$$\zeta^{(4)} = g_1^2 g_2^2 \left(\frac{1}{\Delta_1^2(\Delta_- - \alpha_2)} - \frac{1}{\Delta_2^2(\Delta_- + \alpha_1)} + \frac{\Delta_1 + \Delta_2}{\Delta_2^2 \Delta_1^2} \right) \tag{3.37}$$

with $\Delta_- = \Delta_1 - \Delta_2$. The first two terms coincide with equation (3.36) in the strong dispersive regime, up to $\mathcal{O}(\frac{g^4}{\Delta^3})$.

Assuming $\alpha = \alpha_1 = \alpha_2$ and set $\zeta^{(4)} = 0$ in equation (3.37), we obtain an equation of a circle that describes the location of the zero points

$$\boxed{(\Delta_+ - \alpha)^2 + \Delta_-^2 - \alpha^2 = 0} \tag{3.38}$$

where $\Delta_+ = \Delta_1 + \Delta_2$ and $\Delta_- = \Delta_1 - \Delta_2$. In this λ^4-perturbation, the zero-points depend only on the anharmonicity α but not on the coupling strength g_q. Equation (3.38) indicates that the ZZ interaction can be suppressed by varying the sum and difference of the two qubit-resonator detunings, as illustrated in Figure 3.4a. Because the perturbative approximation is only valid away from the resonant lines,

the useful part of the parameter regime is the half-circle with $\Delta_+ < \alpha$, in particular, the region marked by the grey box in Figure 3.4a.

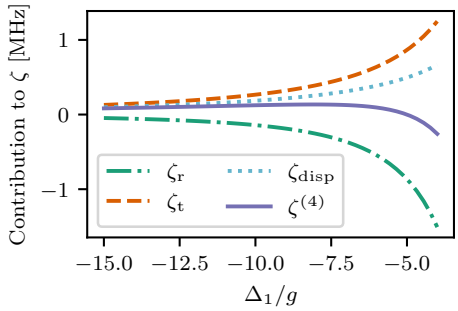
In addition, we also studied different contributions to the ZZ interaction. In Figure 3.5a, we plot the strong dispersive approximation, the λ^4-perturbation as well as the contribution of second excited qubit and resonator state to $\zeta^{(4)}$ (see Section 3.5.3 for analytical expressions). One observes from the plot that, in the quasi-dispersive regime, the increasing virtual interaction with the second excited resonator state acts against the interaction with the second excited qubit states. Notice that all contributions to $\zeta^{(4)}$ are virtual interactions of the second excited state, i.e., $\zeta^{(4)} = \zeta_\mathrm{t} + \zeta_\mathrm{r}$, as illustrated in Figure 3.5b.

Although the λ^4-perturbation gives insight into the different contributions to the energy shift, perturbation beyond the order λ^4 also has a non-negligible contribution in the quasi-dispersive regime. Since RSWT requires considerably fewer commutators, we are able to compute the λ^6-perturbation, with only two iterations and 7 commutators (see Table 3.1). The λ^6-perturbation captures the location of the minimum more accurately, but still shows a false minimum close to the resonant regime, as shown in Figure 3.4b.

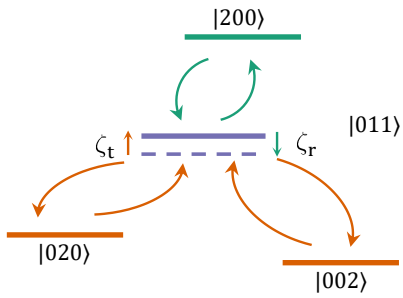
Apart from perturbation, we also apply NPAD to compute the interaction strength. We first define 4 Givens rotations with respect to the direct qubit-resonator coupling terms from the original Hamiltonian. The rotations are then applied sequentially to obtain the first effective Hamiltonian. Next, we apply another 4 rotations targeted at the two-photon couplings, such as the effective qubit-qubit coupling. The indices of those 8 rotations are listed in the first two columns of Table 3.2. These two steps are equivalent to the two iterations in RSWT. However, the recursive Givens rotations replace the BCH expansion, resulting in a much simpler calculation. Illustrated in Figure 3.4b, the approximation with those 8 rotations is as good as the λ^6-perturbation, but without the false minimum. Both capture the zero points very well compared to the numerical diagonalization, where the 4 lowest levels are included for each qubit and the resonator.

With those calculations, we can then investigate the effect of the high-order corrections. We find that, for instance, g_q shifts the zero point to the regime of smaller frequency detuning, corresponding to shrinking the half-circle in the numerical calculation in Figure 3.4a. In addition, for stronger coupling strength, the dip becomes narrower, which indicates a trade-off between the interaction strength and feasibility of qubit fabrication [121]. A detailed description of the effect of higher-order perturbation in the quasi-dispersive regime is presented in Section 3.5.4.

Overall, our investigation reveals different contributions to the ZZ interaction and provides tools to study the energy landscape in this quasi-dispersive regime. Because of the comparably smaller detuning, operations on this regime provide stronger interactions for entangling gates, and hence may achieve a better quantum speed limit for universal gate sets, i.e. without sacrificing local gates [83].



(a)



(b)

Figure 3.5: **(a)**: Different contributions to the ZZ interaction in the quasi-dispersive regime. The symbols ζ_t and ζ_r represent the contribution of virtual interaction with the second excited qubit (t) and resonator (r) states in the λ^4-perturbation. The former is the typical cause of ZZ cross-talk in the strong dispersive regime, while the latter is used to counteract the energy shift. The notation $\zeta^{(4)}$ refers to the λ^4-perturbation [equation (3.37)] that goes pass zero in the quasi-dispersive regime. In addition, ζ_{disp} denotes the strong dispersive approximation [equation (3.36)], which also underestimates the ZZ interaction induced by the non-qubit states. Parameters used are the same as in Figure 3.4. **(b)**: Illustration of the two contributions to the ZZ interaction strength in the quasi-dispersive regime. The solid lines and the curved arrows represent the bare states and the interaction among them. The second excited resonator and transmon states push the qubit $|11\rangle$ state into different directions.

static $\hat{H}$		driving $\hat{H}_{\mathrm{d}}$
Step 1	Step2	
010-100	011-200	00-10
001-100	001-010	01-11
011-101	011-002	10-20
011-110	011-020	11-21

Table 3.2: Leading coupling terms in (block-) diagonalizing the static and the driving Hamiltonians of the cross-resonance gate, upon which the Jacobi iteration is constructed. For the static Hamiltonian (Section 3.3.2), the three numbers refer to the state of the resonator, qubit 1 and qubit 2, respectively. E.g. 010-001 denotes the effective coupling between the two qubits. For the driving Hamiltonian (Section 3.3.3), we use the effective qubit-qubit model. Hence only the qubit states are listed.

3.3.3 The cross-resonance coupling strength

Following the previous examples, we here study superconducting qubits under an external cross-resonance drive. The cross-resonance interaction is activated by driving the control qubit with the frequency of the target qubit, which has been studied intensively and demonstrated in various experiments [106, 122–126]. In the two-qubit subspace, the dominant Hamiltonian term is written as a Pauli matrix ZX, which generates a CNOT gate up to single-qubit corrections. Therefore, ideally, only the population of the target qubit will change after the gate operation. The effective model is usually derived by block diagonalizing the non-qubit leakage levels as well as the population flip of the control qubit [80, 81, 89]. The coupling strength is then characterized by the coefficient of the ZX Hamiltonian term.

The analytical block diagonalization of the Hamiltonian is only possible when neglecting all the non-qubit levels. Hence, perturbative expansion is often used, where the small parameter is defined as Ω/Δ_-, i.e., the ratio between the drive amplitude and the qubit-qubit detuning. However, to achieve fast gates, the qubit-qubit detuning is often designed to be small, ranging from 50 MHz to 200 MHz. Therefore, the perturbative diagonalization only works well for a weak drive.

In the rest of this subsection, we show that with only 4 two-by-two Givens rotations on the single-photon couplings, we can block-diagonalize the drive term and obtain an estimation of the coupling strength as good as the numerical result and far above the perturbative regime.

We start from the static Hamiltonian $\hat{H}$ in equation (3.24) and define a driving Hamiltonian in the rotating frame

$$\hat{H}_{\mathrm{d}} = \frac{\Omega}{2}(\hat{b}_1 + \hat{b}_1^\dagger). \tag{3.39}$$

The full Hamiltonian is then written as $\hat{H} + \hat{H}_{\mathrm{d}} - \hat{H}_{\mathrm{R}}$ where $\hat{H}_{\mathrm{R}} = \omega_d(\hat{b}_1^\dagger\hat{b}_1 + \hat{b}_2^\dagger\hat{b}_2)$ with ω_d the driving frequency [80]. To compute the interaction strength, both the

qubit-qubit effective interaction g and the drive on the control qubit Ω need to be diagonalized. In particular, the second one can be as large as the energy gap and dominant in the unwanted couplings [81]. For simplicity, we assume g is small and diagonalize it with a leading-order perturbation, discarding all terms smaller than $\mathcal{O}(g^2)$. In this frame, one obtains a ZX interaction that increases linearly with the drive strength [80]. This is equivalent to moving to the eigenbases of the idling qubits and allows us to focus on applying NPAD to the drive $\hat{H}_\mathrm{d}$. The same method used in Section 3.3.2 can be applied here to improve this approximation.

Targeting the dominant drive terms listed in the right column of Table 3.2, we construct 4 Givens rotations. The rotations are constructed with respect to the same Hamiltonians and then applied iteratively as separate unitaries. The obtained ZX interaction strength reads

$$\begin{aligned}\omega_{ZX} =& g\Omega\left(\frac{s_1^2c_2^2-c_1^2}{2\Delta_-}-\frac{s_2^2}{(\Delta_-+\alpha_1)}\right.\\ &\left.+\frac{(s_1^2-c_1^2c_2^2)(\alpha_1-\Delta_-)-\sqrt{2}\alpha_1 s_1 s_2 c_2}{2\Delta_-(\Delta_-+\alpha_1)}\right)\end{aligned} \tag{3.40}$$

with $c_j=\cos(\theta_j/2)$, $s_j=\sin(\theta_j/2)$ and $\Delta_-=\omega_1-\omega_2$. The rotation angles are defined by the drive strength $\theta_1=\arctan\left(\frac{\Omega}{\Delta_-}\right)$ and $\theta_2=\arctan\left(\frac{\sqrt{2}\Omega}{2\Delta_-+\alpha_1}\right)$.

This analytical coupling strength is plotted in Figure 3.6, compared with the perturbative expansions in Ref. [80] and numerical block-diagonalization. The result matches well with the numerical calculation, even when the ratio $\frac{\Omega}{\Delta_-}$ is approaching one. On the contrary, the perturbative expansion shows a large deviation as the driving power increases. The numerical block-diagonalization is implemented using the least action method [80, 102, 115]. To our surprise, although no least action condition is imposed on the Jacobi iteration, the method automatically follows this track and avoids unnecessary rotations. This suggests that the Jacobi iteration chooses an efficient path of block-diagonalization.

Notice that in the above example, no rotations are performed for levels beyond the second excited state because they are not directly coupled to the qubit subspace. In other parameter regimes, more coupling terms may become significant and need to be added to the diagonalization. For instance, the two-photon interaction between $|0\rangle$ and $|2\rangle$ of the control qubit will be dominant in the regime where $\Delta_-\approx-\alpha_2/2$ [81]. The fact that high precision can be achieved with only rotations on the single-photon couplings in this example also indicates that the dominant error of perturbation lies in the BCH expansion used in diagonalizing the strong single-photon couplings, rather than in higher levels or high-order interactions.

3.4 Conclusion and outlook

We introduced the symbolic algorithm NPAD, based on the Jacobi iteration, for computing closed-form, parametric expressions of effective Hamiltonians. The

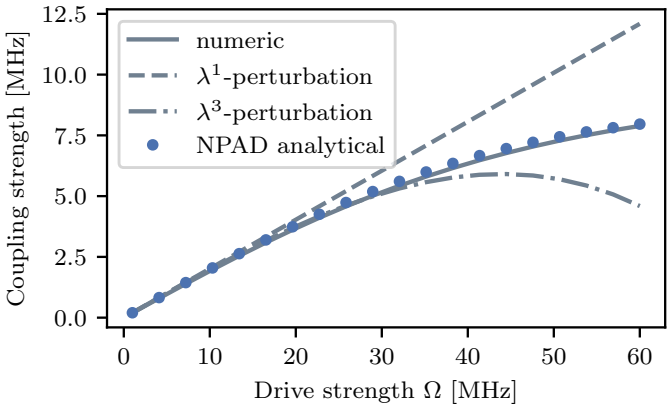


Figure 3.6: Cross-resonance coupling strength as a function of the drive strength. The analytical coupling strength is computed with 4 two-by-two Givens rotations on the single-photon coupling terms [equation (3.40)] and compared to perturbative expansion and the numerical calculation. The parameters used are inspired by the device in Ref. [106], with the qubit-qubit detuning approximately 60 MHz, the effective qubit-qubit coupling −3 MHz and the anharmonicity -0.3 GHz for both qubits.

method applies rotation unitaries iteratively onto a Hamiltonian, with each rotation recursively defined upon the previous result and removing a chosen coupling between two states. Compared to perturbation, it uses two-by-two rotations to avoid the exponentially increasing commutators in the BCH expansion. In the perturbative limit, the method reduces to a modified form of the Schrieffer-Wolff transformation, RSWT, that inherits the recursive structure of the Jacobi iteration. The recursive structure avoids unnecessary expansion and results in an exponential reduction in the number of commutators compared to the traditional perturbative expansion. The two methods can also be combined as a hybrid method, where NPAD is used to remove strong couplings while RSWT is applied afterwards to effectively eliminate the remaining weak coupling.

Applying these methods to superconducting qubit systems, we showed that high precision estimation can be achieved beyond the perturbation regime, either as explicit short analytical expressions, or closed-form parametric expressions for computer-aided calculation. Although in the study we used the Kerr model, more detailed models such as in Ref. [81] can also be incorporated with little additional effort.

Despite the fact that using the Jacobi iteration for machine-precision diagonalization is less efficient than other methods such as QR diagonalization, the iteration can be truncated for symbolic approximation. For many questions in quantum engineering, the largest part of the energy structure and dominant couplings are

known in advance. Therefore, the iterative method can be designed for removing dominant couplings and decoupling a subspace from non-relevant Hilbert spaces, which is often used in modelling dynamics in large quantum systems [91, 127]. The result is, however, always a closed-form, parametric expression, which, though usually harder for humans to read, shows its own advantage in computer-aided calculations.

We expect our method to have significant application in quantum technologies, where the elimination of auxiliary or unwanted spaces (e.g. for block-diagonalization) needs to be done to significant precision to enable practically useful models. In particular, relevant applications include experiment and architecture design, reservoir engineering, cross-talk suppression, few- and many-body interaction engineering, effective qubit models, and more generally improved approximations where Schrieffer-Wolff methods are typically used. We also expect that the methods presented here will find extensions for simplifying other equations of motion, such as in open-quantum systems [128, 129], non-linear systems [130], or for uncertainty propagation [131]. Last but not least, accurate, parametric diagonalization should be especially useful for time-dependent diagonalization where adiabatic following can be enforced by DRAG [64, 132] or other counter-diabatic [49, 133] approaches.

3.5 Appendix

3.5.1 The error bound for truncating the BCH expansion

In the main text, we presented equation (3.20) as the expression to compute the transformed matrix $\hat{H}'$, which is a function of the off-diagonal part of the original matrix $\hat{V}$ and the generator $\hat{S}$. The expression is derived from a truncated BCH formula. In the following, we derive the error bound of the truncation.

Without truncation, equation (3.20) is written as

$$\hat{H}'_{\text{ideal}} = \hat{D} + \sum_{t=1}^{\infty} \frac{t}{(t+1)!} \mathcal{C}_t(\hat{S}, \hat{V}) \tag{3.41}$$

where we neglected the index n for the iteration step. If the expansion is truncated at $t = m - 1$, one obtains

$$\begin{aligned} \epsilon &= \left\| \hat{H}'_{\text{ideal}} - \hat{H}'_{\text{trunc}} \right\| = \left\| \sum_{t=m}^{\infty} \frac{t}{(t+1)!} \mathcal{C}_t(\hat{S}, \hat{V}) \right\| \\ &\leq \sum_{t=m}^{\infty} \frac{t 2^t}{(t+1)!} \|\hat{S}\|^t \|\hat{V}\| \leq \frac{2^m}{m!} \frac{\|\hat{S}\|^m}{1 - \|\hat{S}\|} \|\hat{V}\| \end{aligned} \tag{3.42}$$

where we assume in the last inequality that $\|\hat{S}\| < 1/2$.

3.5.2 Efficiency comparison between RSWT and SWT

We show here that, given a finite-dimensional Hamiltonian $\hat{H}$, RSWT is more efficient than SWT for perturbation beyond level λ^2 with an exponential decrease in the number of commutators. We measure the complexity by the number of commutators that need to be evaluated to compute all eigenenergy corrections up to λ^K, denoted by $\mathcal{N}$. The general formula is presented below while the numbers for $K \leq 8$ are given in Table 3.1 in the main text.

For SWT, one can find the general expression as well as explicit formulas up to λ^5 in Ref. [80]. The number of iterations required to reach order λ^K is $K-1$. In addition, at each iteration n, one needs to include also mixed terms composed of generator $\hat{S}_l$ with $l \leq n$. The number $\mathcal{N}$ is given by

$$\mathcal{N}_{\mathrm{SWT}} = \sum_{n=1}^{K-1} \sum_{l=1}^{n} 2^{l-1} = 2^K - K - 1 \tag{3.43}$$

where 2^{l-1} is the number of distinct tuples $(\hat{S}_{i_1}, \hat{S}_{i_2}, \hat{S}_{i_3}, \cdots)$ with $\sum_j i_j = l$. We have taken into consideration that $[\hat{S}_1, \mathrm{diag}(\hat{\mathrm{H}})] = -\hat{\mathrm{V}}$ and $[\hat{S}_{n+1}, \mathrm{diag}(\hat{\mathrm{H}})]$ is known by the construction of $\hat{S}_{n+1}$.

For RSWT, the calculation of commutators in each iteration is given in equation (3.20). Because $\mathcal{C}_{t+1}(\hat{A}, \hat{B})$ can be calculated from $\mathcal{C}_{t+1}(\hat{A}, \hat{B})$ with only one additional commutators, the number commutators to be evaluated in equation (3.20) is exactly $m - 1 = \lfloor \frac{K}{2^n} \rfloor - 1$. The total number of iteration $n_{\max}$ is given by $\lfloor \log_2(K) \rfloor$. Therefore, we obtain

$$\mathcal{N}_{\mathrm{RSWT}} = \sum_{n=0}^{\lfloor \log_2(K) \rfloor - 1} \lfloor \frac{K}{2^n} \rfloor - 1 < 2K. \tag{3.44}$$

The reduction compared to SWT comes from the fact that the energy difference in $\hat{H}_n$ is used in the definition of $\hat{S}_{n+1}$, rather than the bare energy difference in $\hat{H}$. The recursive expressions avoid unnecessary expansions. One obtains the same final expressions as from SWT up to λ^K, if one expands the energy difference into a polynomial series

$$\frac{1}{\Delta E_{\mathrm{bare}} + \Delta E_{\mathrm{correction}}} = \frac{1}{\Delta E_{\mathrm{bare}}} \mathrm{poly}\left(\frac{\Delta \mathrm{E}_{\mathrm{correction}}}{\Delta \mathrm{E}_{\mathrm{bare}}} \right) \tag{3.45}$$

and substitutes in expressions so that it depends only on the bare energy and couplings.

3.5.3 RSWT results for the ZZ interaction strength

Using RSWT described in Section 3.2.2, we compute the effective Hamiltonian up to λ^6, where λ is defined as the ratio between the largest coupling and energy gap.

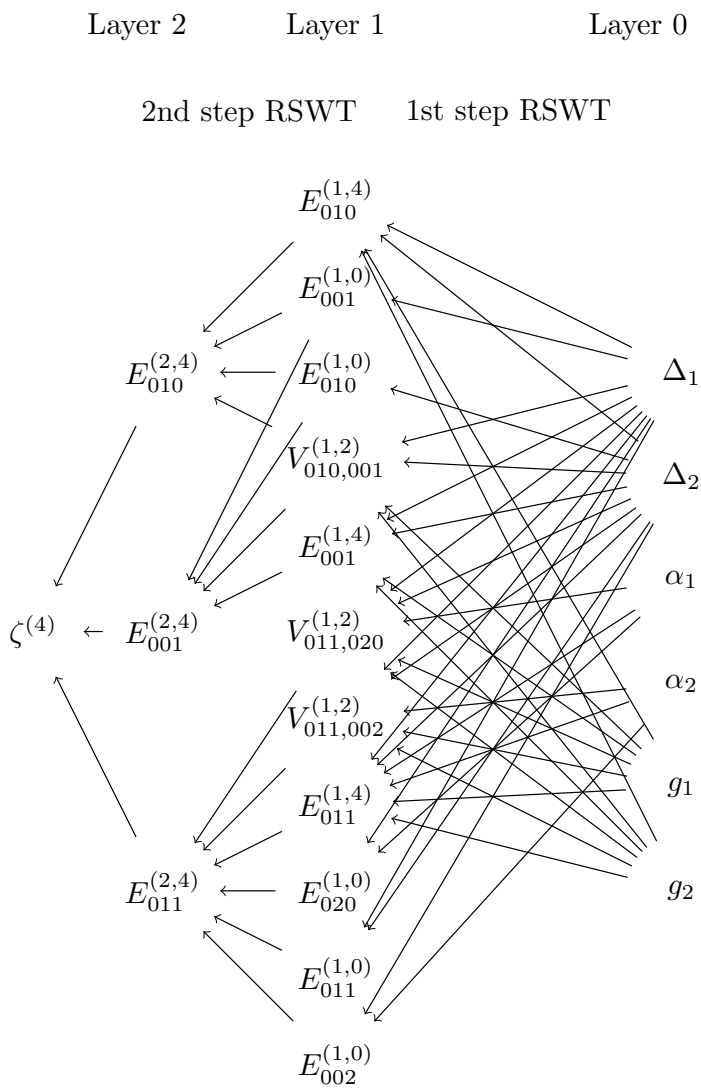


Figure 3.7: The network illustration of the closed-form expression of $\zeta^{(4)}$ parameterized by Δ_1, Δ_2, α_1, α_2, g_1, and g_2, obtained from the two-step RSWT. Each node in the 1st and 2nd layers is a matrix entry in the Hamiltonian $\hat{H}_1$ and $\hat{H}_2$. A node in layer $n+1$ is expressed as a function of the nodes in layer n, represented by an edge. In particular, symbols $E^{(n,k)}_{lpq}$ represent the λ^k diagonal entries of $\langle lpq|\,\hat{H}_n\,|lpq\rangle$ and $V^{(n,k)}_{lpq,l'p'q'}$ the effective coupling. The upper index k denotes the level of perturbation, e.g., $k=4$ means that it is a λ^4-perturbative correction.

To compute the λ^4- and λ^6-perturbation, RWST only takes 2 iterations with 4 and 7 commutators respectively, which is significantly smaller than those required for SWT as shown in Table 3.1. A third iteration only adds an improvement of $\mathcal{O}(\lambda^8)$ to the eigenenergy because the off-diagonal terms of $\hat{H}_2$ are at most $\mathcal{O}(\lambda^4)$.

Because of the recursive structure of RSWT, each matrix element in $\hat{H}_{n+1}$ is given as a function of matrix elements in $\hat{H}_n$. Hence, the final result is a closed-form expression parameterized by the matrix elements of the original Hamiltonian $\hat{H}$, i.e. the hardware parameters. The parametric expression consists only of algebraic expressions and the dependence can be illustrated as a network. For instance, we show the network representation of the λ^4-perturbation $\zeta^{(4)}$ in Figure 3.7. Each symbol in layer $n+1$ is analytically expressed as a function of symbols in layer n, represented by arrows. The arrows between the first and the second layer represent the definition $\zeta^{(4)} = E^{(4)}_{011} - E^{(4)}_{001} - E^{(4)}_{010}$. Given all the six hardware parameters (layer 0), one can evaluate $\zeta^{(4)}$ by recursively evaluating all the nodes it depends on.

In the following, we present the analysis of λ^4- and λ^6-perturbation.

λ^4-perturbation

The λ^4-perturbative correction for ζ is given as

$$\zeta^{(4)} = E^{(2,4)}_{011} - E^{(2,4)}_{010} - E^{(2,4)}_{001}. \tag{3.46}$$

The notation $E^{(n,k)}_{lpq}$ represents the λ^k-perturbation obtained from $\hat{H}_n$. The sub-indices lpq denotes the resonator state $|l\rangle$ and two qubit states $|p\rangle$, $|q\rangle$.

We first calculate $E^{(2,4)}_{011}$. Substituting the expression for $\hat{H}_2$ as a function of entries in $\hat{H}_1$, we obtain

$$\begin{aligned} E^{(2,4)}_{011} =& \frac{V^{(1,2)}_{002,011} V^{(1,2)}_{011,002}}{E^{(1,0)}_{011} - E^{(1,0)}_{002}} + \frac{V^{(1,2)}_{011,020} V^{(1,2)}_{020,011}}{E^{(1,0)}_{011} - E^{(1,0)}_{020}} \\ &+ \frac{V^{(1,2)}_{011,200} V^{(1,2)}_{200,011}}{E^{(1,0)}_{011} - E^{(1,0)}_{200}} + E^{(1,4)}_{011} \end{aligned} \tag{3.47}$$

where $V^{(n,k)}_{lpq,l'p'q'}$ denotes the interaction between state $|lpq\rangle$ and $|l'p'q'\rangle$.

The physical meaning of each term in equation (3.47) can be interpreted as follows: The first two terms are identical to the dispersive approximation given in equation (3.36), which is the consequence of the effective qubit-qubit interaction. The third term, depending on the effective interaction between $|200\rangle$ and $|011\rangle$, is 0 at this order. This is because the destructive interference between the path $|011\rangle \to |110\rangle \to |200\rangle$ and $|011\rangle \to |101\rangle \to |200\rangle$ results in $V^{(1,2)}_{011,200} = V^{(1,2)}_{200,011} = 0$. The last term, $E^{(1,4)}_{011}$, is what the approximation of a strong dispersive regime fails to characterize. It was generated by the commutator $[\hat{S}_1, [\hat{S}_1, [\hat{S}_1, \hat{V}_0]]]$ and the energy gaps in the denominator of entries in $\hat{S}_1$ are always the qubit-resonator

detuning (plus the anharmonicity), which, in the strong dispersive regime, is much larger than the qubit-qubit detuning in equation (3.36). Hence the last term is much smaller in the strong dispersive regime. However, in the quasi-dispersive regime, it plays a key role in suppressing the ZZ interaction as shown in Figure 3.5b.

After including the single-excitation terms $E_{010}^{(2,4)}$ and $E_{001}^{(2,4)}$ using the same two-step RSWT, we separate the contributions of virtual interaction into 2 categories: those including the second excited qubit state (denoted by t) and those including the second excited resonator state (denoted by r):

$$\begin{aligned}\zeta_{\mathrm{t}}^{(4)} =&\zeta_{\mathrm{disp}} - \frac{g_1^2 g_2^2}{2\Delta_2\left(\Delta_1+\alpha_1\right)^2} - \frac{3g_1^2 g_2^2}{2\Delta_2^2\left(\Delta_1+\alpha_1\right)} \\ &- \frac{g_1^2 g_2^2}{2\Delta_1\left(\Delta_2+\alpha_2\right)^2} - \frac{3g_1^2 g_2^2}{2\Delta_1^2\left(\Delta_2+\alpha_2\right)}\end{aligned} \tag{3.48}$$

$$\zeta_{\mathrm{r}}^{(4)} = \frac{2g_1^2 g_2^2}{\Delta_1\Delta_2^2} + \frac{2g_1^2 g_2^2}{\Delta_1^2\Delta_2} \tag{3.49}$$

where ζ_{disp} is given by equation (3.36). Summing all the contributions gives the λ^4-perturbation $\zeta^{(4)}$ in equation (3.37). Notice that virtual interactions that only involve the first excited state have no contribution to the ZZ interaction at this perturbation level, i.e., $\zeta^{(4)} = \zeta_{\mathrm{t}} + \zeta_{\mathrm{t}}$. This is because the energy shift of $|011\rangle$ induced by $|101\rangle$ and $|110\rangle$ cancels that of $|010\rangle$ and $|001\rangle$ induced by $|100\rangle$.

λ^6-perturbation

Using the two-step RSWT, we also computed the λ^6-perturbative correction to the ZZ interaction strength:

$$\zeta^{(6)} = \zeta_{\mathrm{disp}}^{(6)} + \zeta_{\mathrm{rest}}^{(6)} \tag{3.50}$$

The first contribution corresponds to the effective qubit-qubit interaction and dominants in the strong dispersive regime. It turns out that it only includes the next order of effective interaction and energy difference. Hence, for simplicity, we present it together with $\zeta_{\mathrm{disp}}^{(4)}$:

$$\begin{aligned}\zeta_{\mathrm{disp}}^{(4)} + \zeta_{\mathrm{disp}}^{(6)} =&\frac{\left(V_{011,020}^{(1,2)} + V_{011,020}^{(1,4)}\right)\left(V_{020,011}^{(1,2)} + V_{020,011}^{(1,4)}\right)}{\Delta E_{011,020}^{(2,0)} + \Delta E_{011,020}^{(2,2)}} \\ &+\frac{\left(V_{002,011}^{(1,2)} + V_{002,011}^{(1,4)}\right)\left(V_{011,002}^{(1,2)} + V_{011,002}^{(1,4)}\right)}{\Delta E_{011,002}^{(2,0)} + \Delta E_{011,002}^{(2,2)}}\end{aligned} \tag{3.51}$$

with terms regarding to the virtual interactions between states $|011\rangle$ and $|020\rangle$

given by

$$V^{(1,2)}_{011,020} = V^{(1,2)}_{020,011} = \frac{\sqrt{2}g_1 g_2}{2\left(\alpha_1+\Delta_1\right)} + \frac{\sqrt{2}g_1 g_2}{2\Delta_2}, \tag{3.52}$$

$$\begin{aligned} V^{(1,4)}_{011,002} = V^{(1,4)}_{002,011} =& -\frac{\sqrt{2}g_1 g_2^3}{4\left(\alpha_2+\Delta_2\right)^3} + \frac{\sqrt{2}g_1 g_2^3}{8\Delta_2^2\left(\alpha_2+\Delta_2\right)} - \frac{7\sqrt{2}g_1 g_2^3}{4\Delta_1\left(\alpha_2+\Delta_2\right)^2} \\ &+ \frac{3\sqrt{2}g_1 g_2^3}{2\Delta_1\Delta_2\left(\alpha_2+\Delta_2\right)} - \frac{5\sqrt{2}g_1 g_2^3}{8\Delta_1\Delta_2^2} - \frac{7\sqrt{2}g_1^3 g_2}{8\Delta_1^2\left(\alpha_2+\Delta_2\right)} \\ &- \frac{\sqrt{2}g_1^3 g_2}{8\Delta_1^3}, \end{aligned} \tag{3.53}$$

$$\Delta E^{(2,0)}_{011,020} + \Delta E^{(2,2)}_{011,020} = -\alpha_1 - \Delta_1 + \Delta_2 - \frac{2g_1^2}{\alpha_1+\Delta_1} + \frac{g_2^2}{\Delta_2} + \frac{g_1^2}{\Delta_1}. \tag{3.54}$$

Terms corresponding to states $|011\rangle$ and $|002\rangle$ are obtained by interchanging the sub-index 1 and 2 in each expression above.

The rest of the contribution can be summed up as

$$\zeta^{(6)}_{\mathrm{rest}} = \zeta^{(6)}_{\mathrm{rest},g_1^2 g_2^4} + \zeta^{(6)}_{\mathrm{rest},g_1^4 g_2^2} \tag{3.55}$$

with

$$\begin{aligned} \zeta^{(6)}_{\mathrm{rest},g_1^2 g_2^4} =& \frac{9g_1^2 g_2^4}{4\Delta_2^3\left(\Delta_1+\alpha_1\right)^2} + \frac{23g_1^2 g_2^4}{4\Delta_2^4\left(\Delta_1+\alpha_1\right)} \\ &+ \frac{g_1^2 g_2^4}{2\Delta_1\left(\Delta_2+\alpha_2\right)^4} - \frac{g_1^2 g_2^4}{4\Delta_1\Delta_2^2\left(\Delta_2+\alpha_2\right)^2} - \frac{4g_1^2 g_2^4}{\Delta_1^3\Delta_2\left(\Delta_2+\alpha_2\right)} \\ &+ \frac{7g_1^2 g_2^4}{2\Delta_1^2\left(\Delta_2+\alpha_2\right)^3} - \frac{5g_1^2 g_2^4}{2\Delta_1^2\Delta_2\left(\Delta_2+\alpha_2\right)^2} + \frac{3g_1^2 g_2^4}{4\Delta_1^2\Delta_2^2\left(\Delta_2+\alpha_2\right)} \\ &+ \frac{4g_1^2 g_2^4}{\Delta_1^2\Delta_2^3} + \frac{4g_1^2 g_2^4}{\Delta_1^3\left(\Delta_2+\alpha_2\right)^2} - \frac{6g_1^2 g_2^4}{\Delta_1\Delta_2^4}. \end{aligned} \tag{3.56}$$

The second contribution, $\zeta^{(6)}_{\mathrm{rest},g_1^4 g_2^2}$, is obtained again by interchanging the sub-index 1 and 2.

3.5.4 Effect of higher-order perturbation on the zero points of ZZ interaction

The λ^4-perturbation described by equation (3.37) predicts the zero points as a circle with a radius of $2|\alpha|$, independent of g. However, as they are located in the quasi-dispersive regime for systems with weak anharmonicity, the higher-order perturbation is not always negligible. Here, we qualitatively describe how the higher-order (> 4) affects the zero points of ζ.

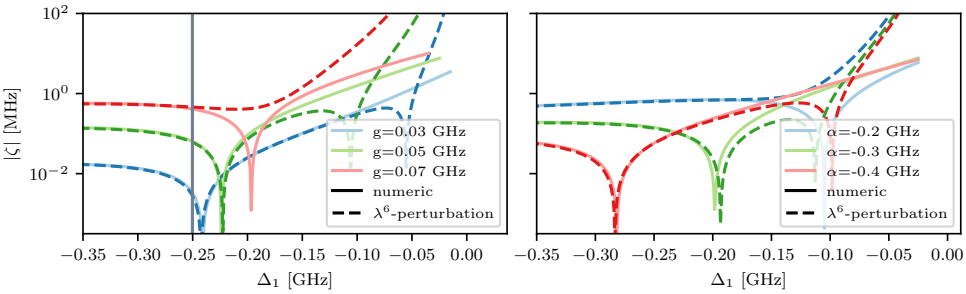


Figure 3.8: The dependency of ζ on the resonator-qubit interaction strength g and the qubit anharmonicity α. Computed with RSWT to the λ^6-perturbation. **Left:** Dependency on g. The vertical line denotes the zero point predicted by the 4th-order perturbation, which is independent on g. Both the numerical result and the λ^6-perturbation indicate that the zero points are shifted to the regime with smaller qubit-resonator detuning. **Right:** Dependency on α. The default parameters used, if not specified in the plots, are $\Delta_- = 0.4|\alpha|$, $g = 50$ MHz, $\alpha_1 = \alpha_2 = \alpha = -330$ MHz.

We observe that, in contrast to the λ^4-perturbation, when including the higher orders, the zero points depend on the coupling strength g. As shown in Figure 3.8, the higher-order perturbation shifts the zero points to the regime of smaller detuning. The larger the coupling, the stronger the shift is.

One can estimate the accuracy of perturbation around the zero points by the ratio $g/|\alpha|$. At the zero points of ζ, the larger the anharmonicity and the smaller the coupling, the better the perturbative approximation. This is because the perturbation is characterized by the small parameter $\lambda = g/\Delta$ and near the zero-points Δ depends linearly on α [see equation (3.37)], hence the ratio $g/|\alpha|$. This is also illustrated in Figure 3.8, where we compare the deviation between the numerical result and the perturbation. The minimum even vanishes in the analytical result when it is close to the resonant lines. This behaviour also indicates that for superconducting qubits with a relatively large anharmonicity, the ZZ interaction can also be completely suppressed in the strong dispersive regime in this qubit-resonator-qubit model.

4

Revisiting the single-Transmon DRAG pulse

4.1 Introduction to DRAG

A Transmon qubit is typically modelled as a non-linear oscillator, with its lowest two energy levels serving as the qubit states [5]. Ideally, population to higher energy levels should be prevented. However, the anhamonicity, i.e., the energy difference between states $|0\rangle$, $|1\rangle$ and states $|1\rangle$, $|2\rangle$ is often kept weak in practice to mitigate other error sources such as charge noise [41]. With the small non-linearity, state $|2\rangle$ may get populated during qubit operation, which can reach 50 MHz for a 20 ns single-qubit gate, while the anharmonicity ranges from 200 to 300 MHz. To address leakage to non-computational states, the Derivative Removal by Adiabatic Gate (DRAG) technique was introduced by Motzoi et al. in 2009 [64]. Nowadays, DRAG finds widespread application in single-qubit gate operations in superconducting qubits. In this chapter, we revisit the derivation of DRAG, from a pedagogical perspective and highlight details that are potentially overlooked or omitted in the literature over the past decade.

In this chapter, we adopt the simplest model that captures the essence of the problem, a three-level system. In the rotating frame, the first two levels are degenerate, with a small free parameter, δ, reserved for fine-tuning the phase. The second excited level, $|2\rangle$, is separated from the others by the anharmonicity Δ. The microwave drive couples both $|0\rangle$, $|1\rangle$ and $|1\rangle$, $|2\rangle$, with the ratio between them denoted by λ. Under the above definition, we write the rotating frame Hamiltonian as

$$\hat{H}_{0,0} = \begin{pmatrix} 0 & \frac{\epsilon\Omega}{2} & 0 \\ \frac{\epsilon\Omega}{2} & \delta\epsilon^2 & \frac{\epsilon\lambda\Omega}{2} \\ 0 & \frac{\epsilon\lambda\Omega}{2} & \Delta + 2\delta\epsilon^2 \end{pmatrix}, \tag{4.1}$$

where ϵ is used to denote the perturbation order. We assume the detuning is a

second-order correction, which will be verified later in this chapter.

The overall goal of the DRAG method is to determine an appropriate effective frame and the corresponding control function $\Omega(t)$. They are constructed such that the dynamics are confined in the qubit subspace, and the effective coupling is real and implements a predefined rotation angle θ. One can consider this frame as an adiabatic frame where any leakage out of the qubit subspace is adiabatically eliminated. It is crucial to note that this holds true only in the effective frame. In the lab frame, the non-computational levels are slightly populated. It is only through the DRAG pulse design that the population is guaranteed to go back to the qubit subspace at the end of the operation.

Various approaches exist for deriving the DRAG pulses, all revolving around a common framework of time-dependent frame transformation. Typically, this transformation is realized through an operator $\hat{V} = \exp(\hat{\mathrm{S}})$, Since $\hat{V}$ is time-dependent, the effective Hamiltonian also has a derivative term,

$$\hat{H}_{\text{eff}} = \hat{V}\hat{H}\hat{V}^{\dagger} + i\dot{\hat{V}}\hat{V}^{\dagger}. \tag{4.2}$$

Typically, if the error coupling is small, $i\dot{\hat{V}}\hat{V}^{\dagger}$ usually has a smaller amplitude than the original coupling. If $\hat{S}(t_1)$ and $\hat{S}(t_2)$ commute for all t_1 and t_2, we also have $\dot{\hat{V}}\hat{V}^{\dagger} = \dot{\hat{S}}$. Often, the coupling is smaller than the anharmonicity, in which case the perturbative form is given by

$$\hat{H}_{\text{eff}} = i\dot{\hat{S}} + \hat{H} + [\hat{S}, \hat{H}] + \frac{1}{2}[\hat{S}, [\hat{S}, \hat{H}]] + \cdots, \tag{4.3}$$

where the expansion is referred to as the BCH expansion.

In this thesis, we explore two different strategies to construct DRAG solutions. The first approach aligns with the strategy of the counterdiabatic drive method [48–50, 133], which is discussed in the review Ref. [132]. Here, the generator $\hat{S}$ is constructed such that $\hat{V}\hat{H}\hat{V}^{\dagger}$ has no coupling between the qubit subspace and the ancillary levels. The DRAG correction is then chosen to cancel the remaining coupling in the derivative terms $i\dot{\hat{V}}\hat{V}^{\dagger}$. Different choices of $\hat{V}$ and priorities in removing the leakage terms result in distinct DRAG corrections. One example will be the block diagonalization introduced in Chapter 3. In cases where multiple error couplings are present, this leads to a system of equations. However, if all the leakage couplings are proportional to the drive Ω, this simplifies to a linear system of equations, which can be readily solved [67, 132]

It is important to point out that, in this approach, the drive Hamiltonian must be also transformed into this effective frame when defining the corrections. This makes the system of equations hard to solve in general since adding a correction pulse to suppress one transition may add more error to another. In particular, for high-order multiphoton processes, we develop a second formulation, a recursive formulation, that will be briefly mentioned in Section 4.7 and extensively investigated in Chapter 5 and Chapter 6.

4.2 Notations

Throughout this chapter, we use two indices to denote different effective Hamiltonians and generators. With the symbol $\hat{H}_{k,l}$, the first index k denotes Hamiltonians with different time evolution, e.g., the original Hamiltonian or Hamiltonians with different correction terms. If two Hamiltonians have the same k, it means that they are equivalent Hamiltonians (up to the truncated expansion terms), i.e., they have the same time-evolution $\hat{U} = \hat{\mathcal{T}} \exp\left[-\int i\hat{H}_k(t)\mathrm{d}t\right]$. The second index l denotes the effective frame, e.g. $l = 0$ represents the lab frame. Hamiltonians with the same k but different l represent the equivalent Hamiltonians in different effective frames. For a Schrieffer Wolff generator $\hat{S}$, we use the subscript $\hat{S}_{l_1 \to l_2}$ to denote the frame transformation from frame l_1 to frame l_2. In this chapter, we discard all the terms smaller or equal to ϵ^3, except for the amplitude correction in Section 4.7.

4.3 First-order DRAG correction

In this section, we derive the first-order DRAG proposed in the original proposal [64]. We deviate slightly from the approach outlined in Ref. [64] to offer more physical motivations for the transformations chosen. The results are equivalent in the first two orders.

To begin with, for Transmon Hamiltonian, the leakage term is the coupling between $|1\rangle$ and $|2\rangle$, hence we start from the transformation

$$\hat{S}_{0\to1} = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & -\frac{\epsilon\lambda\Omega}{2\Delta} \\ 0 & \frac{\epsilon\lambda\Omega}{2\Delta} & 0 \end{pmatrix}. \tag{4.4}$$

This generator is designed such that the first-order coupling between $|1\rangle$ and $|2\rangle$ will be completely diagonalized. The resulting effective Hamiltonian reads

$$\hat{H}_{0,1} = \hat{H}_{0,0} + \left[\hat{S}_{0\to1}, \hat{H}_{0,0}\right] + \frac{1}{2}\left[\hat{S}_{0\to1}, \left[\hat{S}_{0\to1}, \hat{H}_{0,0}\right]\right] + i\dot{\hat{S}}_{0\to1} + \mathcal{O}(\epsilon^3) \tag{4.5}$$

$$= \begin{pmatrix} 0 & \frac{\epsilon\Omega}{2} & \frac{\epsilon^2\lambda\Omega^2}{4\Delta} \\ \frac{\epsilon\Omega}{2} & \delta\epsilon^2 - \frac{\epsilon^2\lambda^2\Omega^2}{4\Delta} & -\frac{i\epsilon\lambda\dot{\Omega}}{2\Delta} \\ \frac{\epsilon^2\lambda\Omega^2}{4\Delta} & \frac{i\epsilon\lambda\dot{\Omega}}{2\Delta} & \Delta + 2\delta\epsilon^2 + \frac{\epsilon^2\lambda^2\Omega^2}{4\Delta} \end{pmatrix} \tag{4.6}$$

In the above effective Hamiltonian (kept up to the second-order perturbation), the following are observed

- The leading coupling error shows up now as a derivative term coupling $|1\rangle$ and $|2\rangle$;
- The qubit level separation is slightly renormalized by the Stark shift. The drive is not perfectly resonant with the qubit frequency. Thus there exists a phase error;

- There is a new two-photon ϵ^2 leakage between $|0\rangle$ and $|2\rangle$. In this section, we do not deal with the ϵ^2-leakage as we only have one degree of freedom in the single-derivative DRAG.

It is important to point out that $\hat{H}_{0,1}$ is equivalent to $\hat{H}_{0,0}$, i.e., they generate the same dynamics. We have to introduce a correction terms to really suppress the error.

Similarly, we can compute the new drive Hamiltonians in this effective frame

$$\hat{H}_{y,1} = \begin{pmatrix} 0 & -\frac{1}{2}i\epsilon\Omega_y & -\frac{i\epsilon^2\lambda\Omega\Omega_y}{4\Delta} \\ \frac{1}{2}i\epsilon\Omega_y & 0 & -\frac{1}{2}i\epsilon\lambda\Omega_y \\ \frac{i\epsilon^2\lambda\Omega\Omega_y}{4\Delta} & \frac{1}{2}i\epsilon\lambda\Omega_y & 0 \end{pmatrix}. \tag{4.7}$$

To compensate for the leakage error from $|1\rangle$ to $|2\rangle$, we introduce the correction $\Omega_y = -\frac{i\dot{\Omega}}{\Delta}$ to remove the leading order leakage

$$\hat{H}_{1,1} = \hat{H}_{0,1} + \hat{H}_{y,1} = \begin{pmatrix} 0 & \frac{\epsilon\Omega}{2} + \frac{i\epsilon\dot{\Omega}}{2\Delta} & \frac{\epsilon^2\lambda\Omega^2}{4\Delta} + \frac{i\epsilon^2\lambda\Omega\dot{\Omega}}{4\Delta^2} \\ \frac{\epsilon\Omega}{2} - \frac{i\epsilon\dot{\Omega}}{2\Delta} & \delta\epsilon^2 - \frac{\epsilon^2\lambda^2\Omega^2}{4\Delta} & 0 \\ \frac{\epsilon^2\lambda\Omega^2}{4\Delta} - \frac{i\epsilon^2\lambda\Omega\dot{\Omega}}{4\Delta^2} & 0 & \Delta + 2\delta\epsilon^2 + \frac{\epsilon^2\lambda^2\Omega^2}{4\Delta} \end{pmatrix}. \tag{4.8}$$

The resulting Hamiltonian becomes decoupled from $|2\rangle$ in the first order. The correction also introduces a Y component to the target coupling. This slightly alters the rotation axis in a time-dependent manner and needs to be corrected to generate a perfect π rotation.

This imaginary Y component, together with the Stark shift, contributes to what is known as the phase error. Although the rotation is restricted in the qubit subspace, it does not implement a perfect rotation around the desired axis. In the following, we show that the Y error can be transformed into the Stark shift Z error, which will then be corrected with a time-dependent detuning δ. This technique is referred to as the interaction picture transformation in Ref. [67].

The idea is to use the Lie algebra structure to design a frame transformation that transforms time-dependent Y control to Z control. The strategy is to construct a generator $\hat{S}$ such that the derivative $i\dot{\hat{S}}$ cancels the Y term. The generator $\hat{S}$ is given by

$$\hat{S}_{1\to 2} = \begin{pmatrix} 0 & -\frac{\epsilon\Omega}{2\Delta} & 0 \\ \frac{\epsilon\Omega}{2\Delta} & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \tag{4.9}$$

which results in a new effective Hamiltonian

$$\hat{H}_{1,2} = \begin{pmatrix} -\frac{\epsilon^2\Omega^2}{2\Delta} & \frac{\epsilon\Omega}{2} & \frac{\epsilon^2\lambda\Omega^2}{4\Delta} + \frac{i\epsilon^2\lambda\Omega\dot{\Omega}}{4\Delta^2} \\ \frac{\epsilon\Omega}{2} & \delta\epsilon^2 + \frac{\epsilon^2\Omega^2}{2\Delta} - \frac{\epsilon^2\lambda^2\Omega^2}{4\Delta} & 0 \\ \frac{\epsilon^2\lambda\Omega^2}{4\Delta} - \frac{i\epsilon^2\lambda\Omega\dot{\Omega}}{4\Delta^2} & 0 & \Delta + 2\delta\epsilon^2 + \frac{\epsilon^2\lambda^2\Omega^2}{4\Delta} \end{pmatrix} \tag{4.10}$$

Notably, the Y error is transformed to a Stark shift between $|0\rangle$ and $|1\rangle$. The remaining error is a phase error that can be corrected using a time-dependent detuning or through phase ramping introduced in the next section.

In general, the error in the qubit subspace contributes significantly more to the gate error than the leakage, even if they are of the same or higher order. Simulation shows that it is necessary to include the ϵ^2 phase correction and the ϵ^3 correction to the pulse amplitude (Fig. 4 in [66]). In practice, the total rotation angle is usually simply calibrated by adjusting the total pulse amplitude. Since we only kept the ϵ^2 terms in the above calculation, we leave this to Section 4.7.

4.4 Detuning and phase ramping

As seen from $\hat{H}_{1,2}$ in equation (4.10), the phase error can be corrected by a time-dependent detuning δ, proportional to Ω^2. Although some superconducting qubit architecture has a tunable qubit frequency, it is often desired to implement it solely through microwave drive. In this context, the detuning is implemented via phase ramping [65].

The phase frame unitary for a truncated Transmon oscillator is defined as

$$\hat{V}_{\mathrm{PR}} = \exp\left(\hat{S}_{\mathrm{PR}}\right) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \mathrm{e}^{-i\theta_{\mathrm{PR}}} & 0 \\ 0 & 0 & \mathrm{e}^{-2i\theta_{\mathrm{PR}}} \end{pmatrix}. \tag{4.11}$$

The phase ramped Hamiltonian is given by $\hat{H}_{\mathrm{PR}} = \hat{V}_{\mathrm{PR}}^{\dagger}\hat{H}_{0,0}\hat{V}_{\mathrm{PR}} - i\dot{\hat{S}}_{\mathrm{PR}}$. Notice that the order of $\hat{V}_{\mathrm{PR}}^{\dagger}$ and $\hat{V}_{\mathrm{PR}}$ is exchanged and the sign of the derivative term is flipped. This is because it is the inverse transformation and $\hat{H}_{\mathrm{PR}}$ is the Hamiltonian that is implemented in the experiment, which should have no detuning

$$\hat{H}_{\mathrm{PR}} = \begin{pmatrix} 0 & \frac{1}{2}\mathrm{e}^{-i\theta_{\mathrm{PR}}}\epsilon\Omega & 0 \\ \frac{1}{2}\mathrm{e}^{i\theta_{\mathrm{PR}}}\epsilon\Omega & \delta - \dot{\theta}_{\mathrm{PR}} & \frac{1}{2}\mathrm{e}^{-i\theta_{\mathrm{PR}}}\epsilon\lambda\Omega \\ 0 & \frac{1}{2}\mathrm{e}^{i\theta_{\mathrm{PR}}}\epsilon\lambda\Omega & \Delta\epsilon^2 + 2\delta - 2\dot{\theta}_{\mathrm{PR}} \end{pmatrix}. \tag{4.12}$$

By choosing the time-dependent phase $\theta_{\mathrm{PR}}(t) = \int_0^t \delta(\tau)\mathrm{d}\tau$ for the drive, we implement an equivalent detuning $\delta(t)$.

For the DRAG pulse derived above, the correction drive is given as

$$\Omega_{\mathrm{PR}} = \left(\Omega(t) - i\frac{\dot{\Omega}(t)}{\Delta}\right)\exp\left(i\int_0^t \frac{\Omega(\tau)^2}{4\Delta}\left(\lambda^2 - 4\right)\mathrm{d}\tau\right). \tag{4.13}$$

If $\frac{\Omega}{\Delta}$ is small, this time-dependent detuning can be replaced by a constant one as in Ref. [134]. An important distinction between the previously defined frame transformation and the phase ramping lies in the fact that $\hat{V}_{\mathrm{PR}}(t)$ is not an identity at the end of the gate $t = T$. The implemented unitary is actually $\hat{U} = \hat{V}_{\mathrm{PR}}(t = T)\hat{U}_{\mathrm{PR}}\hat{V}_{\mathrm{PR}}^{\dagger}(t = 0)$. It is evident that $\hat{V}_{\mathrm{PR}}(t)$ is simply the accumulated detuning phase. To recover the target unitary $\hat{U}_{\mathrm{PR}}$, a virtual phase gate $\hat{V}_{\mathrm{PR}}^{\dagger}(t)$ must be added after the gate, effectively undoing the accumulated detuning.

4.5 The Y-only DRAG correction

Here, we introduce another variant of the first-derivative DRAG, considering the scenario where only Y control is available in the system and no phase ramping is implemented [135]. This is the earliest use of DRAG pulse in experiments. The difference lies mainly in the choice of the $(0,1)$ element of the generator $\hat{S}$ [66].

Since the phase error plays an important role in the fidelity measure, the DRAG correction here is designed to remove the phase error (or Y error) rather than the leakage. We chose the generator such that there is no phase mismatch between state $|0\rangle$ and $|1\rangle$ after the transformation:

$$\hat{S}_{0\to 3} = \begin{pmatrix} 0 & -\frac{\epsilon\lambda^2\Omega}{8\Delta} & 0 \\ \frac{\epsilon\lambda^2\Omega}{8\Delta} & 0 & -\frac{\epsilon\lambda\Omega}{2\Delta} \\ 0 & \frac{\epsilon\lambda\Omega}{2\Delta} & 0 \end{pmatrix}, \tag{4.14}$$

which gives the effective Hamiltonian

$$\hat{H}_{0,3} = \begin{pmatrix} -\frac{\epsilon^2\lambda^2\Omega^2}{8\Delta} & \frac{\epsilon\Omega}{2} - \frac{i\epsilon\lambda^2\dot{\Omega}}{8\Delta} & \frac{\epsilon^2\lambda\Omega^2}{4\Delta} - \frac{\epsilon^2\lambda^3\Omega^2}{32\Delta} \\ \frac{\epsilon\Omega}{2} + \frac{i\epsilon\lambda^2\dot{\Omega}}{8\Delta} & -\frac{\epsilon^2\lambda^2\Omega^2}{8\Delta} & -\frac{i\epsilon\lambda\dot{\Omega}}{2\Delta} \\ \frac{\epsilon^2\lambda\Omega^2}{4\Delta} - \frac{\epsilon^2\lambda^3\Omega^2}{32\Delta} & \frac{i\epsilon\lambda\dot{\Omega}}{2\Delta} & \Delta + \frac{\epsilon^2\lambda^2\Omega^2}{4\Delta} \end{pmatrix}. \tag{4.15}$$

The Y drive Hamiltonian reads in this effective frame

$$\hat{H}_{y,3} = \begin{pmatrix} 0 & -\frac{1}{2}i\epsilon\Omega_y & -\frac{i\epsilon^2\lambda\Omega\Omega_y}{4\Delta} + \frac{i\epsilon^2\lambda^3\Omega\Omega_y}{16\Delta} \\ \frac{1}{2}i\epsilon\Omega_y & 0 & -\frac{1}{2}i\epsilon\lambda\Omega_y \\ \frac{i\epsilon^2\lambda\Omega\Omega_y}{4\Delta} - \frac{i\epsilon^2\lambda^3\Omega\Omega_y}{16\Delta} & \frac{1}{2}i\epsilon\lambda\Omega_y & 0 \end{pmatrix}. \tag{4.16}$$

Applying the Y-DRAG correction $\Omega_y \to -\frac{\lambda^2\dot{\Omega}}{4\Delta}$ removes the Y error and also partially suppresses the leakage. In particular, if the Duffing model is used, with $\lambda = \sqrt{2}$, the DRAG coefficient is exactly $1/2$ and half of the leakage coupling is removed. Hence, it is often referred to as the half-DRAG correction.

4.6 An alternative derivation of the first-derivative DRAG

In this section, we provide an alternative derivation of the first-derivative DRAG, highlighting the equivalence to Section 4.3 and also some physical intuition behind the DRAG design. We start again from the $\hat{H}_{0,0}$. Instead of apply two transformations $\hat{S}_{0\to 1}$ and $\hat{S}_{1\to 2}$ sequentially, we combine them into a single generator $\hat{S}_{0\to 4}$

$$\hat{S}_{0\to 4} = \begin{pmatrix} 0 & -\frac{\epsilon\Omega}{2\Delta} & 0 \\ \frac{\epsilon\Omega}{2\Delta} & 0 & -\frac{\epsilon\lambda\Omega}{2\Delta} \\ 0 & \frac{\epsilon\lambda\Omega}{2\Delta} & 0 \end{pmatrix}. \tag{4.17}$$

Notice that the transformation is equivalent in the leading order, i.e., $\hat{S}_{0\to4}$ also diagonalizes the leakage coupling and removes the Y term. And $\hat{S}_{0\to4}$ commute with itself at any time t. The mathematical argument behind it is that $\hat{S}_{0\to4}$ commutes with the $\hat{H}_y$ drive. Therefore, we can use $\hat{H}_y$ to completely remove the derivative $\dot{\hat{S}}_{0\to4}$, without leaving any residual terms in the Hamiltonian after the correction.

This becomes obvious after calculating the effective Hamiltonian

$$\hat{H}_{0,4} = \begin{pmatrix} -\frac{\epsilon^2\Omega^2}{2\Delta} & \frac{\epsilon\Omega}{2} - \frac{i\epsilon\dot{\Omega}}{2\Delta} & \frac{\epsilon^2\lambda\Omega^2}{8\Delta} \\ \frac{\epsilon\Omega}{2} + \frac{i\epsilon\dot{\Omega}}{2\Delta} & \frac{\epsilon^2\Omega^2}{2\Delta} - \frac{\epsilon^2\lambda^2\Omega^2}{4\Delta} & -\frac{i\epsilon\lambda\dot{\Omega}}{2\Delta} \\ \frac{\epsilon^2\lambda\Omega^2}{8\Delta} & \frac{i\epsilon\lambda\dot{\Omega}}{2\Delta} & \Delta + \frac{\epsilon^2\lambda^2\Omega^2}{4\Delta} \end{pmatrix}, \tag{4.18}$$

where the derivative terms align with the Y drive of the Hamiltonian. In addition, we have $\hat{H}_{y,4} = \hat{H}_{y,0}$ for the same reason. Notice that $\hat{H}_{0,4}$ is equivalent to $\hat{H}_{0,1}$, up to the second order.

To correctly calculate the ϵ^3 correction to the amplitude later in the next section, we also compute the Z drive in this effective frame

$$\hat{H}_{z,4} = \epsilon^2 \begin{pmatrix} 0 & -\frac{\delta\epsilon\Omega}{2\Delta} & 0 \\ -\frac{\delta\epsilon\Omega}{2\Delta} & \delta & -\frac{\delta\epsilon\lambda\Omega}{2\Delta} \\ 0 & -\frac{\delta\epsilon\lambda\Omega}{2\Delta} & 2\delta \end{pmatrix}. \tag{4.19}$$

Applying the Y-DRAG correction and the time-dependent detuning, we obtain

$$\hat{H}_{2,4} = \begin{pmatrix} -\frac{\epsilon^2\Omega^2}{2\Delta} & \frac{\epsilon\Omega}{2} & \frac{\epsilon^2\lambda\Omega^2}{8\Delta} \\ \frac{\epsilon\Omega}{2} & -\frac{\epsilon^2\Omega^2}{2\Delta} & 0 \\ \frac{\epsilon^2\lambda\Omega^2}{8\Delta} & 0 & \Delta - \frac{2\epsilon^2\Omega^2}{\Delta} + \frac{3\epsilon^2\lambda^2\Omega^2}{4\Delta} \end{pmatrix}. \tag{4.20}$$

We adopt this as the complete first-derivative DRAG solution. The next order error is clearly the ϵ^2-leakage between $|0\rangle$ and $|2\rangle$, which is the same as the one in equation (4.10), up to an integral by part [67]. We will address this in Section 4.7.

4.7 High-order DRAG correction

The first-derivative DRAG has been widely adopted for Transmon qubits in the last decades, with the gate time pushed to 10 ns for a π-rotation [134]. However, the observed optimal DRAG correction strength deviates from the analytical value and needs to be calibrated experimentally. Albeit from hardware derivation such as pulse distortion and cable attenuation, the shift of the DRAG parameter is also observed in simulation under a strong drive. This is mainly because the derived first-derivative DRAG solution does not consider the higher order corrections, and, thus, the DRAG coefficient needs to be optimized to balance between the two leakage transitions from $|1\rangle$ and $|0\rangle$.

Here, we go beyond the first-derivative DRAG and search for higher-order solutions, in particular, we derive the correction to the $|0\rangle \leftrightarrow |2\rangle$ leakage error.

The derivation is based on Section VI of Ref. [67]. Note that to be consistent with the notation in this chapter, we add a factor of 1/2 to the drive Hamiltonian in Ref. [67].

We start from equation (4.20), with the two-photon transition $|0\rangle \leftrightarrow |2\rangle$. Effectively, we need to engineer the frequency property of Ω^2. For these kinds of multi-photon transitions, it is easier to use the recursive formulation. In the recursive formulation, instead of deriving corrections as an expression of the original drive pulse Ω, we derive recursive expressions of the pulse $\Omega = f_1(\Omega_1), \Omega_1 = f_2(\Omega_2), \cdots, \Omega_{k-1} = f_k(\Omega_k))$ where the Ω_k is the initial pulse shape. In general, the substitution f_k is not always expressed as a linear combination of Ω_k and corrections.

For the $|0\rangle \leftrightarrow |2\rangle$ error, we choose a generator $\hat{S}_{4\to5}$ that is defined by a different drive profile Ω_1. With this definition, we can construct an expression Ω such that the $|0\rangle \leftrightarrow |2\rangle$ coupling is zero in the effective frame:

$$\hat{S}_{4\to5} = \begin{pmatrix} 0 & 0 & -\frac{\epsilon^2\lambda\Omega_1^2}{8\Delta^2} \\ 0 & 0 & 0 \\ \frac{\epsilon^2\lambda\Omega_1^2}{8\Delta^2} & 0 & 0 \end{pmatrix}, \tag{4.21}$$

and

$$\hat{H}_{2,5} = \begin{pmatrix} -\frac{\epsilon^2\Omega^2}{2\Delta} & \frac{\epsilon\Omega}{2} & \frac{\epsilon^2\lambda\Omega^2}{8\Delta} - \frac{\epsilon^2\lambda\Omega_1^2}{8\Delta} - \frac{i\epsilon^2\lambda\Omega_1\dot{\Omega}_1}{4\Delta^2} \\ \frac{\epsilon\Omega}{2} & -\frac{\epsilon^2\Omega^2}{2\Delta} & 0 \\ \frac{\epsilon^2\lambda\Omega^2}{8\Delta} - \frac{\epsilon^2\lambda\Omega_1^2}{8\Delta} + \frac{i\epsilon^2\lambda\Omega_1\dot{\Omega}_1}{4\Delta^2} & 0 & \Delta - \frac{2\epsilon^2\Omega^2}{\Delta} + \frac{3\epsilon^2\lambda^2\Omega^2}{4\Delta} \end{pmatrix} \tag{4.22}$$

Notice that since $\hat{S}_{4\to5}$ is already a ϵ^2 term, the transformation does not change the other matrix elements up to our truncation order ϵ^3.

From the above Hamiltonian, we see that we can remove this $|0\rangle \leftrightarrow |2\rangle$ leakage, if we find an expression $\Omega\,(\Omega_1)$ such that the matrix element is zero. However, $\hat{H}_{1,5}$ has an imaginary term, which cannot be incorporated because we have assumed that Ω is real. In Chapter 5, we will stop here and use the imaginary term directly, but the derivation needs to be changed to keep the Hamiltonian hermitian. Here, for single-qubit gates, assuming Ω to be real simplifies the handling of the Y error. To address this, we perform again a transformation targeted at that imaginary term only, which would then give back a real term with an additional derivative.

$$\hat{S}_{5\to6} = \begin{pmatrix} 0 & 0 & \frac{i\epsilon^2\lambda\Omega_1\dot{\Omega}_1}{4\Delta^3} \\ 0 & 0 & 0 \\ \frac{i\epsilon^2\lambda\Omega_1\dot{\Omega}_1}{4\Delta^3} & 0 & 0 \end{pmatrix}, \tag{4.23}$$

which results in

$$\hat{H}_{2,6} = \begin{pmatrix} -\frac{\epsilon^2\Omega^2}{2\Delta} & \frac{\epsilon\Omega}{2}+\frac{\epsilon^3\Omega^3}{4\Delta^2} & \frac{\epsilon^2\lambda\Omega^2}{8\Delta}-\frac{\epsilon^2\lambda\Omega_1^2}{8\Delta}-\frac{\epsilon^2\lambda\left(\dot{\Omega}_1^2+\Omega_1\ddot{\Omega}_1\right)}{4\Delta^3} \\ \frac{\epsilon\Omega}{2}+\frac{\epsilon^3\Omega^3}{4\Delta^2} & -\frac{\epsilon^2\Omega^2}{2\Delta} & 0 \\ \frac{\epsilon^2\lambda\Omega^2}{8\Delta}-\frac{\epsilon^2\lambda\Omega_1^2}{8\Delta}-\frac{\epsilon^2\lambda\left(\dot{\Omega}_1^2+\Omega_1\ddot{\Omega}_1\right)}{4\Delta^3} & 0 & \Delta-\frac{2\epsilon^2\Omega^2}{\Delta}+\frac{3\epsilon^2\lambda^2\Omega^2}{4\Delta} \end{pmatrix} \tag{4.24}$$

In the above expression, we kept the ϵ^3-correction to the drive amplitude $-\frac{\Omega^3}{2\Delta^2}$. This is obtained by performing all the above calculations while always keeping the third-order corrections to the $|0\rangle \leftrightarrow |1\rangle$ coupling.

The solution for the $|0\rangle \leftrightarrow |2\rangle$ leakage can be derived as $\Omega^2 = \Omega_1^2 + 2\frac{\left(\dot{\Omega}_1\right)^2+\Omega_1\ddot{\Omega}_1}{\Delta^2}$. A real solution for Ω exist for large enough Δ compared to $\Omega_1\ddot{\Omega}_1$. Altogether, we have the final recursive solution[1] :

$$\Omega_{\mathrm{PR}} = \left(\Omega - i\frac{\dot{\Omega}}{\Delta} - \frac{\Omega^3}{2\Delta^2}\right)\exp\left(i\int_0^t \frac{\Omega^2}{4\Delta}\left(\lambda^2-4\right)\mathrm{d}\tau\right), \tag{4.25}$$

$$\Omega = \sqrt{\Omega_1^2 + 2\frac{\dot{\Omega}_1^2+\Omega_1\ddot{\Omega}_1}{\Delta^2}}. \tag{4.26}$$

The initial pulse shape Ω_1 needs to be chosen such that all the derivatives up to the third order must start and end in zero.

[1] There is a factor of 2 missing in Ref. [67]. If $\Delta \to \infty$, all the corrections should converge to zero and one gets the original pulse shape Ω, while in the paper it actually converges to $\Omega/\sqrt{2}$.

5

Experimental error suppression in Cross-Resonance gates via DRAG

While quantum circuits are reaching impressive widths in the hundreds of qubits, their depths have not been able to keep pace. In particular, cloud computing gates on multi-qubit, fixed-frequency superconducting chips continue to hover around the 1% error range, contrasting with the progress seen on carefully designed two-qubit chips, where error rates have been pushed towards 0.1%. Despite the strong impetus and a plethora of research, experimental demonstration of error suppression on these multi-qubit devices remains challenging, primarily due to the wide distribution of qubit parameters and the demanding calibration process required for advanced control methods. Here, we achieve this goal, using a simple control method based on multi-derivative, multi-constraint pulse shaping, which acts simultaneously against multiple error sources. Our approach establishes a two to fourfold improvement on the default calibration scheme, demonstrated on four qubits on the IBM Quantum Platform with limited and intermittent access, enabling these large-scale fixed-frequency systems to fully take advantage of their superior coherence times. The achieved CNOT fidelities of 99.7(1)% on those publically available qubits come from both coherent control error suppression and accelerated gate time.

This chapter has been published, with minor changes, as Boxi Li, Tommaso Calarco, and Felix Motzoi, *Experimental error suppression in Cross-Resonance gates via multi-derivative pulse shaping*, npj Quantum Information 10.1 (2024): 66. [136]. It has been accepted by npj Quantum Information. The thesis author conducted most of the analysis and all the experiments, composed all the figures and wrote the manuscript with input from the advisors.

5.1 Introduction

Superconducting qubits have experienced significant improvement in the last decade, reaching the error correction threshold [137, 138] and been used to study nontrivial quantum phenomena [139, 140]. Additionally, quantum devices have become more accessible outside research labs, with cloud-based platforms like the IBM Quantum platform [141] providing access to multi-qubit devices, on which near-term quantum computing applications with error mitigation have been demonstrated [142]. Although very high gate fidelity has been achieved on isolated chips [106, 143], gate performance on scalable, publicly available multi-qubit devices is still bottlenecked, especially for two-qubit operations [144]. These control imperfections not only limit the fidelity and depth of quantum circuits, but also give rise to correlated errors that propagate across the qubit lattice, sabotaging quantum error correction [40], making error mitigation and benchmarking more challenging [145–148].

The Cross-Resonance (CR) gate is one of the most widely used two-qubit entangling gates for superconducting qubits, using microwave controls and avoiding noisy flux lines [122, 123, 149–151]. It is the default gate on most devices provided by IBM and has found applications in high-quality circuit implementation, parity measurement, and state preparation [106, 152, 153]. While the absence of flux control lines extends qubit coherence time, it limits qubit tunability and necessitates weak coupling between qubits. Consequently, achieving fast two-qubit gates requires a strong drive, which often leads to coherent errors due to non-adiabatic dynamics. In practice, limiting the drive amplitude and a long pulse ramping time are used to prevent undesired dynamics, including off-resonant transitions introduced by the drive [89, 154, 155] and unwanted dynamics in the effective qubits' subspace [80, 81, 154] (see Figure 5.1a and Figure 5.1b).

To circumvent control errors while maintaining substantial coupling strength, a combination of careful qubit parameter engineering and advanced control schemes has been employed. With these techniques, the best CR gate infidelity reported lies between 0.1% and 0.3% [106, 143]. However, extending these advancements to scalable multi-qubit devices proves challenging. For instance, on the 127-qubit chip *ibm_brisbane*, the best echoed CR gate has an error of 0.35%, while the median is only around 0.8%, considerably higher than their coherence limit (median T1$\approx$200μs and T2$\approx$135μs) [141]. An important factor contributing to the challenge lies in the intentional distribution of qubit parameters over a wide range, a design choice aimed at mitigating cross-talk [39, 121, 156]. This uncertainty in the qubit parameters also stems unavoidably from the inhomogeneity in the fabrication process [121, 157]. Therefore, designing an efficient control scheme that operates seamlessly across diverse parameter regimes is essential for achieving optimal performance across hundreds of qubits.

In this chapter, we devise and test a simple and scalable control scheme for CR gates that counteracts all the aforementioned control errors, following the ideal of the well-known Derivative Removal by Adiabatic Gate (DRAG) method [64, 66, 67, 132]. For the transition error, we demonstrate that previously suggested single-derivative DRAG correction [89] is insufficient for typical parameters in multi-qubit devices,

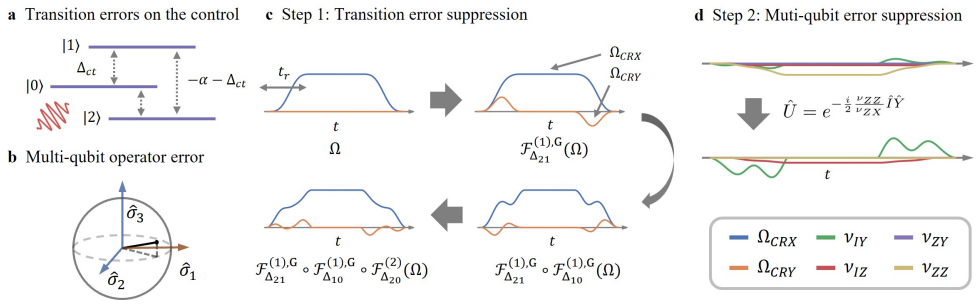


Figure 5.1: Illustration of the dominant coherent errors in the Cross-Resonance gate and the proposed pulse schemes to correct them. **a.** Transition errors in the rotating frame for a Transmon qubit driven off-resonantly, with Δ_{ct} the detuning and α the anharmonicity. **b.** Two-qubit errors of the CR gate. The three axes $\{\hat{\sigma}_1, \hat{\sigma}_2, \hat{\sigma}_3\}$ represent either $\{\hat{\sigma}_{ZX}, \hat{\sigma}_{ZY}, \hat{\sigma}_{ZZ}\}$ or $\{\hat{\sigma}_{IX}, \hat{\sigma}_{IY}, \hat{\sigma}_{IZ}\}$. The Hamiltonians ZX and IX (brown) commute and are defined as the ideal dynamics, while the others are considered errors (blue). **c.** Schematic illustration of the recursive DRAG pulse that suppresses different error transitions on the control Transmon. **d.** Schematic illustration of different multi-qubit errors in the effective frame during the CR operation, and the transformations of the error terms. The remaining IY and IZ errors are compensated for by an IY-DRAG pulse on the target qubit and the detuning of the CR drive.

where multiple transition errors are present. We introduce a novel recursive multi-derivative DRAG pulse, considering all three possible error transitions, capable of experimentally suppressing the error to high precision without the need for any calibration, or additional free parameters. For the two-qubit rotation operator error such as the ZZ error in the effective two qubits' subspace, we present a new approach. While other schemes typically involve hardware modifications or additional detuned microwave drive terms [107, 143, 158], we show that a simple, DRAG-like correction tone applied on the target qubit, along with a detuning on the drive frequency, is sufficient to eliminate dominant entangling error terms while avoiding additional hardware engineering. An overview of the derived pulse schemes is shown in Figure 5.1c and Figure 5.1d.

In comparison to alternative pulse shaping techniques [126, 159, 160], this multi-derivative pulse Ansatz stands out for its simplicity. It provides an efficient parameterization of the control pulse as a simple expression of the qubits' frequency and anharmonicity. This simplicity is essential for scalable quantum devices as all the qubits need to be calibrated quickly and repeatedly to ensure high fidelity. With the qiskit-pulse [161] interface, we implemented our drive scheme on multi-qubit devices provided by IBM Quantum. Despite the limited calibration time due to sporadic access to busy, public machines, our experimental results validate the efficient suppression of coherent errors. We observe a two- to fourfold reduction in infidelity, achieving beyond state-of-the-art fidelities in the 99.6-99.8% range on

multiple qubit pairs publicly available on the IBM platform.

The rest of the chapter is organised as follows: We start by presenting the theoretical framework of the derivative-based pulse shaping methods. Next, we derive the pulse schemes for the CR gate and experimentally validate the error suppression for both the transition errors on the control qubit and the multi-qubit errors. Finally, we demonstrate the performance and scalability of the proposed control scheme by benchmarking the custom-implemented CR gate on multi-qubit quantum hardware, accompanied by numerical simulations across a wide range of experimentally relevant regimes.

5.2 Multi-derivative pulse shaping

We start explaining the general theory for the systematic, iterative error suppression with a generic two-level system

$$\hat{H} = \Delta\hat{\Pi}_j + g(t)\frac{\hat{\sigma}^+_{jk}}{2} + g^*(t)\frac{\hat{\sigma}^+_{kj}}{2} \tag{5.1}$$

where $\hat{\Pi}_j = |j\rangle\langle j|$ and $\hat{\sigma}^+_{jk} = |k\rangle\langle j|$, $g(t)$ denotes the coupling strength between the two levels. In the following, we omit the explicit time dependence on t for ease of notation. In general, g could take the (perturbative) form of an n-photon interaction, $\frac{\Omega^n}{\Delta_{\text{eff}}^{n-1}}$, where Δ_{eff} is an effective energy gap and Ω the drive strength. In particular, if Ω denotes the CR drive strength on the control qubit, with $n = 1$, it describes the transition $|0\rangle \leftrightarrow |1\rangle$ (or $|1\rangle \leftrightarrow |2\rangle$) and with $n = 2$ the two-photon transition $|0\rangle \leftrightarrow |2\rangle$.

The goal is to suppress the undesired transition introduced by the coupling g. If $g \ll \Delta$, we may perform a perturbative expansion with the antihermitian generator $\hat{S}(\tilde{g}) = \frac{\tilde{g}}{2\Delta}\hat{\sigma}^+_{jk} - h.c.$ and obtain under the transformation $\hat{H}'(g) = \hat{V}(\tilde{g})\hat{H}(g)\hat{V}^\dagger(\tilde{g}) + i\dot{\hat{V}}(\tilde{g})\hat{V}^\dagger(\tilde{g})$ with $\hat{V}(\tilde{g}) = \mathrm{e}^{\hat{S}(\tilde{g})}$,

$$\begin{aligned}\hat{H}'(g) =& i\dot{\hat{S}}(\tilde{g}) + \hat{H}(g) + [\hat{S}(\tilde{g}), \hat{H}(g)] + \cdots \\ =& \Delta\hat{\Pi}_j + \left(g - \tilde{g} + i\frac{\mathrm{d}}{\mathrm{d}t}\frac{\tilde{g}}{\Delta}\right)\frac{\hat{\sigma}^+_{jk}}{2} + h.c. + \mathcal{O}(\epsilon^2),\end{aligned} \tag{5.2}$$

where $\epsilon \propto g/\Delta$. We deliberately distinguish between g, the actual physical coupling, and $\tilde{g}$, which is used to define the generator $\hat{S}$ that diagonalizes the Hamiltonian. As a result, for a time-dependent coupling g, to suppress the transition, we require

$$g = \tilde{g} - i\frac{\mathrm{d}}{\mathrm{d}t}\frac{\tilde{g}}{\Delta}. \tag{5.3}$$

The above equation also provides an alternative interpretation: Transition-less evolution is possible if we find a (counter-diabatic) control $g(t)$ by choosing any continuous function $\tilde{g}(t)$ and making sure that $\hat{S}(\tilde{g})$ is zero at the beginning and

at the end of the time evolution [48–50, 133]. Thus, equation (5.3) provides a substitution rule to derive a time-modulated coupling $g(t)$ with the transition between the two levels suppressed. If the coupling describes an n photon interaction generated by a drive Ω, i.e., $g = \frac{\Omega^n}{\Delta_{\text{eff}}^{n-1}}$ with a constant Δ_{eff}, we obtain

$$\Omega = \mathcal{F}_{\Delta}^{(n)}(\tilde{\Omega}) \coloneqq \left(\tilde{\Omega}^n - i\frac{\mathrm{d}}{\mathrm{d}t}\frac{\tilde{\Omega}^n}{\Delta}\right)^{\frac{1}{n}}. \tag{5.4}$$

Here, we choose $\tilde{g} = \frac{\tilde{\Omega}^n}{\Delta_{\text{eff}}^{n-1}}$ to keep the notation intuitive. The fractional exponent is defined for complex numbers and needs to ensure the continuity of Ω as a function of t. For $n = 1$ this gives the familiar result of the single-derivative DRAG expansion [67]. If needed, a free parameter a can be added before the derivative term to adjust the strength DRAG correction.

More generally, a two-level Hamiltonian (or subspace), in equation (5.1), is diagonalized exactly by the unitary transformation (referred to as Givens rotation) [73]

$$\hat{V} = \begin{pmatrix} \cos\left(\frac{\theta}{2}\right) & \mathrm{e}^{-i\phi}\sin\left(\frac{\theta}{2}\right) \\ -\mathrm{e}^{i\phi}\sin\left(\frac{\theta}{2}\right) & \cos\left(\frac{\theta}{2}\right) \end{pmatrix}, \tag{5.5}$$

resulting in an exact substitution rule [c.f. equation (5.3)]:

$$g = \mathrm{e}^{i\phi}\left(-(\Delta + \dot{\phi})\tan(\theta) + i\dot{\theta}\right), \tag{5.6}$$

where θ and ϕ can in principle be chosen arbitrarily provided $\hat{V} = \mathbb{1}$ at the beginning and the end of the drive.

This exact diagonalization becomes useful in scenarios involving strong drive amplitudes or small detunings. In those cases, equation (5.6) provides a compact expression for the DRAG pulse beyond the perturbation limit. To be consistent with the perturbative solution, we set $\theta = \arctan(-|\tilde{g}|/\Delta)$ and define ϕ as the complex phase of the coupling, i.e., $\tilde{g} = \mathrm{e}^{i\phi}|\tilde{g}|$. We note that, in general, Δ could also depend on g and equation (5.6) becomes an implicit equation for g instead of a closed-form expression. To obtain an expression for the drive strength Ω, one needs to invert the dependence of $g = f(\Omega)$. For instance, for a linear dependence, $g = \kappa\Omega$ (and $\tilde{g} = \kappa\tilde{\Omega}$), with κ a constant factor, we get

$$\Omega = \mathcal{F}_{\Delta}^{(1),\mathrm{G}}(\tilde{\Omega}) \coloneqq \frac{\Delta + \dot{\phi}_{\tilde{\Omega}}}{\Delta}\tilde{\Omega} + \frac{i\mathrm{e}^{i\phi_{\tilde{\Omega}}}}{\kappa}\frac{\mathrm{d}}{\mathrm{d}t}\arctan\left(-\frac{|\kappa\tilde{\Omega}|}{\Delta}\right) \tag{5.7}$$

with $\mathrm{e}^{i\phi_{\tilde{\Omega}}} = \tilde{\Omega}/|\tilde{\Omega}|$. Equation (5.4) and equation (5.7) will be the building blocks throughout the remaining of this article as we extend our analysis to multilevel systems.

5.3 Application to control-qubit errors

The CR interaction is typically activated by driving the control qubit with the frequency of the target [80, 81, 154], leading to a rotation in the target qubit

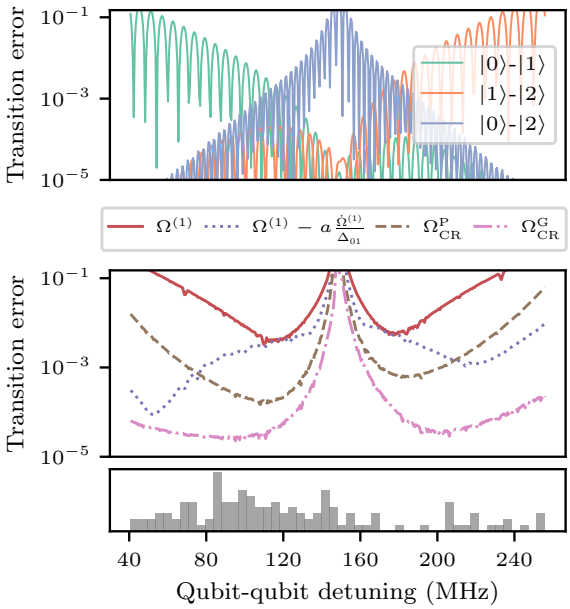


Figure 5.2: **Top:** Simulated transition error among the 3 levels of the control Transmon introduced by the CR drive using a flat-top Gaussian pulse with $t_{\mathrm{r}} = 2\sigma = 10$ ns. **Middle:** The total transition error for different pulse schemes. We plot the envelope of the oscillation by taking the maximum over different pulse lengths with the ramping time $t_{\mathrm{r}} = 10$ ns fixed. Parameters used are $\Omega_{\mathrm{max}}/2\pi = 30$ MHz, $(\Delta_{21} - \Delta_{10})/2\pi = -300$ MHz and $\lambda = \sqrt{2}$. **Bottom:** Distribution of the qubit-qubit detunings in *ibm_brisbane*. A few outliers that are far away out of the studied range are left out.

depending on the state of the control, equivalent to a CNOT gate up to single-qubit operations. Ideally, the state of the control qubit should remain unaltered at the end of the gate. However, despite the detuning, the drive may still excite the control qubit, especially when operating in the straddling regime for fast entanglement [81, 154], where the qubit-qubit detuning is smaller than their anharmonicities. Depending on the parameter regimes, it manifests both as single-photon transitions between $|0\rangle \leftrightarrow |1\rangle$, $|1\rangle \leftrightarrow |2\rangle$ as well as the two-photon transition between $|0\rangle \leftrightarrow |2\rangle$ [89].

To counter these transition errors, the single-derivative DRAG pulse has been employed which introduces a term proportional to the first derivative of the drive pulse, i.e., $\Omega - ia\frac{\dot{\Omega}}{\Delta}$, with a constant factor a to be optimized [89]. This heuristic is proven useful when the qubit-qubit detuning is very small, ranging from 50 MHz to 70 MHz [106, 155], because in this range there is only one dominant transition. In contrast, for the scalable, multi-qubit fixed-frequency architecture such as the

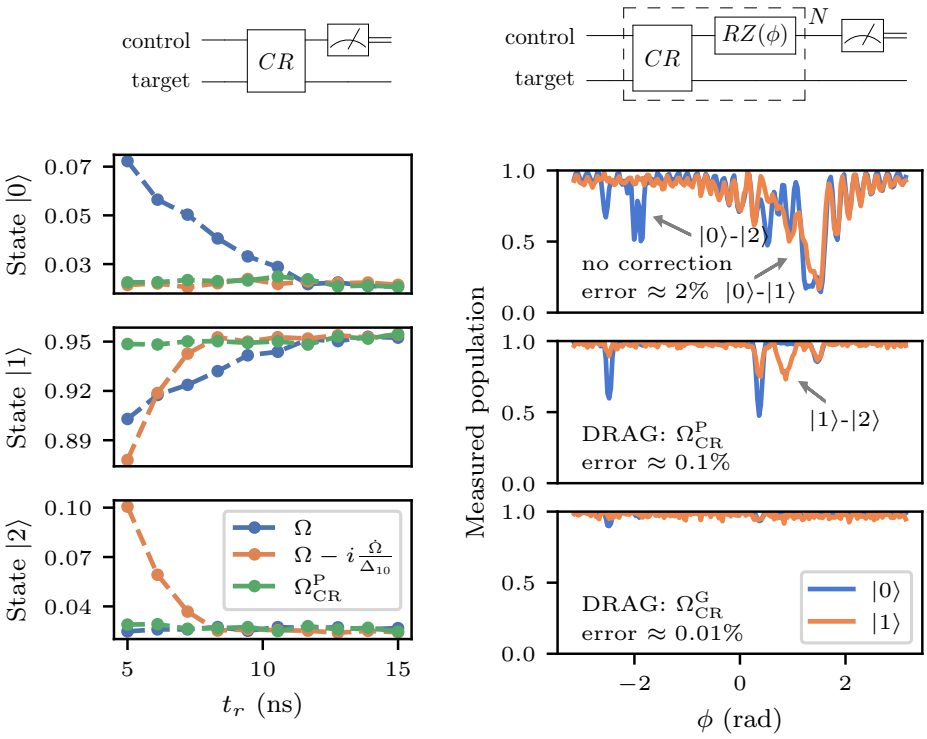


Figure 5.3: **Left.** The circuit and measured transition after preparing the state in $|1\rangle$ and applying a CR pulse with different rising time t_r. The data is obtained with a detuning of 120 MHz and a drive strength of about 40 MHz. **Right.** The amplification circuit and measured transition error. Deviations from the expected population of one indicate the presence of error. The three plots correspond to a default flat-top Gaussian, recursive DRAG pulse $\Omega^\mathrm{P}_\mathrm{CR}$ in equation (5.8) and $\Omega^\mathrm{G}_\mathrm{CR}$ in equation (5.9), with no calibration of additional parameters. The data is obtained from *ibm_nairobi* Q2 and Q1 with the drive amplitude 0.5 ($\approx$ 60 MHz), $t_\mathrm{r} = 10$ ns and $N = 30$. The qubit-qubit detuning is about 104 MHz.

IBM Quantum plaform, the detuning between neighbouring qubits is distributed over a much broader range (Figure 5.3c). An efficient drive scheme must be able to suppress the error in all the operating parameter regimes. As shown later in this section, in some prevalent parameter regimes, the single-derivative DRAG pulse provides only minimal improvement. Even with a numerically optimized scale factor, a compromise arises among different transitions [89].

Following the multi-derivative pulse described above, we propose the following pulse shape derived by recursively applying the DRAG correction targeting at the three dominant transitions

$$\Omega_{\mathrm{CR}}^{\mathrm{P}} = \mathcal{F}_{\Delta_{21}}^{(1)} \circ \mathcal{F}_{\Delta_{10}}^{(1)} \circ \mathcal{F}_{\Delta_{20}}^{(2)}(\Omega) \tag{5.8}$$

with the perturbative substitution equation (5.4) or

$$\Omega_{\mathrm{CR}}^{\mathrm{G}} = \mathcal{F}_{\Delta_{21}}^{(1),\mathrm{G}} \circ \mathcal{F}_{\Delta_{10}}^{(1),\mathrm{G}} \circ \mathcal{F}_{\Delta_{20}}^{(2)}(\Omega) \tag{5.9}$$

using the exact expression equation (5.7) for the two single-photon transitions. The energy difference between state $|j\rangle$ and $|k\rangle$ in the rotating frame is denoted by Δ_{jk}. The symbol $\circ$ denotes the composition of different substitutions $\mathcal{F}$, applied sequentially from right to left on the pulse shape. Recursively chaining the DRAG correction as above suppresses all three dominant errors. While the two single-photon corrections $\mathcal{F}^{(1)}$ are interchangeable, the substitution for the two-photon transitions needs to be applied first, as detailed in the Methods section. The explicit formula for the perturbative DRAG pulse [equation (5.8)] is given by the following recursive expressions:

$$\Omega_{\mathrm{CR}}^{\mathrm{P}} = \Omega_1 - i\frac{\dot{\Omega}_1}{\Delta_{10}}, \tag{5.10}$$

$$\Omega_1 = \Omega_2 - i\frac{\dot{\Omega}_2}{\Delta_{21}}, \tag{5.11}$$

$$\Omega_2 = \sqrt{\Omega_3^2 - i\frac{2\Omega_3\dot{\Omega}_3}{\Delta_{20}}}. \tag{5.12}$$

Here, Ω_3 needs to be chosen such that the obtained pulse is continuous and starts and ends in zero. Without the last equation for the two-photon transition, the derived pulse aligns with the multi-derivative DRAG solution proposed for multiple linear couplings in [67]. Notably, if one of the DRAG correction strengths is fine-tuned by an additional parameter, it will not affect the other correction significantly because of the recursive design.

Typically, a CR pulse consists of a rising, a holding and a lowering period, during which the pulse is turned on from zero to the maximum, held for a while and then turned off. We choose the rising portion of the pulse to be

$$\Omega^{(m)}(t) = \Omega_{\max}\mathcal{I}_0 \int_0^t \mathrm{d}t' \sin^m\left(\frac{\pi t'}{2t_{\mathrm{r}}}\right), \quad 0 \le t \le t_{\mathrm{r}} \tag{5.13}$$

with the normalization $\mathcal{I}_0$ fixed via $\Omega^{(m)}(t_r) = \Omega_{max}$. This definition ensures that the pulse is m times differentiable and the derivatives are zero at $t = 0$ and $t = t_r$, which guarantees the validity of the frame transformation $\hat{V}$ introduced above. Other pulse shapes can also be used as long as this property is satisfied. After the holding time, the lowering phase takes the time-reversed shape. An example of the CR pulse is shown in Figure 5.1c. For $m = 1$ and with zero holding time, the pulse is the same as the Hann window, very close to the flat-top Gaussian pulse commonly adopted. It is important to note that, as m increases, more high-frequency components become incorporated into the pulse shape, leading to higher non-adiabatic transition error if not compensated for. Therefore, it is advisable to keep m as small as possible. For our study, we use $m = 3$ as the initial shape for the recursive DRAG pulse.

To verify the performance of the error suppression, we numerically simulate the dynamics of the three-level Hamiltonian of the control Transmon (see Methods) and the result is shown in Figure 5.3a and Figure 5.3b. First, we examine in Figure 5.3a the contribution of the three transition errors for an uncorrected pulse, across the typical experimentally relevant qubit-qubit detuning values. The error is defined as the probability of unwanted population transfer among different states. The plot indicates that all three transitions must be considered for sufficient error suppression. Moreover, we observe that partial suppression of the errors (using only one or two derivatives) may increase the unsuppressed ones, as demonstrated in detail in Section 5.8.1, making them non-negligible even if they were initially small.

Next, we compare the total transition error introduced by different pulse schemes in Figure 5.3b. To better illustrate the difference between the pulse schemes, we take the sum of the three transition errors and the maximum over pulses with various holding lengths. In this way, the oscillation caused by the pulse timing is removed and only the upper envelope remains. As a baseline, we plot the error for pulse shape $\Omega^{(1)}$, which is similar to the flat-top Gaussian pulse used in qiskit-pulse [161]. The recursive DRAG pulse shapes we derived, Ω_{CR}^{P} and Ω_{CR}^{G}, suppress the error by several orders of magnitude, without any numerical optimization, as long as the drive is not resonant with the two-photon transition. In comparison, the single-derivative DRAG pulse, used in previous works [89, 106, 155], performs well only when the error is dominated by one single-photon transition (very large or very small detuning). Outside of these regimes, its performance is restricted due to the compromise between different transitions, even if the DRAG coefficient a is calibrated to minimize the total error.

This observation is further supported by experimental results shown in Figure 5.3d and Figure 5.3e. In Figure 5.3d, the state was initialized in state $|1\rangle$, and a CR pulse of 200 ns with varying rising times t_r was applied. As the rising time decreases, the error transition grows quadratically. Without any correction, the error is dominated by the transition between $|0\rangle \leftrightarrow |1\rangle$. Applying a DRAG pulse designed to suppress this transition, i.e., $\Omega - i\dot{\Omega}/\Delta_{10}$, effectively suppresses this error but introduces a new transition error between $|1\rangle \leftrightarrow |2\rangle$. Calibrating the DRAG coefficient only compromises between these errors. In contrast, with the

recursive pulse shape defined in equation (5.8), all errors are suppressed below the state preparation and measurement error.

Typically, achieving high-fidelity quantum operations requires the transition errors to be suppressed to the order of 10^{-4}. Resolving this population error often needs a large number of sampling points. Therefore, we employ the error amplification circuits outlined in [155], which add virtual Z gates $RZ(\phi)$ between the repetitions with different phases ϕ (Figure 5.3). Different transition errors will be selected by different choices of ϕ, as detailed in Section 5.8.3. To perform the measurement, we calibrate an X gate between states $|1\rangle$, $|2\rangle$ and build a measurement discriminator for qutrits [162].

In Figure 5.3e, the measured population of the state $|0\rangle(|1\rangle)$ is plotted after the initial preparation in $|0\rangle(|1\rangle)$ and 30 repetition of the CR pulse. A population close to one implies negligible errors, while any deviation indicates a transition to other states. For instance, overlapping blue and orange curves indicate the $|0\rangle \leftrightarrow |1\rangle$ transition, while a drop solely in the blue curve suggests the $|0\rangle \leftrightarrow |2\rangle$ transition. It is evident that for this short rising time ($t_r = 10$ ns), there exists a significant transition error between state $|0\rangle$ and $|1\rangle$, but also a non-negligible contribution from other transitions between $|0\rangle$ and $|2\rangle$. After applying the perturbative DRAG pulse, a substantial reduction in the error is observed, with some remaining small transitions. Using the recursive DRAG pulse derived by Givens rotation proves highly effective, suppressing all transition errors below the threshold. In both cases, no calibration of DRAG coefficients is required, and the analytical formulas are completely predictive. In general, free parameters can be added to each substitution before the derivative terms to fine-tune the strength of DRAG corrections for each individual transition error.

It is crucial to highlight that previous applications of a single-derivative DRAG [89, 106, 155] to CR gates primarily focus on the case of very small qubit-qubit detuning (ranging from 50 MHz to 70 MHz), the errors of which is dominated only by the $|0\rangle \leftrightarrow |1\rangle$ transition. However, qubit pairs on IBM Quantum Platform have detuning distributed in a much larger range from 40 to 260 MHz (Figure 5.3c), where other transitions become nonnegligible (Figure 5.3a). In contrast, the recursive DRAG solution showcased in this study exhibits remarkable universal performance even in the presence of multiple types of errors, without any calibration necessary. In Section 5.8.5, we show similar error suppression on qubit pairs with qubit-qubit deutning of 143 MHz and 189 MHz, together with an example where the single-derivative DRAG fails to suppress the error even with a full-sweep calibration of the DRAG coefficient.

5.4 Application to multi-qubit operator errors

A second major part of the error in the CR operation comes from the remaining dynamical operators in the two-qubit subspace that do not commute with the ideal dynamics ZX. Assuming the transition errors on the control qubit are all

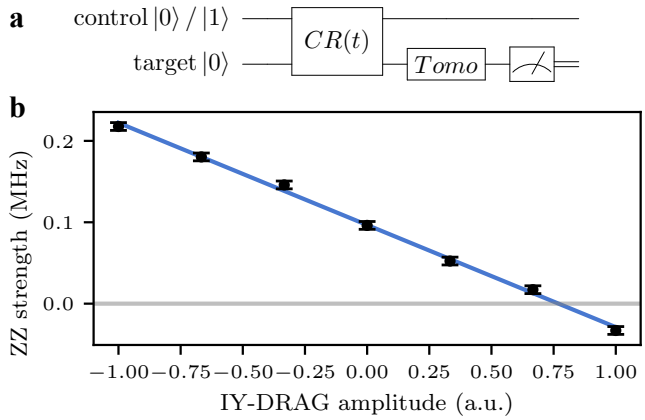


Figure 5.4: Measurement of the ZZ error and calibration of the IY-DRAG schemes. **a.** Circuit used for the CR Hamiltonian tomography [151]. **b.** Experimental data for calibrating the IY-DRAG amplitude to minimize the ZZ coupling, obtained from *ibm_perth*.

suppressed, the effective Hamiltonian in the two-qubit subspace is given by

$$H = \frac{\nu_{ZX}}{2}\hat{Z}\hat{X} + \frac{\nu_{ZY}}{2}\hat{Z}\hat{Y} + \frac{\nu_{ZZ}}{2}\hat{Z}\hat{Z} + \frac{\nu_{IX}}{2}\hat{I}\hat{X} + \frac{\nu_{IY}}{2}\hat{I}\hat{Y} + \frac{\nu_{IZ}}{2}\hat{I}\hat{Z}. \tag{5.14}$$

The coefficient $\nu(t)$ for each term can be derived by perturbative expansion, with the explicit expressions given in appendix C of Ref. [80]. Experimentally, they can be measured by Hamiltonian tomography [151]. An overview of the multi-qubit errors and the frame transformations used below to remove them is shown in Figure 5.1d.

When implementing a CR gate, the ZY term is removed by calibrating the phase of the CR drive and the single-qubit rotations, IX and IY, compensated for by a target drive [151]. To achieve high-fidelity operations, we iteratively fine-tune the drive pulse until the error terms ZY, IY and IX are all below 0.015 MHz (see Section 5.8.6). After this standard calibration procedure, one can describe the dynamics with the following effective Hamiltonian

$$\hat{H} = \frac{\nu_{ZX}}{2}\hat{Z}\hat{X} + \frac{\nu_{ZZ}}{2}\hat{Z}\hat{Z} + \frac{\nu_{IZ}}{2}\hat{I}\hat{Z}, \tag{5.15}$$

where the first term is the desired Hamiltonian dynamic while the other two are multi-qubit errors to be suppressed.

We now show that an IY-DRAG correction and a detuning are sufficient to suppress the remaining errors. Note that the two Hamiltonian terms ZX and ZZ are connected by a rotation along the IY axis. Hence, we define the transformation

$$\hat{V}_{ZZ} = \hat{I} \otimes \exp\left(-i\beta(t)\hat{Y}/2\right) \tag{5.16}$$

with $\beta = \arctan\left(\frac{\nu_{ZZ}}{\nu_{ZX}}\right) \approx \frac{\nu_{ZZ}}{\nu_{ZX}}$. In the perturbaitve expansion, the coefficients are given as $\nu_{ZZ} = \frac{J^2|\Omega_{\mathrm{CR}}|^2}{2\Delta_{\mathrm{eff}}(\alpha_1+\Delta)2}$ and $\nu_{ZX} + i\nu_{ZY} = -\frac{J\Omega_{\mathrm{CR}}\alpha_1}{\Delta(\alpha_1+\Delta)}$, where Δ_{eff} is a constant depending on the Transmons' frequency and anharmonicity [80]. This transformation results in an enhanced ZX strength $\sqrt{\nu_{ZX}^2 + \nu_{ZZ}'^2}$ and an additional single-qubit term $\dot{\beta}\hat{I}\hat{Y}/2$ to be compensated by a $\hat{Y}$ drive on the target qubit. It is not difficult to verify that this corresponds to a DRAG-like correction which is non-zero only during the pulse ramping time.

In general, to completely remove the error, one needs to match the shape of the IY-DRAG pulse exactly with $\dot{\beta}/2$. In typical CR gates, the holding period is much longer than the ramping time. Therefore, we can neglect the coherent error introduced by the time-dependent part $0 < t < t_{\mathrm{r}}$ and focus only on the holding period $t_{\mathrm{r}} < t < t_{\mathrm{f}} - t_{\mathrm{r}}$. This simplified approach allows us to neglect the shape of the IY-DRAG pulse and only calibrate the area (amplitude) such that the ZZ error is removed during the holding period. We choose the IY-DRAG shape as the first derivative of the target drive, i.e., $c_{IY}\dot{\Omega}_{IX}$. Given that the ZZ error is typically small (<0.1 MHz), the IY-DRAG correction is also very weak. Thus, the correct coefficient c_{IY} can be obtained by measuring the ZZ coupling strength for a few different c_{IY} and conducting a linear fit, as illustrated in Figure 5.4. In practice, we find that three sampling points are sufficient for the accurate calibration of the IY-DRAG amplitude. It is worth noting that in the calibration, the removed ZZ error consists of both the dynamic ones introduced by the drive and the static ZZ terms caused by residual coupling [102].

Compared to the previous approach applied by IBM, the target rotatory pulse [163], our proposed method require only three sampling points and a linear fit, employing the same tomography circuit as used in the standard calibration [151], which renders it more practical for implementation on the IBM platform with limited calibration time. In contrast, the calibration of the target rotatory pulse amplitude requires a sweep across various amplitudes and finding a minimum of total measured errors. Furthermore, our method does not require the echoed CNOT structure and thus can be used to construct a direct CNOT gate. The two methods can also be combined, introducing new degrees of freedom to suppress more residual errors at the same time, which are left for future study.

Finally, the only untreated error, the IZ term, is compensated for by detuning the CR drive. In general, the exact cancellation of the IZ error requires time-dependent detuning, i.e., a chirped pulse or phase ramping. Here, as the IZ term is usually small, it is sufficient to compensate for it with a constant detuning. This is implemented by adding an additional phase term to the pulse shape $\exp(-i\nu_{IZ}t/2)$, where ν_{IZ} denotes the measured IZ coefficient from the tomography, similar to the phase error in single qubit gates [134].

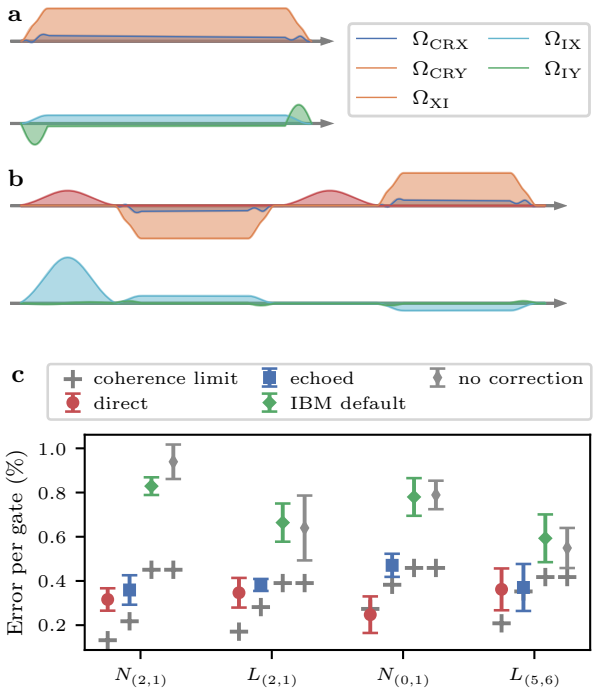


Figure 5.5: Calibrated pulses for the CNOT gate and randomized benchmarking. Examples of the custom pulse shapes used are shown for the direct (**a**) and echoed (**b**) CNOT gate on *ibm_lagos* qubit Q2 and Q1. The amplitudes for different drives are rescaled for visualization purposes. (**c**) Error per gate of the self-calibrated CNOT and the default CNOT gate, estimated by interleaved randomized benchmarking. The coherence limit is calculated including relaxation (T1) and dephasing (T2) noise provided by IBM Quantum. Data obtained from *ibm_lagos* (L) and *ibm_nairobi* (N). The error bar represents the standard deviation of five randomized benchmarking experiments.

FIG. 4. Example of the custom pulse shape used for the direct (**a**) and echoed (**b**) CNOT gate on *ibm_lagos* qubit Q2 and Q1. The amplitudes for different drives are rescaled for visualization purposes. (**c**) Error per gate of the self-calibrated CNOT and the default CNOT gate, estimated by interleaved randomized benchmarking. The coherence limit is calculated including relaxation (T1) and dephasing (T2) noise provided by IBM Quantum. Data obtained from *ibm_lagos* (L) and *ibm_nairobi* (N).

5.5 Benchmarking the improved CR gate

The investigations outlined above underscore the performance of our proposed methods in addressing both the transition errors on the control Transmon induced by rapid driving and the multi-qubit operator errors arising from (static and dynamic) residual coupling. The improved precision in control not only reduces the coherent error but also facilitates the exploration of higher drive amplitudes and faster tuning speeds, which usually introduces more coherent error if left uncompensated [81, 154]. As a result, the attained reduction in gate time allows us to exceed the impact of decoherence and achieve higher fidelities. For instance, by reducing t_r from 28 ns (default) skit-pulse parameter) to 10 ns, one gains about 35 ns for an echoed CR gate. Moreover, because of the simplified calibration, eliminating the two single-qubit gates and further reducing the gate time. Examples of the applied pulse shapes are shown in Fig. 4a and b, which include the recursive-DRAG pulse correction on the CR drive (ramping up of the CR pulse), the IY-DRAG correction (green pulse), and the frequency detuning (asymmetry between

uration, we repeat the
about 5 to 10 minutes i
tion time) and compute
of the measured gate er
error should be interpr
following hours after t
detrimental drift in the
ensuing time period.
significant reduction o
qubits, reaching the fid
default CNOT gate. V
calibrated CNOT gate
ing the default Gaussia
This comparison verifie
is not solely attributab
Detailed information o
tive coupling strength
in the Supplementary

Next, we characteriz
a wider range of typ
regimes. In particular
simulations to demonst
rent bottlenecks given
and limited calibration

To show the applica
large-scale quantum de
its performance across
resentative of a real c
randomized is chosen t
and ZZ error rate to
tum Platform (see Met
operation, we exclude
single-qubit corrections
$\Omega_{\max}$ and t_r while also
$\Delta/2\pi$={70, 110, 200} M

In Fig. 5, we compa
posed pulse and the fla
drastic reduction in the
our approach, with ord
ilarly seen for the thre
correction reduces the
transition error rate.
tion cancels the ZZ er
amplitude. Both result
coherence.

For commensurate q
liseconds, as already de
as low as 10^{-4} are with
while the standard pu
of magnitude larger er
times coming from larg
ing times, the large co
amplifies any expected

procedure, a complete removal of multi-qubit errors is achieved only when the drive is at its maximum. Therefore, the reduction in t_r contributes not only to shorter gate time but also to improved accuracy in the CR drive.

In consideration of these factors, we select pairs of qubits from the available Transmon qubits on the IBM Quantum Platform that exhibit sufficiently large coupling and relatively long coherence time ($>30\mu$s). Due to our limited access, a comprehensive search for the speed limit at different drive strengths to determine the optimal gate time is not feasible. Therefore, we opt for an empirical approach in choosing the CR drive amplitude based on the qubit-qubit detuning and the effective coupling strength. In addition to the commonly used echoed CNOT gates [151], we also calibrate direct CNOT gates, eliminating the two single-qubit gates and further reducing the gate time. Examples of the applied pulse shapes are shown in Figure 5.5a and b, which include the recursive-DRAG pulse correction on the CR drive (ramping up of the CR pulse), the IY-DRAG correction (green pulse), and the frequency detuning (asymmetry between the ramping up and off). The calibration procedure is explained in detail in Section 5.8.6.

To accurately characterize the fidelity, we measure the infidelity of the self-calibrated CR gates through interleaved randomized benchmarking. For each pulse configuration, we repeat the experiment five times (each takes about 5 to 10 minutes including the classical communication time) and compute the mean and standard deviation of the measured gate error. Therefore, the presented gate error should be interpreted as the average error over the following hours after the calibration, including possible detrimental drift in the prior system parameters over the ensuing time period. As shown in Figure 5.5, we obtain a significant reduction of the error on several pairs of the qubits compared to the default CNOT gate. Over the four pairs of qubits studied, we obtain an average gate fidelity of 99.7(1)%. We further compare it to the self-calibrated CNOT gate with no corrections applied, using the default Gaussian shape, drive amplitude and t_r. This comparison verifies that the observed improvement is not solely attributable to our more recent calibration. Detailed information on the used qubits, measured effective coupling strength and drive parameters is presented in Section 5.8.4.

Next, we characterize the possible improvement over a wider range of typical and prospective parameter regimes. In particular, we perform thorough numerical simulations to demonstrate what is possible beyond current bottlenecks given by present-day coherence times and limited calibration access on high-demand systems.

To show the applicability of the derived pulse on a large-scale quantum device, it is important to evaluate its performance across various parameter regimes representative of a real quantum system. The Hamiltonian model is chosen to have similar coupling strength and ZZ error rate to those qubits on the IBM Quantum Platform (see Methods). Since we focus on the CR operation, we exclude the error that can be removed by single-qubit corrections. We perform a sweep for different $\Omega_{\max}$ and t_r while also varying the qubit-qubit detuning $\Delta/2\pi=\{70, 110, 200\}$ MHz in the straddling regime.

In Figure 5.6, we compare the infidelity between the proposed pulse and the

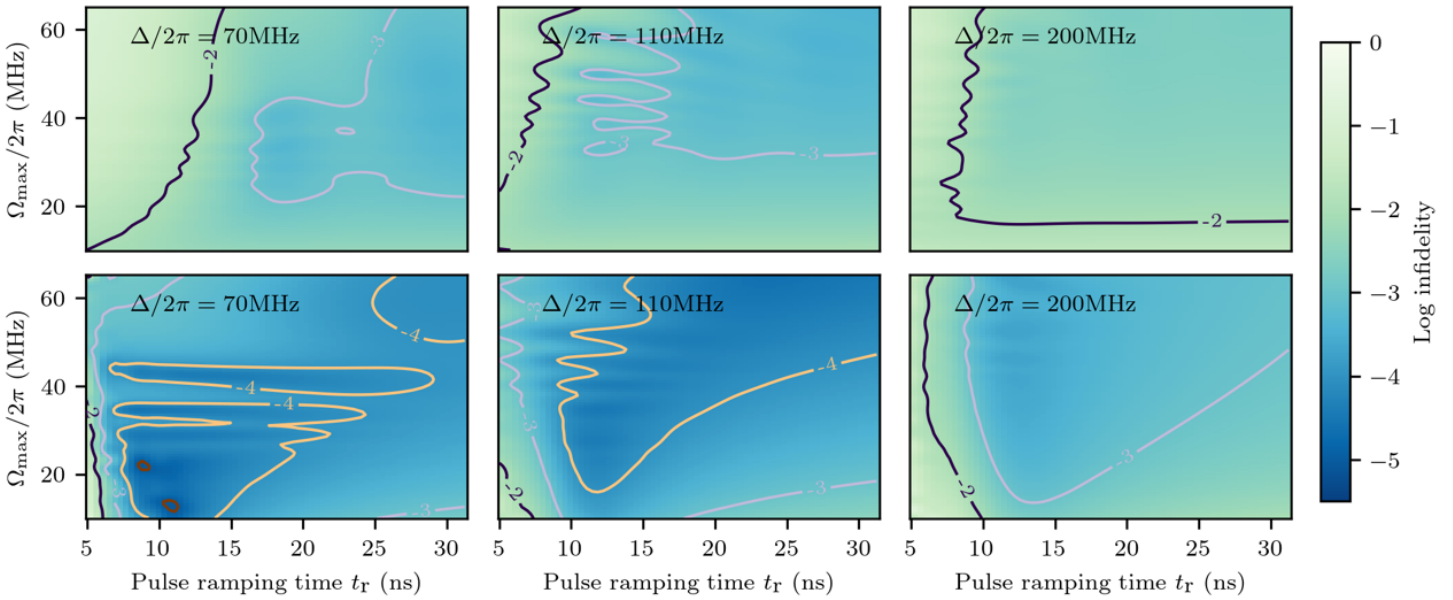


Figure 5.6: Infidelities of simulated CR gates implemented by the flat-top Gaussian pulse (top) and the proposed multi-derivative pulse (bottom), including the derived recursive DRAG correction ($\Omega_{\mathrm{CR}}^{\mathrm{G}}$) and the IY-DRAG correction. All the pulses are closed-form and computed deterministically without numerical calibration. The simulation is repeated for 3 different values of the qubit-qubit detuning Δ.

flat-top Gaussian pulse. It shows a drastic reduction in the coherent error in all regimes via our approach, with orders of magnitude suppression similarly seen for the three detunings. The recursive DRAG correction reduces the ramping time while keeping a low transition error rate. Meanwhile, the IY-DRAG correction cancels the ZZ error and allows for stronger drive amplitude. The observed optimal selection of the pulse ramping time between 10 to 15 ns in the simulations results from a compromise between the static ZZ error in IBM qubit parameters and the transition error. In our simulation, we considered IBM hardware with fixed coupler frequencies, resulting in a static ZZ error that cannot be fully corrected, especially during the ramping period. Shorter ramping times lead to reduced accumulation of this static ZZ error, at the expense of increased transition error.

For commensurate qubit lifetimes in the range of milliseconds, as already demonstrated in Refs. [164, 165], errors as low as 10^{-4} are within reach using our proposed pulse, while the standard pulse would be limited to an order of magnitude larger error. Along with the shorter gate times coming from larger amplitudes and shorter ramping times, the large coherent error suppression further amplifies any expected gains coming from improvements in qubit fabrication. This is already seen, for example, in Fig. 5.5, where not only is the coherent error on the IBM Quantum devices suppressed, but there is also a reduction in the total gate duration (with reduced coherence limit). This is especially important looking forward, as advantages in coherence times for fixed-frequency architectures vs. tunable-qubit architectures tilts the advantage towards the former with appropriate pulse shaping. Note also that even if the parasitic ZZ error is engineered to be very small [73, 83, 107, 158], as coherence times improve, the standard pulses must choose a long ramping time to match the incoherent error, while our pulse shaping approach can continue to use very short times, fully taking advantage of such improvements.

Importantly, these pulses are constructed following the analytical expression without additional optimization or fitting parameters. This means that compared to all but the simplest approaches available, including Ref. [89], these high-performance pulses are much faster and more straightforward to calibrate. Additionally, we observe that the transition error is barely affected by the drift of the drive strength and is also relatively robust against frequency drift (see Section 5.8.2).

5.6 Discussion

We introduced an analytical multi-derivative pulse shape tailored for driving the CR interaction in superconducting qubits, adept at eliminating undesired transitions on the control qubit and unwanted multi-qubit dynamics. Our approach extends the DRAG formalism to a recursive structure capable of suppressing multiple error transitions simultaneously. Additionally, we developed a novel technique to eliminate multi-operator errors by dynamically transforming the errors into the desired entangling form. This resulted in several orders of magnitude suppression in the coherent error when simulating across the range of typical c-QED regimes,

without extensive requirement on calibration. The simplicity and universality of the proposed pulse shape make it well-suited for implementation on the IBM Quantum Platform as an efficient high-quality calibration across hundreds of qubits. We demonstrate this on several qubits, showing a significant suppression in the state-of-the-art error using our customized pulse shape. The results are reproducible over a wide range of qubit frequency spacings and with prescriptive pulse shapes across the spectrum.

These analytical approaches are general and also applicable to other entangling gates in c-QED and various quantum technologies [166–168]. The control error addressed in this chapter extends beyond CR gates and is relevant to other off-resonant drive schemes, such as microwave-activated gates [112, 143, 169, 170], as well as the use of microwave drives for suppressing quantum cross-talk and leakage [67, 156, 171, 172]. The coherent error suppression demonstrated here also has implications for fixed-frequency architectures, allowing them to take advantage of longer coherence times compared to tunable architectures. Moreover, errors involving a spectator qubit [81, 156, 163, 171] can be addressed by incorporating the ancillary level into the modeling and introducing new derivative-based corrections accordingly.

Apart from the pursuit of improving multi-qubit gates, it is noteworthy that the suppression of coherent errors also indirectly enhances the fabrication process' yield. For instance, the transition errors addressed in this chapter were also identified as frequency collisions in Ref. [39, 121]. The proposed drive scheme effectively increases the threshold for frequency collisions, thereby contributing to an increased fabrication yield. Similar error models for frequency collision also apply to the tunable-coupler architecture [173], extending the potential application domain.

5.7 Methods

5.7.1 Derivation of the recursive DRAG pulse

We use the following three-level Hamiltonian to model the control Transmon

$$\hat{H} = \frac{\epsilon\Omega_{\mathrm{CR}}}{2}(\hat{\sigma}_{01}^{+} + \lambda\hat{\sigma}_{12}^{+}) + h.c. + \Delta_{10}\hat{\Pi}_1 + (\Delta_{10} + \Delta_{21})\hat{\Pi}_2, \tag{5.17}$$

where λ is the relative coupling strength of the second transition and ϵ is used to denote the perturbation order. For detuning $\Delta_{10} = 0$, the pulse is on resonance and implements a single-qubit gate. When the drive is resonant with the frequency of the target qubit, a CR operation is activated. In the rotating frame with respect to the driving frequency, we have Δ_{10} equal to the qubit-qubit detuning and $\Delta_{21} = \Delta_{10} + \alpha_{\mathrm{c}}$, with α_{c} the anharmonicity. To the leading order perturbation, the coupling strength is proportional to Ω_{CR} [80]. An ideal CR pulse generates rotations on the target qubit depending on the control qubit state while leaving the latter intact. This approximation holds well as long as the dressing of the qubit is perturbative. Therefore, we aim at finding a pulse Ω_{CR} with non-zero real integral

but introducing no population transfer among any of the three levels of the control qubit. This model, equation (5.17), includes both the leakage error and population flipping on the control qubit [154].

In the following, we show the derivation of the substitution rule of Ω in equation (5.8) via Schrieffer Wolff perturbation. We omit the perturbative corrections to the diagonal part of the Hamiltonian as they have no effect on the leading-order perturbative coupling strength. The derivation includes three steps, each targeting one coupling. The perturbative transformation generated by an anti-hermitian matrix $\hat{S}$ is defined as

$$\hat{H}' = i\dot{\hat{S}} + \hat{H} + \left[\hat{S}, \hat{H}\right] + \frac{1}{2}\left[\hat{S}, \left[\hat{S}, \hat{H}\right]\right] + \cdots . \tag{5.18}$$

First, we apply the perturbative diagonalization targeting the $|0\rangle \leftrightarrow |1\rangle$ transition

$$\hat{S}_1 = \frac{\epsilon}{2}\left(\frac{\Omega_1}{\Delta_{10}}\hat{\sigma}_{01}^{+} + \frac{\lambda\Omega_1}{\Delta_{10}}\hat{\sigma}_{12}^{+}\right) - h.c. \tag{5.19}$$

The first component in $\hat{S}_1$ is chosen to remove the $|0\rangle \leftrightarrow |1\rangle$ coupling perturbatively. According to the derivation in the main text, we define a substitution for $\Omega_{\mathrm{CR}}^{\mathrm{P}}$

$$\Omega_{\mathrm{CR}}^{\mathrm{P}} = \Omega_1 - i\frac{\dot{\Omega}_1}{\Delta_{10}}. \tag{5.20}$$

The second term in equation (5.19) is chosen such that $i\dot{\hat{S}}_1$ is proportional to the Y control Hamiltonian. This ensures that in the derived effective Hamiltonian, no $\dot{\Omega}_1$ appears in the $|1\rangle \leftrightarrow |2\rangle$ coupling, because it is absorbed in Ω_{CR}. Note that it does not diagonalize the $|1\rangle \leftrightarrow |2\rangle$ coupling, which would need $\frac{\lambda\Omega_1}{\Delta_{21}}\hat{\sigma}_{12}^{+}$ instead. As a result, we obtain

$$\begin{aligned}\hat{H}_1 = &\left(1 - \frac{\Delta_{21}}{\Delta_{10}}\right)\left(\frac{1}{2}\lambda\Omega_1\epsilon\hat{\sigma}_{12}^{+} - \frac{\lambda\Omega_1^2\epsilon^2}{8\Delta_{10}}\hat{\sigma}_{02}^{+}\right) + h.c. \\ &+ \mathrm{diag} + \mathcal{O}(\epsilon^3).\end{aligned} \tag{5.21}$$

In the second step, we perform another perturbative diagonalization that removes the $|1\rangle \leftrightarrow |2\rangle$ transition:

$$S_2 = \left(1 - \frac{\Delta_{21}}{\Delta_{10}}\right)\frac{\lambda\Omega_2\epsilon}{2\Delta_{21}}\hat{\sigma}_{12}^{+} - h.c. \tag{5.22}$$

and substitute

$$\Omega_1 = \Omega_2 - i\frac{\dot{\Omega}_2}{\Delta_{21}}. \tag{5.23}$$

This gives the effective Hamiltonian

$$\begin{aligned}\hat{H}_2 = &\left(\frac{\Delta_{21}}{\Delta_{10}} - 1\right)\left(\Omega_2 - i\frac{\dot{\Omega}_2}{\Delta_{21}}\right)^2\frac{\lambda\epsilon^2}{8\Delta_{10}}\hat{\sigma}_{02}^{+} + h.c. \\ &+ \mathrm{diag} + \mathcal{O}(\epsilon^3)\end{aligned} \tag{5.24}$$

where both single-photon transitions are removed to the leading order. Notice that the DRAG pulse shape is independent of the relative drive amplitude λ in this first-order approximation.

It may seem strange that the remaining coupling for the $|0\rangle \leftrightarrow |2\rangle$ transition is not symmetric with respect to the order of the transformations of $|0\rangle \leftrightarrow |1\rangle$ and $|1\rangle \leftrightarrow |2\rangle$, although the two substitutions commute. In fact, we can perform a transformation $\hat{S}_3$

$$\hat{S}_3 = -\left(\frac{1}{\Delta_{21}} - \frac{1}{\Delta_{10}}\right)\frac{\left(\lambda\Omega_2^2\epsilon^2\right)}{8\Delta_{10}}\hat{\sigma}_{02}^{+} - h.c. \tag{5.25}$$

which only removes the $\Omega\dot{\Omega}$ term and gives

$$H_3 = \frac{1}{8}\lambda\epsilon^2\left(\frac{1}{\Delta_{21}} - \frac{1}{\Delta_{10}}\right)\left(\frac{\dot{\Omega}_2^2}{\Delta_{21}\Delta_{10}} + \Omega_2^2\right)\hat{\sigma}_{02}^{+} + h.c. + \text{diag} + \mathcal{O}(\epsilon^3). \tag{5.26}$$

Lastly, we perform the third step to suppress the remaining $|0\rangle \leftrightarrow |2\rangle$ coupling. To fully remove this transition one needs to solve the differential equation

$$\left(\frac{\dot{\Omega}_2^2}{\Delta_{21}\Delta_{10}} + \Omega_2^2\right) = \left(\frac{\dot{\Omega}_3^2}{\Delta_{21}\Delta_{10}} + \Omega_3^2\right) - i\frac{\mathrm{d}}{\mathrm{d}t}\frac{\left(\frac{\dot{\Omega}_3^2}{\Delta_{21}\Delta_{10}} + \Omega_3^2\right)}{\Delta_{20}}, \tag{5.27}$$

which is difficult because of the non-linearity. Moreover, it may result in a pulse that does not fulfil the boundary condition, unless Ω_3 is carefully chosen to ensure that. In practice, numerical solutions may be employed to solve the equation, though it will pose challenges for fast calibration. For simplicity, we here assume that the pulse ramping is quasi-adiabatic i.e. $\Omega_2^2 \gg \frac{\dot{\Omega}_2^2}{\Delta_{10}\Delta_{21}}$. For the parameters studied in this work, with $\dot{\Omega} \approx \frac{\Omega}{t_r}$, this threshold lies around $t_r \approx 6$ ns. In this case, we can ignore the term proportional to $\dot{\Omega}_2^2$. We then define the last transformation that diagonalizes the $|0\rangle \leftrightarrow |2\rangle$ transition

$$\hat{S}_4 = \frac{1}{8}\lambda\epsilon^2\left(\frac{1}{\Delta_{21}} - \frac{1}{\Delta_{10}}\right)\frac{\Omega_2^2}{\Delta_{20}}\hat{\sigma}_{02}^{+} - h.c. \tag{5.28}$$

and substitute

$$\Omega_2 = \sqrt{\Omega_3^2 - i\frac{2\Omega_3\dot{\Omega}_3}{\Delta_{20}}}. \tag{5.29}$$

Here, $\Omega_3 = \Omega^{(3)}$, defined in equation (5.13), which is a continuously three-times differentiable function and ensures that the final expression starts and ends at zero. As a result, we suppress all three transitions up to $\mathcal{O}(\epsilon^3) + \mathcal{O}(\dot{\Omega}^2/\Delta^4)$.

Combining the three expressions, we obtain the explicit formula for the perturbative recursive DRAG pulse presented in the main text. As simple as the perturbative DRAG expression is, it may not sufficiently suppress the error if the qubits frequencies are very close to the one of $|0\rangle \leftrightarrow |1\rangle$ or $|1\rangle \leftrightarrow |2\rangle$ and the

perturbative approximation no longer holds, as shown in Figure 5.3b. To address this limitation, we replace the substitutions for the single-photon transitions with the exact diagonalization based on Givens rotations defined in equation (5.7). It is important to note that the substitution $\mathcal{F}^{(1),G}$ is only exact concerning the two-level subsystem; corrections to the energy gaps and other couplings are still disregarded. Nevertheless, it still significantly improves the performance compared to the perturbative expressions.

5.7.2 Numerical simulation of the CR gate

In the simulation, we use an effective Duffing model [117] truncated at 4 levels

$$\hat{H}_0 = \omega_a \hat{a}^\dagger \hat{a} + \sum_{j=1,2} \omega_j \hat{b}_j^\dagger \hat{b}_j + \frac{\alpha_j}{2} \hat{b}_j^\dagger \hat{b}_j^\dagger \hat{b}_j \hat{b}_j + g_j (\hat{b}_j \hat{a}^\dagger + \hat{b}_j^\dagger \hat{a}), \tag{5.30}$$

where $\hat{b}_j$ and $\hat{a}$ are the annihilation operators for qubit j and the resonator, respectively, and g_j is the coupling strength. The microwave drive on qubit j is written as

$$\hat{H}_c = \mathrm{Re}(\Omega_{\mathrm{CR}}) \cos(\omega_d t)(\hat{b}_j^\dagger + \hat{b}_j) + i\, \mathrm{Im}(\Omega_{\mathrm{CR}}) \cos(\omega_d t)(\hat{b}_j^\dagger - \hat{b}_j), \tag{5.31}$$

where ω_d is the driving frequency, initially chosen as the frequency of the target qubit. For simplicity, we use the same drive frequency for both the control and the target qubit.

We choose the anharmonicity $\alpha = -300$ MHz and $g_j = 80$ MHz. The detuning of the coupler from the control qubit, i.e., $\omega_1 - \omega_r$, is about -1.4 GHz and adjusted such that the effective qubit-qubit coupling strength is about 3 MHz, with ZZ crosstalk around 0.06 MHz, similar to the Transmons on the IBM platform (see Section 5.8.4). Based on the model above, we derive the CR pulse following the analytical expressions derived in this chapter. The effective coupling strength of ZX and ZZ are computed using the Non-Perturbative Analytical Diagonalization (NPAD) method [73], from which the gate time, i.e., the holding duration of the pulse, is calculated.

When computing the fidelity in the simulation, we ignore the contribution of the (commuting) single-qubit corrections $\hat{Z}\hat{I}$ and IX, because they can be easily calibrated in the experiment. Given an ideal unitary $\hat{\mathcal{U}}_{\mathrm{I}}$ for a two-qubit gate, the average gate fidelity is defined as [174]

$$F[\hat{\mathcal{U}}_{\mathrm{Q}}] = \frac{\mathrm{Tr}\left[\hat{\mathcal{U}}_{\mathrm{Q}} \hat{\mathcal{U}}_{\mathrm{Q}}^\dagger\right]}{d(d+1)} + \frac{\left|\mathrm{Tr}\left[\hat{\mathcal{U}}_{\mathrm{Q}} \hat{\mathcal{U}}_{\mathrm{I}}^\dagger\right]\right|^2}{d(d+1)}, \tag{5.32}$$

where $\hat{\mathcal{U}}_{\mathrm{Q}}$ is the full unitary truncated to the two-qubit subspace and $d = 4$. Because we ignore the possible single-qubit correction $\hat{Z}\hat{I}$ and IX, we compute the maximal fidelity optimized over the possible single-qubit rotation angles

$$\tilde{F} = \max_{\{\theta_1, \theta_2\}} F\left[\mathrm{e}^{-i(\theta_1 \hat{I}\hat{X} + \theta_2 \hat{Z}\hat{I})} \hat{\mathcal{U}}_{\mathrm{Q}} \mathrm{e}^{i(\theta_1 \hat{I}\hat{X} + \theta_2 \hat{Z}\hat{I})}\right]. \tag{5.33}$$

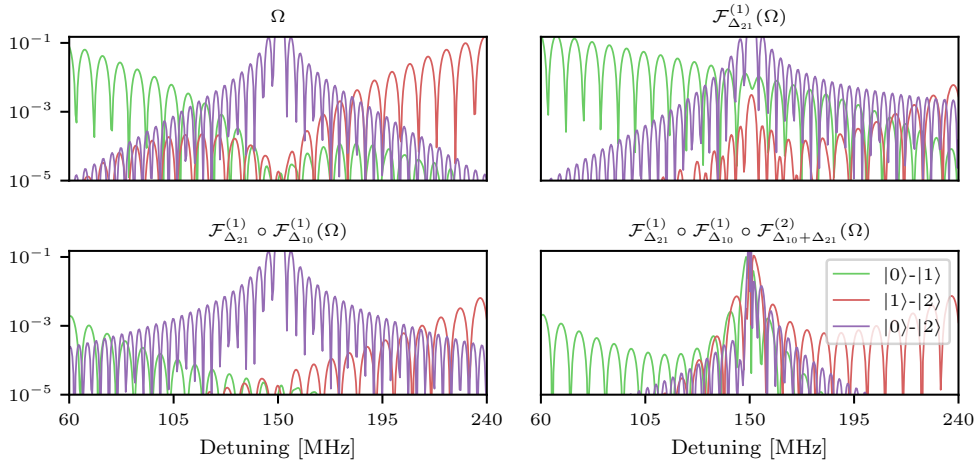


Figure 5.7: Transition probabilities using different perturbative pulse substitutions.

5.8 Appendix

5.8.1 Partial suppression of transition errors

In Figure 5.7 and Figure 5.8, we compare the impact of partial suppression of certain errors on the overall performance. We plot the transition probabilities among the three levels, using pulses designed to suppress none, one, two and all three of the transitions. The plots illustrate the corresponding suppression of different transitions through the prescribed pulse substitutions. Additionally, it is evident that a solution targeting only partial suppression inadvertently increases other transitions due to the newly introduced high-frequency components, which underlines the importance of the simultaneous suppression of all transitions.

5.8.2 Robustness of the recursive pulse

Superconducting qubits often suffer from the drift of the qubit frequency and the drive strength. In the following, we investigate the performance of the derived analytical pulse shape against those drifts. For simplicity, we assume that the drift is constant during the CR drive. We derive the pulse shape using $\Omega_{\max}$ and the control qubit frequency Δ_1 and then perform the two-Transmon simulation with parameter deviations: $\Omega_{\max} + \epsilon_\Omega$ and $\Delta_1 + \epsilon_\Delta$.

The total error transition probability is computed from the unitary evolution and depicted on Figure 5.9. Although the drift of the drive strength causes some oscillations, it does not significantly increase the error. A qualitative explanation can be found in the two-level derivation: since the X and Y amplitudes drift simultaneously, the suppression remains the same in the first-order perturbation. Only in the next order does it come into the picture through the correction to

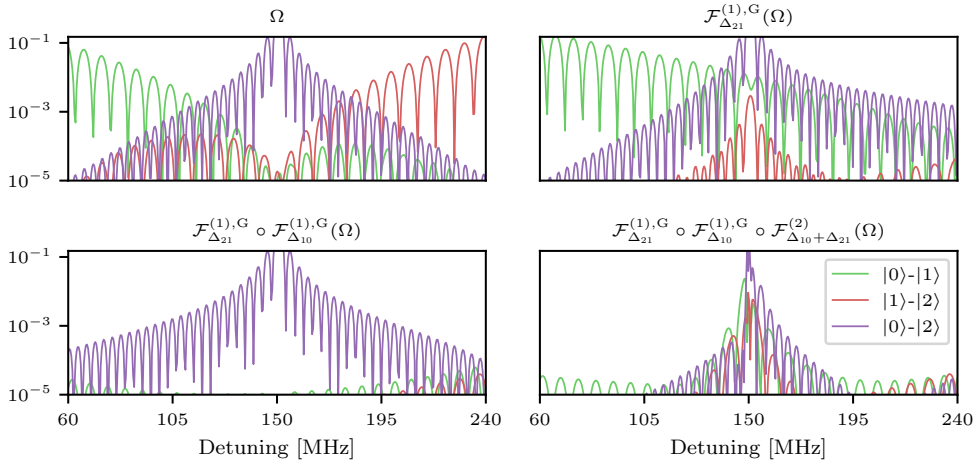


Figure 5.8: Transition probabilities using the substitutions derived from the exact two-level diagonalization for the single-photon transitions.

the energy gap via Stark shift. The transition error is increased by one order of magnitude if the frequency drifts about 10% with respect to the qubit-qubit detuning. Note that in practice drifts are usually much smaller, in the kHz regime. Not surprisingly, the analytical pulse shape is not located at the region with the absolute lowest error. Therefore, the performance will benefit from further calibration, both in simulation and experiment. In fact, in the experiment here, we nonetheless do not calibrate these parameters, to demonstrate the remarkable in situ precision of the out-of-the-box pulses.

5.8.3 Amplifying the transition error

Here we present a simplified derivation of the error amplification technique for the off-resonant error, as discussed in [155]. In particular, we demonstrate its applicability to multi-photon transitions, such as the $|0\rangle \leftrightarrow |2\rangle$ transition.

For simplicity, we restrict our analysis to a two-level subsystem in which the transition occurs, assuming that other error transitions are not amplified simultaneously. The Hamiltonian is given as

$$\hat{H} = \begin{pmatrix} -\frac{\Delta}{2} & \frac{g}{2} \\ \frac{g}{2} & \frac{\Delta}{2} \end{pmatrix}. \tag{5.34}$$

The time evolution for a duration t, given by $\hat{\mathcal{U}} = \exp\left(-i\hat{H}t\right)$, yields:

$$\hat{\mathcal{U}} = \begin{pmatrix} \cos\left(\frac{t\Delta'}{2}\right) + \frac{i\Delta \sin\left(\frac{t\Delta'}{2}\right)}{\Delta'} & -\frac{ig \sin\left(\frac{t\Delta'}{2}\right)}{\Delta'} \\ -\frac{ig \sin\left(\frac{t\Delta'}{2}\right)}{\Delta'} & \cos\left(\frac{t\Delta'}{2}\right) - \frac{i\Delta \sin\left(\frac{t\Delta'}{2}\right)}{\Delta'} \end{pmatrix} \tag{5.35}$$

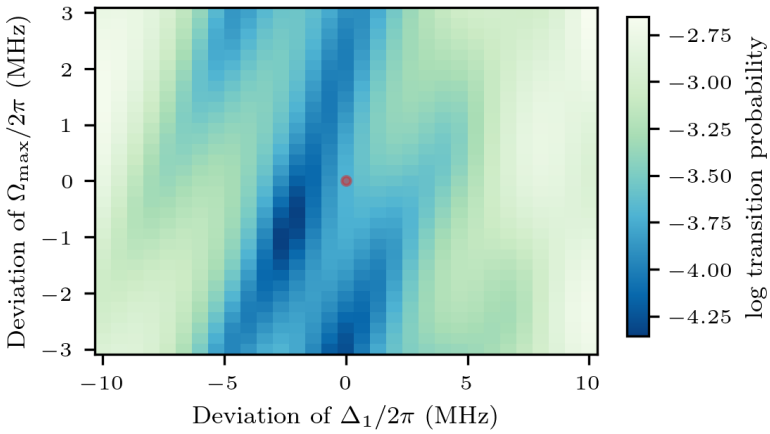


Figure 5.9: The total error transition probabilities for a precalculated pulse shape under the effect of parameter drift. The red dot marks the data point that uses the initial parameters: $\Omega_{\max}/2\pi = 40$ MHz, $\alpha/2\pi = -300$ MHz, $\Delta/2\pi = 110$ MHz and $t_r = 10$ ns.

where $\Delta' = \sqrt{(\Delta^2 + \Omega^2)}$. It clearly shows in the above equation that by prolonging the evolution time t, the population error is upper-bounded by $\frac{g^2 \sin^2(t\Delta'/2)}{\Delta'^2}$.

According to Ref. [155], the transition error can be amplified by introducing a virtual phase gate $\mathrm{RZ}(\phi)$ between the two levels. The angle ϕ is selected to ensure that $\mathrm{RZ}(\phi)\hat{\mathcal{U}}$ induces rotation solely around a fixed axis on the equator, with no rotation around the Z-axis. Solving the equation under the approximation $\Omega \ll \Delta$ yields $\phi = \Delta' t$.

In practice, determining the required angle ϕ is not straightforward, requiring a sweep over all possible angles to identify the resonant one. If the transition is between states $|0\rangle$ and $|1\rangle$, the virtual phase is commonly implemented by shifting the phase of the drive [175]. Here, we show that this approach remains valid when the coupling g couples other states such as $|1\rangle$ and $|2\rangle$ or is a multi-photon process like $|0\rangle$ and $|2\rangle$.

To this purpose, we replace g by $\mathrm{e}^{in\phi}\frac{\Omega^n}{\Delta_{\mathrm{eff}}^{n-1}}$ and write the corresponding Hamiltonian

$$H_\Omega(\theta) = \begin{pmatrix} -\frac{\Delta}{2} & \frac{1}{2}\mathrm{e}^{in\theta}\frac{\Omega^n}{\Delta_{\mathrm{eff}}^{n-1}} \\ \frac{1}{2}\mathrm{e}^{-in\theta}\frac{\Omega^n}{\Delta_{\mathrm{eff}}^{n-1}} & \frac{\Delta}{2} \end{pmatrix} \tag{5.36}$$

where Ω is the drive amplitude with a phase θ. The corresponding unitary evolution is denoted by $\hat{\mathcal{U}}_\Omega(\theta) = \mathrm{e}^{-iH_\Omega(\theta)t}$. It is then straight forward to show that $\mathrm{RZ}(\phi)\hat{\mathcal{U}}_\Omega(\theta) = \hat{\mathcal{U}}_\Omega(\theta - \phi/n)\mathrm{RZ}(\phi)$. This shows that the virtual phase gate of angle ϕ for this two-level subsystem can also be implemented by phase shifting the drive Ω by $-\phi/n$. Therefore, both the single photon transitions $|0\rangle \leftrightarrow |1\rangle$, $|1\rangle \leftrightarrow |2\rangle$ and

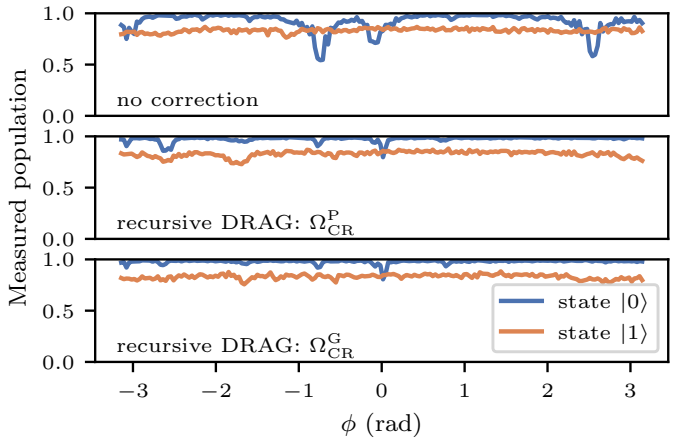


Figure 5.10: The transition error under different pulse schemes for a pair of qubits with $\Delta_{10} = 143$ MHz. The data is obtained from *ibm_nairobi* Q1 and Q3 with the drive amplitude is 0.3 (≈ 51 MHz), $t_r = 15$ ns and $N = 60$.

the two-photon transition $|0\rangle \leftrightarrow |2\rangle$ can be amplified by adding virtual phase gate with different ϕ. Moreover, sweeping ϕ from 0 to 2π is expected to reveal two peaks for the two-photon transition. Although Ref. [155] suggests preparing the target qubit in the $|+\rangle$ state for a better understanding of the amplification dynamics, we omit it here as we focus solely on the dynamics of the control Transmon.

5.8.4 Data of the used Transmon qubits

In Table 5.1 and Table 5.2, we provide the data of qubits and the parameters used for the CR drive. The coherence time, qubit frequency and the default drive amplitude J were obtained from the IBM Quantum Platform [141] on 17th October 2023. The numbers vary from day to day but in a reasonable range. The effective coupling strength J and the idling ZZ strength were also obtained from the default calibration data.

For the default CR gate, the effective Hamiltonian terms and the default gate time are measured during our calibration procedure. For the self-calibrated CR gate, the drive amplitude is chosen empirically. Due to our limited access, we are unable to sweep through different drive amplitudes and optimize for the optimal gate time. Instead, the drive strength and t_r are chosen empirically based on our knowledge of the qubit-qubit detuning and the saturation of the effective ZX coupling. Further improvements are expected through a more comprehensive and in-depth calibration.

Data of the default CR gate							
qubit pairs	N (0, 1)	N (2, 1)	N (1, 3)	L (5, 6)	L (3, 1)	L (2, 1)	L (5, 4)
control qubit T1 (μs)	109	127	122	158	117	108	158
target qubit T1 (μs)	122	122	121	103	62	62	79
control qubit T2 (μs)	39	141	82	89	96	93	89
target qubit T2 (μs)	82	82	58	80	95	95	30
qubit-qubit detuning Δ (MHz)	91	104	143	112	-113	88	-109
qubit-qubit coupling J (MHz)	2.42	3.3	3.25	3.24	3.21	3.31	3.4
default drive amplitude (a.u.)	0.345	0.127	28.5	0.495	0.74	0.254	0.5
estimated amplitude (MHz)	70.2	19	46.6	48.8	56.7	28.7	49.3
t_r (ns)	28.4	28.4	28.4	28.4	28.4	28.4	28.4
ZZ (MHz)	0.037	0.071	0.076	0.069	0.0.67	0.069	0.076
ZX CR default (MHz)	-1.89	-0.94	-1.67	-1.85	-1.39	-1.48	-1.21
default gate duration (ns)	249	391	270	256	299	291	327

Table 5.1: Data of the default CR gate

Data of the self-calibrated CR gate				
qubit pairs	N (0, 1)	L (2, 1)	N (2, 1)	L (5, 6)
drive amplitude (a.u.)	0.345	0.5	0.4	0.6
estimated amplitude (MHz)	70.2	56.5	59.8	59.1
t_r (ns)	10	10	10	13
ZX strength (MHz)	-1.89	-2.48	-2.43	-2.12
direct CNOT duration (ns)	146	114	121	135
echoed CNOT duration (ns)	227	199	206	220

Table 5.2: Data of the self-calibrated CR gate

5.8.5 Additional data on the transition error suppression

In the following, we show additional data on the validation of the recursive DRAG pulse for suppressing the transition errors on the control qubit. We compare it to the single-derivative DRAG pulse used in previous experiments and discuss the calibration of multiple DRAG parameters.

In addition to the qubit pair with the qubit-qubit detuning 104 MHz shown in Figure 5.3, we show two pairs of qubits with detuning 143 MHz and 189 MHz in Figures 5.10 and 5.11. In both cases, the recursive DRAG demonstrates excellent performance without any further calibration. Because the single-photon transition error is not very large, there is little difference between the perturbative solution and the Gives rotation in those two cases.

In Figure 5.3b, we compare the performance of the single-derivative DRAG and the proposed recursive DRAG methods through simulation. In Figure 5.12, we show an example where, despite the calibration of the DRAG coefficient, the single-derivative DRAG fails to sufficiently suppress all errors, whereas the proposed methods exhibit excellent performance. We plot the amplified error for different DRAG coefficients with the single-derivative DRAG scheme with a free parameter a_{01}. Evidently, while the $|0\rangle \leftrightarrow |1\rangle$ transition can be sufficiently suppressed with a properly chosen DRAG coefficient, other errors, such as the $|0\rangle \leftrightarrow |2\rangle$ transitions, remain largely unaffected. In contrast, both recursive methods show substantial improvement, with the pulses derived by Givens rotation achieving a perfect suppression up to the resolution of our amplification circuits, consistent with the performance observed in the qubit pairs illustrated in Figure 5.3e.

Although the recursive DRAG pulse needs little calibration for this problem, in some scenarios, especially only perturbative DRAG is used, calibration of the DRAG parameter may still prove useful. This can be achieved by adding a free parameter before each correction term in the substitution formula. In Figure 5.11, we replace the substitution for the $|0\rangle \leftrightarrow |2\rangle$ transition in equation (5.12) to

$$\Omega_2 = \sqrt{\Omega_3^2 - ia_{02}\frac{2\Omega_3\dot{\Omega}_3}{\Delta_{20}}}. \tag{5.37}$$

By varying the free parameter a_{02}, the $|0\rangle \leftrightarrow |2\rangle$ transition can be fine-tuned.

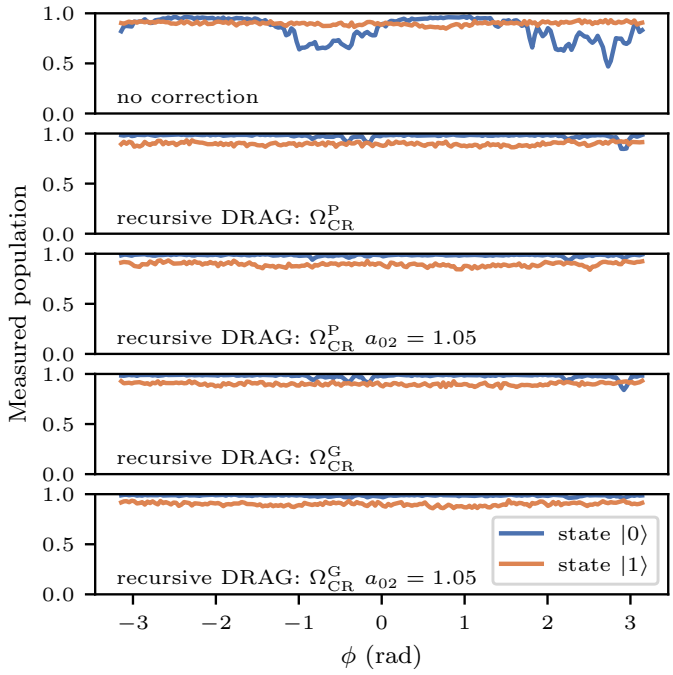


Figure 5.11: The transition error under different pulse schemes for a pair of qubits with $\Delta_{10} = 189$ MHz. The data is obtained from *ibm_lagos* Q5 and Q3 with the drive amplitude is 0.5 (≈ 49 MHz), $t_r = 15$ ns and $N = 30$.

Thanks to its recursive structure, the suppression of other transitions remains unaffected.

This independence between different parameters is illustrated more clearly in the simulation result in Figure 5.13. Here we also modify the perturbative DRAG substitution for the $|0\rangle \leftrightarrow |1\rangle$ transition to

$$\Omega_{CR}^{P} = \Omega_1 - ia_{01}\frac{\dot{\Omega}_1}{\Delta_{10}}. \tag{5.38}$$

We simulate the dynamics of the three-level Hamiltonian introduced in the main text. By sweeping the DRAG parameters a_{01} and a_02, we obtain the transition error probabilities shown in Figure 5.13. The calibration of one of the DRAG parameters has little effect on the other. Thus, the two parameters can be calibrated independently with a few iterations without a full two-dimensional sweep.

5.8.6 Calibration of the CNOT gate

In the following, we detail the calibration of the CNOT gate using CR interaction. Our calibration routine is based on the default calibration data of the Transmon

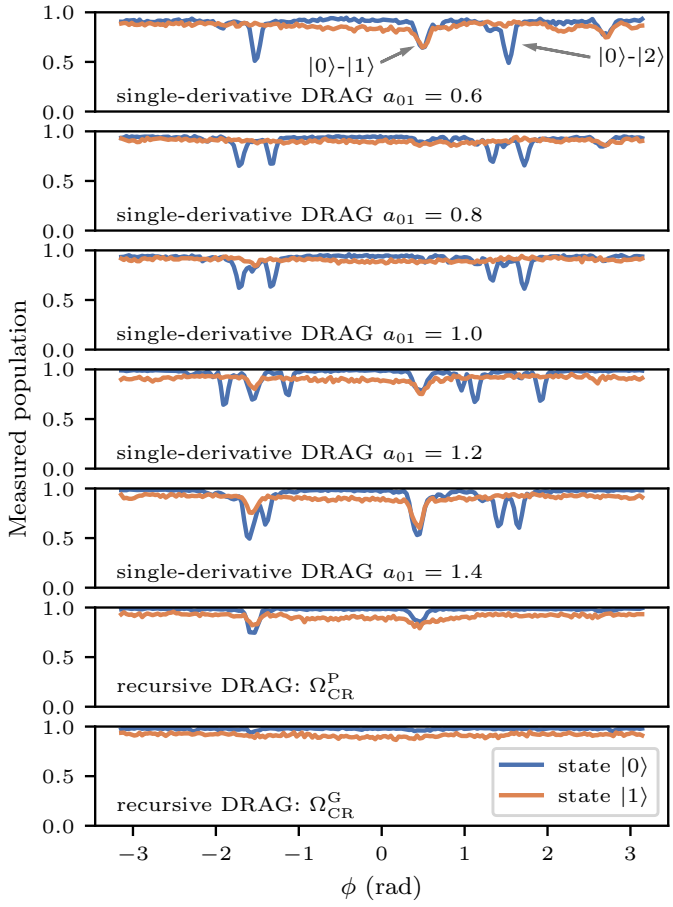


Figure 5.12: Comparison between different DRAG schemes. Plotted are the amplified transition errors of single-derivative DRAG pulses with varying DRAG coefficients, alongside the two recursive DRAG pulses proposed in this chapter. Data is obtained from *ibm_lagos* Q5 and Q6, with $\Delta = 112$ MHz, tr $= 10$ ns and a drive amplitude about 40 MHz.

frequency, anharmonicity and single-qubit X gate on the IBM Quantum Platform.

Hamiltonian tomography

We present here the Hamiltonian tomography used in calibrating the CR pulse, based on Ref. [151, 176] and the qiskit online tutorial. We start by the CR Hamiltonian in the effective frame equation (5.14). Calibrating a CR gate involves measuring the coefficients ν and eliminating undesired terms, achieved by selecting an appropriate phase for the CR drive and applying a compensatory drive on the

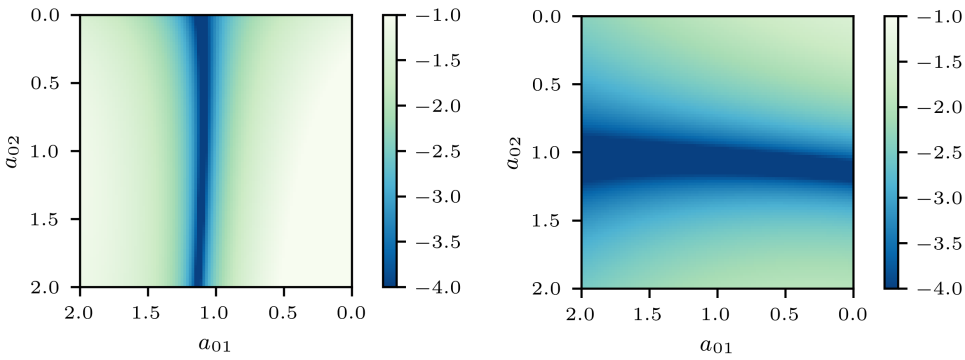


Figure 5.13: Transition error probability between $|0\rangle \leftrightarrow |1\rangle$ (left) and $|0\rangle \leftrightarrow |2\rangle$ (right) as a function of two different DRAG coefficients. The DRAG coefficient affects the targeted error but has only little effect on the other one. The parameters used in the simulation are $\alpha = -300$ MHz, $\Delta_{10} = 110$ MHz, $\Omega_{\max} = 50$ MHz and $t_r = 12$ ns.

target qubit. An example of the tomography data is shown in Figure 5.14.

To begin with, we observe that in equation (5.14), the CR dynamics involve the rotation of the target qubit, depending on the state of the control qubit. Therefore, characterization can be achieved by conducting single-qubit Hamiltonian tomography on the target qubit while preparing the control qubit in states $|0\rangle$ and $|1\rangle$.

In the following, we derive the equations that are used to fit the measured data in Figure 5.14 and obtain the coefficients ν. In the Heisenberg picture, the time evolution of an observable $\hat{O}$ is given by

$$\frac{\mathrm{d}\hat{O}}{\mathrm{d}t} = i[\hat{H}, \hat{O}]. \tag{5.39}$$

Measuring the target qubits on different bases yields the expectation values $\langle \hat{I}\hat{X} \rangle$, $\langle \hat{I}\hat{Y} \rangle$ and $\langle \hat{I}\hat{Z} \rangle$. Plugging these into equation (5.39) results in the following expressions:

$$\begin{aligned}
\frac{\mathrm{d}}{\mathrm{d}t}\hat{I}\hat{X} &= \left(\nu_{ZY}\hat{Z}\hat{Z} - \nu_{ZZ}\hat{Z}\hat{Y} + \nu_{IY}\hat{I}\hat{Z} - \nu_{IZ}\hat{I}\hat{Y}\right) \\
\frac{\mathrm{d}}{\mathrm{d}t}\hat{I}\hat{Y} &= \left(-\nu_{ZX}\hat{Z}\hat{Z} + \nu_{ZZ}\hat{Z}\hat{X} - \nu_{IX}\hat{I}\hat{Z} + \nu_{IZ}\hat{I}\hat{X}\right) \\
\frac{\mathrm{d}}{\mathrm{d}t}\hat{I}\hat{Z} &= \left(+\nu_{ZX}\hat{Z}\hat{Y} - \nu_{ZY}\hat{Z}\hat{X} + \nu_{IX}\hat{I}\hat{Y} - \nu_{IY}\hat{I}\hat{X}\right).
\end{aligned} \tag{5.40}$$

Assuming that the control qubit is prepared in a computational basis and remains unchanged during the evolution, these equations can be further simplified with the

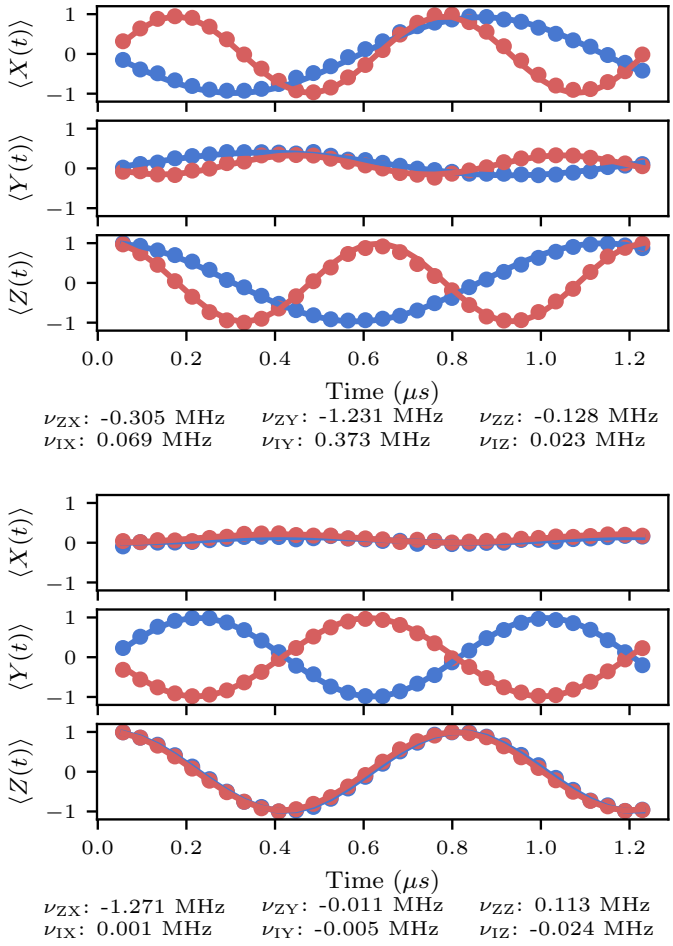


Figure 5.14: Experimental Hamiltonian tomography data of the CR pulse, before (top panel) and after (bottom panel) the calibration. The blue (red) colour corresponds to the dynamics of the target qubits under the CR drive when the control qubit is in $|0\rangle$ ($|1\rangle$).

expectation values on the target qubit:

$$\begin{aligned}
\frac{\mathrm{d}}{\mathrm{d}t}\langle\hat{X}\rangle &= (b\nu_{ZY} + \nu_{IY})\,\langle\hat{Z}\rangle - (b\nu_{ZZ} + \nu_{IZ})\,\langle\hat{Y}\rangle \\
\frac{\mathrm{d}}{\mathrm{d}t}\langle\hat{Y}\rangle &= -(b\nu_{ZX} + \nu_{IX})\,\langle\hat{Z}\rangle + (b\nu_{ZZ} + \nu_{IZ})\,\langle\hat{X}\rangle \\
\frac{\mathrm{d}}{\mathrm{d}t}\langle\hat{Z}\rangle &= (b\nu_{ZX} + \nu_{IX})\,\langle\hat{Y}\rangle - (b\nu_{ZY} + \nu_{IY})\,\langle\hat{X}\rangle,
\end{aligned} \tag{5.41}$$

where $b = 1$ ($b = -1$) if the control qubit is in state $|0\rangle$ ($|1\rangle$). To further simplify

the notation, we define $\omega_X^{(b)} = b\nu_{ZX} + \nu_{IX}$ and the same for Y and Z axis.

The differential equation can be solved by exponentiating the generator

$$G^{(b)} = \begin{pmatrix} 0 & -\omega_Z & \omega_Y \\ \omega_Z & 0 & -\omega_X \\ -\omega_Y & \omega_X & 0 \end{pmatrix} \tag{5.42}$$

where we omit the upper script (b) of ω for simplicity. Providing that the target qubit is always initialized in the ground state, the solution is as follows:

$$\begin{aligned} \langle \hat{X}(t) \rangle &= \frac{1}{\omega^2} \left(-\omega_X \omega_Z \cos(t\omega) + \omega \omega_Y \sin(t\omega) + \omega_X \omega_Z \right) \\ \langle \hat{Y}(t) \rangle &= \frac{1}{\omega^2} \left(-\omega \omega_X \sin(t\omega) - \omega_Y \omega_Z \cos(t\omega) + \omega_Y \omega_Z \right) \\ \langle \hat{Z}(t) \rangle &= \frac{1}{\omega^2} \left((\omega_X^2 + \omega_Y^2) \cos(t\omega) + \omega_Z^2 \right), \end{aligned} \tag{5.43}$$

where $\omega = \sqrt{\omega_X^2 + \omega_Y^2 + \omega_Z^2}$. These equations are then used to fit the measured data.

Although the above derivation is based on a constant drive pulse, it also applies to our time-dependent pulses in this chapter. During a tomography experiment, the time-dependent pulse ramping period is fixed while the holding time is adjusted from zero to a maximal duration. The tuning-up (-off) period of the CR drive, brings the system into (out of) the effective frame, while the tomography assesses the dynamics of the holding period. Additionally, the variation in the constant phase of the drive pulse merely affects the rotation axis of the target qubit without changing the underlying dynamics.

In practical applications, fitting trigonometric functions with undetermined oscillation frequencies can be challenging, depending heavily on the initial values. Therefore, an iterative fitting procedure is employed. The dynamics of $\langle \hat{Z}(t) \rangle$ are first fitted to obtain a good estimation of ω. Then, the other two equations are included one by one, forming an iterative fitting process. In addition, it is helpful to not force the renormalization $\omega_X^2 + \omega_Y^2 + \omega_Z^2 = \omega^2$ at the beginning. Instead, it is used to fine-tune the result in later stages, leveraging the previous values as an initial guess.

Calibration of the echoed CNOT gate

The calibration process for the echoed CNOT gate involves three main steps:

1. Adjusting the phase of the CR drive and calibrating the target compensation drive. This step ensures that the ZY, IX and IY terms are removed from the effective Hamiltonian in equation (5.14).

2. Calibrating the IY-DRAG amplitude and determining the pulse detuning. In this step, three different IY-DRAG amplitudes are sampled. The zero points of the ZZ coupling strength are determined through a linear fit (see

Figure 5.4). Simultaneously, the measured IZ coefficient provides information about the detuning of the two drives.

3. Computing the pulse duration from the tomography data. In particular, for an echoed CNOT gate, each CR pulse should be configured to last one-eighth of the period (see Figure 5.14), since the target qubit rotates towards the opposite direction depending on the state of the control. This ensures the generation of a precise 90-degree ZX rotation.

While the second step requires only a linear fit, calibration of the CR and target drive in step one cannot be completed in one round in many cases because of the nonlinearity. Therefore, we iterate a few steps until the unwanted terms are suppressed below a certain threshold. In the following, we derive the update function of one calibration step.

We first define the following notation

$$\Omega_{\mathrm{CR}} = |\Omega_{\mathrm{CR}}|\mathrm{e}^{i\theta_1} := \Omega_{\mathrm{CRX}} + i\Omega_{\mathrm{CRY}} \tag{5.44}$$

$$\Omega_{\mathrm{T}} = |\Omega_{\mathrm{T}}|\mathrm{e}^{i\theta_2} := \Omega_{IX} + i\Omega_{IY}. \tag{5.45}$$

The notation introduced provides a clear separation of amplitude and phase in the pulse design. It's important to note that the time dependence (pulse shape) is not included in this definition and is not changed during the calibration. Here, Ω represents only the maximal amplitude and a constant phase of the pulse.

The iterative calibration process begins with a predefined $|\Omega_{\mathrm{CR}}|$, with θ_1, $|\Omega_{\mathrm{T}}|$ and θ_2 all set to zero, to be updated iteratively. At each iteration k, we perform two tomography experiments. The first tomography is performed with the calibrated parameters from the previous step and measures different coupling coefficients ν of the Hamiltonian

$$\hat{H}(\Omega_{\mathrm{CR}}, \Omega_{\mathrm{T}}) = \nu_{ZX}\hat{Z}\hat{X} + \nu_{ZY}\hat{Z}\hat{Y} + \nu_{IX}\hat{I}\hat{X} + \nu_{IY}\hat{I}\hat{Y}. \tag{5.46}$$

Here we omit the ZZ and IZ terms as they are not the target in this calibration. If only the ZX term is significant and all other three terms small enough, the calibration terminates.

Given the first tomography, the phase of the CR drive for the $k+1$ iteration can be easily adjusted by

$$\theta_1^{(k+1)} = \theta_1 - \arctan\frac{\nu_{ZY}}{\nu_{ZX}} \tag{5.47}$$

where on the right-hand side we omit the upper index for step k.

To calibrate the compensation target drive, a second tomography experiment is performed with a different Ω'_{T} and results in the following Hamiltonian

$$\hat{H}(\Omega_{\mathrm{CR}}, \Omega'_{\mathrm{T}}) = \nu_{ZX}\hat{Z}\hat{X} + \nu_{ZY}\hat{Z}\hat{Y} + \nu'_{ix}\hat{I}\hat{X} + \nu'_{IY}\hat{I}\hat{Y} \tag{5.48}$$

with $\Omega'_{\mathrm{T}} = |\Omega_{\mathrm{T}} + \Delta\Omega|\mathrm{e}^{i\theta_2}$, introducing a change $\Delta\Omega$ in the drive amplitude. Note that the coefficients of the coupling terms do not change because we only

changed the target drive amplitude. This step is crucial for precisely calibrating the compensation target drive because in the qiskit user interface, the amplitude Ω_{T} is defined in a renormalized arbitrary unit from zero to one.

With the above two tomography experiments, the new amplitude $\Omega_{\mathrm{T}}^{(k+1)}$ and phase $\theta_2^{(k+1)}$ of the target drive can be computed as follows. First, the difference between the two measured Hamiltonian yields

$$\nu_{\mathrm{T}} - \nu'_{\mathrm{T}} = \left(\Omega_{\mathrm{T}} - \Omega'_{\mathrm{T}}\right) \mathrm{e}^{i\theta_2} C_{\mathrm{T}} \mathrm{e}^{-i\phi_{\mathrm{T}}} \tag{5.49}$$

where $\nu_{\mathrm{T}} = \nu_{IX} + i\nu_{IY}$ and $\nu'_{\mathrm{T}} = \nu'_{IX} + i\nu'_{IY}$. This follows from the assumption that locally the drive amplitude and the coefficients of the effective Hamiltonian show a linear relation characterized by $C_{\mathrm{T}}\mathrm{e}^{-i\phi_{\mathrm{T}}}$. Similarly, for the desired effective Hamiltonian terms with the coefficients denoted by $\nu_{\mathrm{T,ideal}}$, we have

$$\nu_{\mathrm{T,ideal}} - \nu_{\mathrm{T}} = \left(\Omega_{\mathrm{T}}^{(k+1)} \mathrm{e}^{i\theta_2^{(k+1)}} - \Omega_{\mathrm{T}} \mathrm{e}^{i\theta_2}\right) C_{\mathrm{T}} \mathrm{e}^{-i\phi_2}. \tag{5.50}$$

The solution is given by:

$$\Omega_{\mathrm{T}}^{(k+1)} \mathrm{e}^{i\theta_2^{(k+1)}} = \Omega_{\mathrm{T}} \mathrm{e}^{i\theta_2} + \frac{\nu_{\mathrm{ideal}} - \nu}{\nu - \nu'} (\Omega_{\mathrm{T}} - \Omega'_{\mathrm{T}}) \mathrm{e}^{i\theta_2}. \tag{5.51}$$

This equation provides the updated parameters for the next iteration in the calibration process. If $\theta_1^{(k+1)}$ is updated, the update $\theta_1^{(k+1)} - \theta_1^{(k)}$ must also be added to the target drive $\theta_2^{(k+1)}$.

Calibration of the direct CNOT gate

The calibration process for the direct CNOT gate is based on the echoed CNOT calibration and involves two additional steps.

First, we adjust the target drive such that $\nu_{IX} = \nu_{ZX}$. Essentially, we aim for the target qubit to rotate exclusively when the control is in the state $|1\rangle$. This can be easily implemented with the previously introduced iterative calibration process by setting $\nu_{\mathrm{ideal}} = \nu_{ZX}$ in equation (5.51). It is noteworthy that, in this case, the tomography experiment with the control qubit in state $|0\rangle$ yields minimal information and can be omitted.

Following the target drive calibration, the next step involves calibrating the phase shift on the control qubit. This phase shift is caused by the Stark shift induced by the CR drive. Unlike the echoed gate, where the phase accumulated is automatically removed by the echoing configuration, the direct gate requires explicit calibration of this phase shift. To accomplish this, we employ the circuits depicted in Figure 5.15a and b. The first circuit in Figure 5.15a applies $2N$ uncalibrated CR gate, each combined with a $RZ(\phi)$ rotation on the control. This gate sequence is sandwiched by Hadamard gates to measure the accumulated phase. The qubits return to the initial state only if the CR gate combined with the rotation gives a CNOT or a CNOT with a 180-degree rotation. To select the correct result, we use the verification circuit depicted in Figure 5.15b, which returns to the original state only for the correct CNOT gate. An example of the calibration data is shown in Figure 5.15c.

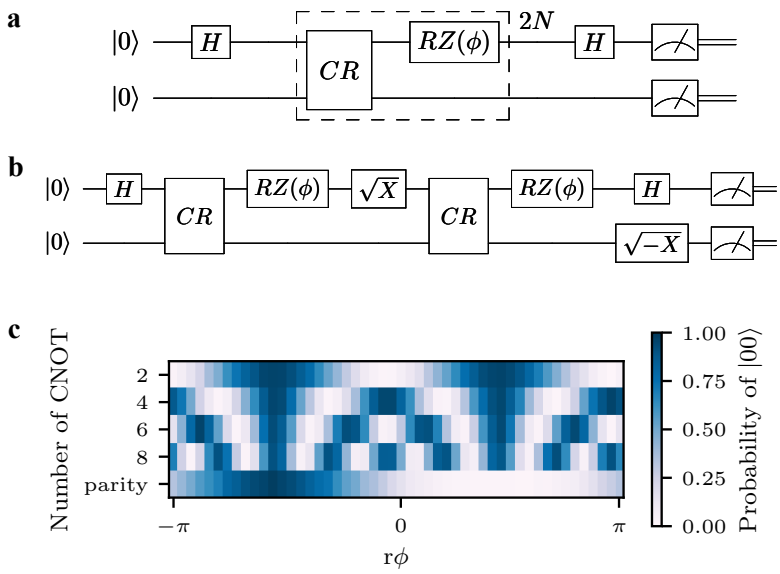


Figure 5.15: Calibration of the Z phase on the control qubit. (a) Angle calibration circuit for the Z phase. (b) Verification circuit for the Z phase. (c) Example of the calibration data. The first 4 rows correspond to the circuit a while the last row is the result of the circuit b.

6

Leakage suppression in transmon qudits with DRAG

Qudits, generalizations of qubits to multi-level quantum systems, offer enhanced computational efficiency by encoding more information per lattice cell, avoiding costly swap operations and providing even exponential speedup in some cases. Utilizing the d-level manifold, however, requires high-speed gate operations because of the stronger decoherence at higher levels. While analytical control methods have proven effective for qubits in achieving fast gates with minimal control errors, their extension to qudits is nontrivial due to the increased complexity of the energy level structure arising from additional ancillary states. In this chapter, we present a universal pulse construction for generating rapid, high-fidelity unitary rotations between adjacent qudit levels. Control errors in these operations are effectively analyzed within a four-level subspace, including two leakage levels with approximately opposite detuning. By identifying the optimal degrees of freedom, we derive concise analytical pulse schemes that suppress multiple control errors and outperform existing methods. Remarkably, our approach achieves consistent coherent error scaling across all levels, approaching the quantum speed limit independently of parameter variations between levels. Validation on transmon circuits demonstrates significant improvements in gate fidelity for various qudit sizes aiming for 10^{-4} *error.*

This chapter is part of a manuscript, coauthored with F. A. Càrdenas-López, Adrian Lupascu, and Felix Motzoi, *Universal pulses for superconducting qudit ladder gates.* [177]. The thesis author derived the DRAG expressions and the four-level effective Hamiltonians, composed most of the figures, with the exceptions of Figures 6.1d, 6.7, and 6.8, and contributed significantly to the writing of the manuscript.

6.1 Introduction

Quantum computation and quantum information processing protocols often rely on qubits, or two-level systems, as the fundamental units of computation due to their simplicity and close analogy to classical computing. However, most quantum systems comprise more than just two levels. These additional quantum levels can also be used as an information register, which is known as a qudit, a generalization of the qubit to a d-level system. Exploring the full Hilbert space of qudits enables more efficient computation by increasing the amount of information stored per quantum unit.

Qudit-based quantum computation offers several known advantages over qubit-based approaches. For instance, a d-level system can encode $\log_2(d)$ qubits [178]. This has been exploited for efficient compilation of arbitrary unitaries, requiring an exponentially reduced number of circuit layers [179–183], to simulate bosonic modes for studying light-matter processes [184, 185] and lattice gauge theories [186–188], for enhancing the robustness in quantum cryptography [189, 190], and for simplified implementation of quantum error correction protocols [191–193]. In general, the larger density of registers means that the connectivity of qubit-based architectures is increased, since neighbouring qudits can share up to d^2 level couplings. Qudit processors have been implemented across various physical platforms, including trapped-ions [194–197], Rydberg atoms [198], ultracold atomic mixtures [199], molecular spins [200, 201], photonic systems [202–206] and superconducting circuits [162, 207–213]; such implementations contribute to significant progress on qudit-based quantum computation.

Despite these advancements, maintaining coherent control of all the qudit levels poses complex challenges. In transmon superconducting circuits, where the quantum system is represented by a non-linear oscillator [41], each qudit operation needs to be addressed differently due to the varying surrounding level structure. Compared to a qubit operation, the presence of additional leakage channels significantly, limits the gate performance, as shown in Figure 6.1a-c. For instance, it has been reported that the gate time of a single-qutrit gate is around 30 ns [162, 208, 211], which is three times longer than that required for single-qubit gates with state-of-the-art quantum control techniques [134]. Therefore, developing quantum control protocols for qudits is crucial for making qudit computation practical. Of particular relevance is the Derivative Removal by Adiabatic Gate (DRAG) method [64, 66, 67, 132], successfully employed in superconducting qubit systems to reduce leakage and phase errors. DRAG's simplicity and flexibility allow engineering efficient pulses with easy-to-calibrate parameters, making it ubiquitous in the superconducting qubits platform [94, 134, 135, 143, 214]. The same advantages remain even with the presence of multiple error sources, whereby multiple DRAG corrections can be combined, offering efficient yet compact solutions [67, 136].

In this chapter, we extend the DRAG framework to engineer universally applicable and high-precision analytical control pulses for qudit systems within a ladder structure. The ladder gateset that connects levels k and $k+1$ is sufficient for universal gates within the qubit. We show that porting the widely-used

single-derivative DRAG method, as previously suggested in [66] and experimentally implemented in [215], offers little benefit in the qudit case because it is overconstrained in removing multiple leakage channels. To address this, we introduce a recursive DRAG approach that incorporates higher-order derivatives, providing new degrees of freedom that are used to suppress both single- and multi-photon errors. Our systematic study conducted on a transmon circuit across various qudit sizes demonstrates that higher-level control can be designed within a variable four-level subspace involving two nearest-neighbour interactions. Despite the presence of multiple parameters in the circuit description, we find a universal behaviour in the pulse-specific quantum speed limits. In particular, the speed limits collapse to the same times irrespective of all but one system parameter, but are strongly dependent on whether certain multi-photon transitions are suppressed. We observe significant improvements in gate performance and successfully reduce gate times to mitigate dephasing caused by voltage fluctuations during gate implementation. These results are broadly applicable to any qudit platform with multiple connected ancillary levels.

In the following, we start with the transmon model and derive the four-level effective Hamiltonian in Section 6.2. Next, we introduce and explore the recursive DRAG method in detail and perform a systematic study of its performance in Section 6.3. Significant improvement in fidelity is observed across a wide range of parameters, with a universal behaviour across all levels independent of parameter variations between levels. In Section 6.4, we discuss other potential control errors beyond the two-level transition and provide a summary of our findings in Section 6.5.

6.2 Qudit Model for universal quantum gates

6.2.1 Native gate set for superconducting qudit

Our objective is that, for a transmon system, each individual ladder transition between adjacent states, $|k\rangle$ and $|k+1\rangle$, can be selectively controlled. This allows arbitrary unitaries on $SU(d)$ to be implemented. Since a calibrated $\pi/2$ gate combined with a virtual Z gate [175] is a complete native gate set, as we show in appendix 6.6.1, our primary focus in the following study is on the $\pi/2$ gate. In addition, we also present results for the π gate, which represents the most challenging Givens rotation for a fixed gate duration due to its requirement for the strongest drive. Overall, these methods can be extended to rotations of arbitrary angles, which can be very helpful for reducing circuit compilation depths [216].

6.2.2 The transmon Hamiltonian

In this subsection, we derive the effective Hamiltonian for selectively driving the $|k\rangle \leftrightarrow |k+1\rangle$ transition in a superconducting transmon. A transmon nonlinear oscillator is described by the following Hamiltonian [41]

$$\hat{H} = 4E_C[\hat{n} - n_g(t)]^2 - E_J \cos(\hat{\varphi}) \tag{6.1}$$

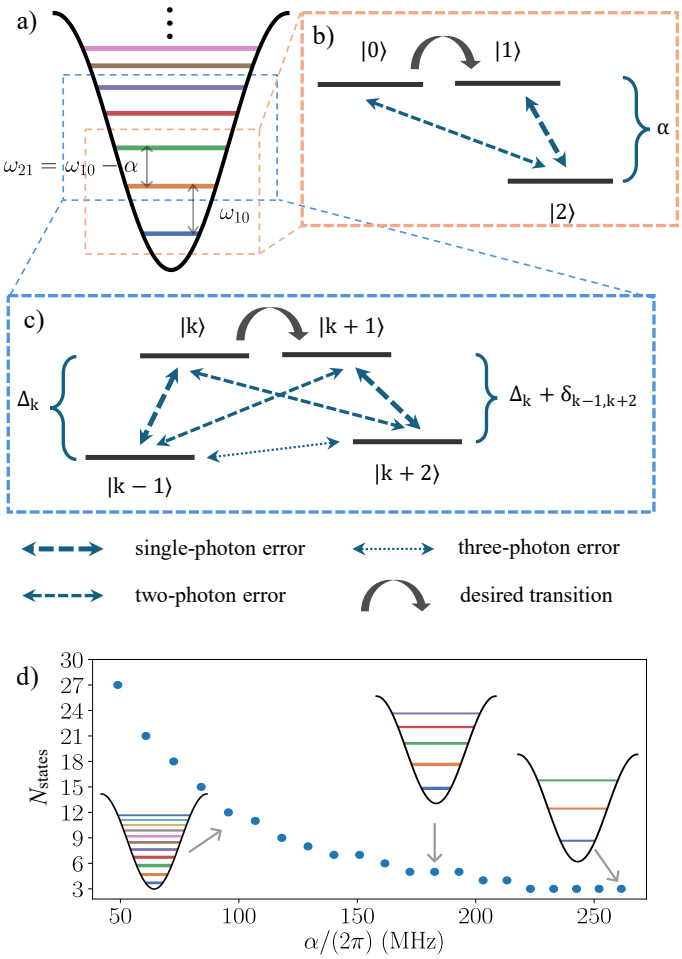


Figure 6.1: Energy structure of driving a two-level transition in a qudit system. (a) A typical energy structure of a transmon system. Energy levels and error transitions for (b) the ground and first excited states, and (c) general ladder transition between higher levels in the rotating frame. (d) The number of levels that can be used as a qudit quantum register as a function of the anharmonicity. The upper bound is set by the decoherence introduced by charge fluctuations. The detailed discussion can be found in appendix 6.6.2.

where E_C and E_J represent the charge and Josephson energies, respectively, and $n_g(t)$ is the dimensionless gate voltage. The operator $\hat{n}$ is the charge operator, indicating the number of Cooper pairs on the island, and $\hat{\varphi}$ denotes the phase operator. For implementing single-qudit operations, we capacitively drive the

transmon using $n_g(t) = n_0(t)\cos(\omega_d t)$ resulting in the Hamiltonian

$$\hat{H}_{\text{ctrl}} = \Omega(t)\cos(\omega_d t)\hat{n} \tag{6.2}$$

where $\Omega(t) = -8E_C n_0(t)$ is the drive envelope, and ω_d is the drive frequency.

When only the lowest few energy levels are considered, the transmon can be modelled as an approximate Duffing oscillator [6, 117]. In this model, the control operator is expressed as $\hat{n} \propto (\hat{b}^\dagger + \hat{b})$, where $\hat{b}$ is the annihilation operator of a linear oscillator. Consequently, the control Hamiltonian adopts a ladder configuration, connecting states $|k\rangle \leftrightarrow |k \pm 1\rangle$. The transmon Hamiltonian without external drive then simplifies to

$$\hat{H}_0^{\text{duf}} = \omega_q \hat{b}^\dagger \hat{b} + \frac{\alpha}{2}\hat{b}^\dagger \hat{b}^\dagger \hat{b}\hat{b}, \tag{6.3}$$

with $\omega_q = \sqrt{8E_J E_C} - E_C$ and $\alpha = -E_C$. As long as the states are in the potential well, the dominant coupling is still this ladder coupling between the adjacent levels, as will be shown later. However, the eigenenergies and coupling strengths deviate from the Duffing model at higher levels due to the higher-order expansion of the cosine term in equation (6.1) [117].

An accurate effective model requires exact diagonalization up to a truncation level $N_{\max}$, which gives

$$H_0 = \sum_{k=0}^{N_{\max}} \omega_k |k\rangle , \tag{6.4}$$

and for the charge operator:

$$\begin{aligned} \hat{n} \quad = \quad & \sum_{k=0}^{N_{\max-1}} \Big[n_{k,k+1} |k\rangle\langle k+1| \\ + \quad & \sum_{j=1} n_{k,k+2j+1} |k\rangle\langle k+2j+1| \Big] + \text{h.c.} \end{aligned} \tag{6.5}$$

Unlike the Duffing oscillator model, the $\hat{n}$ operator in the effective frame exhibits additional transitions, these non-zeros matrix elements are related to the underlying parity symmetry from the Mathieu functions, the formal solution of the Hamiltonian in equation (6.1). Here, we distinguish between the ladder coupling between $|k\rangle \leftrightarrow |k+1\rangle$ and high-order couplings. The latter are typically orders of magnitude smaller and are suppressed by the rotating wave approximation, as we will demonstrate later.

From this point, it is more convenient to express the Hamiltonian in the rotating frame defined by the transformation $R = \exp\left(-i\omega_d t \sum_k k |k\rangle\langle k|\right)$, where ω_d is the

selected driving transition frequency. This leads to the following total Hamiltonian:

$$\hat{H} = \sum_{k=0}^{N_{\max}} \tilde{\Delta}_k \hat{\Pi}_k + \Omega(t)\cos(\omega_d t)\Bigg[\sum_{k=0}^{N_{\max-1}} \sum_{j=1} n_{k,k+2j+1} \left|k\right\rangle\!\left\langle k+2j+1\right| e^{(2j+1)i\omega_d t} + \sum_{k=0}^{N_{\max-1}} n_{k,k+1} \left|k\right\rangle\!\left\langle k+1\right| e^{i\omega_d t} + \text{h.c}\Bigg], \tag{6.6}$$

where $\tilde{\Delta}_k = \omega_k - k\omega_d$ is the detuning between the k-th level with the k-th driving frequency harmonic. The maximal j is chosen such that $k + 2j + 1$ falls within the truncated levels. In the rotating frame, the coupling terms oscillate rapidly except for the ladder coupling between $|k\rangle \leftrightarrow |k+1\rangle$. Therefore, we can neglect these rapidly oscillating terms within the rotating wave approximation (RWA), leading to

$$\hat{H}_{\text{rwa}} = \sum_{k=0}^{N_{\max}} \tilde{\Delta}_k \hat{\Pi}_k + \frac{\Omega(t)}{2} \sum_{k=0}^{N_{\max-1}} (n_{k,k+1} \left|k\right\rangle\!\left\langle k+1\right| + \text{h.c}). \tag{6.7}$$

In this Hamiltonian, we recover the desired ladder coupling, albeit with renormalized eigenenergies and coupling strengths. To target a specific ladder transition $(k, k+1)$, the drive frequency is chosen such that $\tilde{\Delta}_k = \tilde{\Delta}_{k+1}$ for the desired k. The nonlinearity, captured by the remaining $\tilde{\Delta}_k - \tilde{\Delta}_j$ for $j \notin \{k, k+1\}$, permits selective driving of any transition between neighbouring levels.

In addition to the Hamiltonian, we also consider the possible decoherence of the higher levels. To use the quantum states as a qudit, we demand that they are robust against charge fluctuations, which increase exponentially up the ladder. This condition sets an upper bound on the maximal number of usable states N_{states}, which a priori depends on the ratio E_J/E_C, and is graphed in Figure 6.1d as a function of anharmonicity $\alpha \approx -E_C$. A higher E_J corresponds to a deeper potential, allowing for more confined states, while a lower E_C reduces charge fluctuations. However, in such a regime we also have decreases in the frequency difference between transitions, making selective driving more challenging. Instead of using the E_J/E_C ratio, it is more natural to select the usable states in terms of their coherence times T_1 and T_ϕ. In our case, as we consider fixed-frequency transmons, the main source of error will correspond to capacitive losses and dephasing due to charge fluctuations. As an optimistic forward-looking estimation, we set the upper bound of T_ϕ for the highest level N_{states} to be around 100 μs such that we are able to potentially achieve gate error below 10^{-4} for a 10 ns gate. For the parameters we choose, this corresponds to a charge dispersion of about 10^{-3} GHz. The details on the calculation of the charge fluctuation and coherence times are presented in appendix 6.6.2. In principle, for certain quantum operations, decoherence could

be partially mitigated by applying dynamical decoupling methods [217]. Here, however, we consider more generally using quantum control shaping to speed up the operation time and reduce the irreversible effect of decoherence.

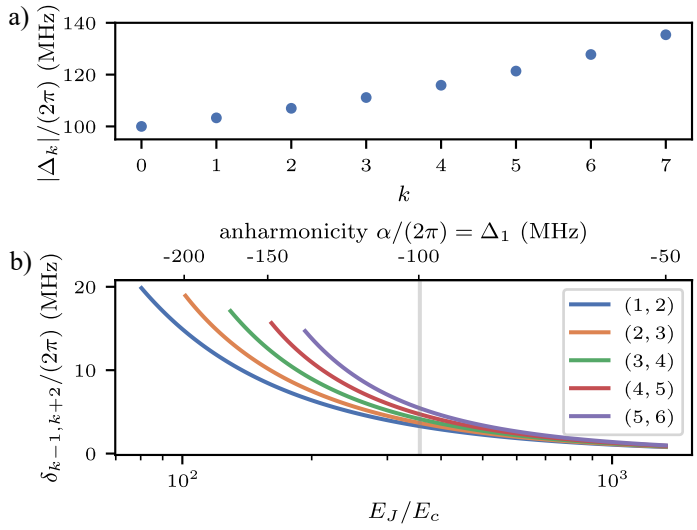


Figure 6.2: Properties of transmon qudits. (a) Detuning between the target subspace and the leakage levels in the rotating frame. The qubit frequency and anharmonicity of the ground state are 5 GHz and -100 MHz, corresponding to $E_J/E_C \approx 355$. For $k = 0$, $\Delta_k = \alpha$. (b) The small energy gap between the two leakage levels $|k-1\rangle$ and $|k+2\rangle$ for the first five transitions. This is much smaller than Δ_k, leading to the energy structure shown in Figure 6.1. The grey vertical line marks the parameters used in (a).

6.2.3 Four-level effective model

Although the full qudit has many levels, to drive a $|k\rangle \leftrightarrow |k+1\rangle$ transition many of the states are very far away detuned and thus play little role in the dynamics. Therefore, we focus on nearest-neighbour transitions and further simplify the model to a four-level system. This choice is validated by the numerical simulations that follow. We define $\omega_d = \omega_{k+1} - \omega_k - \delta_d$, with δ_d denoting a designed small detuning between the drive frequency and the energy separation. The special case for qubits, $k = 0$, has been studied over the last decade [65]. The primary control error arises from the coupling to the nearest neighbouring levels, $|k-1\rangle$ and $|k+2\rangle$, as illustrated in Figure 6.1c. To simplify the analysis, we truncate the Hamiltonian to

a four-level subsystem, described by

$$\hat{H}_k^{(4)} = \begin{pmatrix} \Delta_k & \frac{\lambda_{k-1}\bar{\Omega}_0}{2} & 0 & 0 \\ \frac{\lambda_{k-1}\Omega_0}{2} & \delta_d & \frac{\lambda_k\bar{\Omega}_0}{2} & 0 \\ 0 & \frac{\lambda_k\Omega_0}{2} & 2\delta_d & \frac{\lambda_{k+1}\bar{\Omega}_0}{2} \\ 0 & 0 & \frac{\lambda_{k+1}\Omega_0}{2} & 3\delta_d + \Delta_k + \delta_{k-1,k+2} \end{pmatrix}, \tag{6.8}$$

where $\bar{\Omega}_0$ denotes the complex conjugate of the complex pulse envelope. For clarity, a constant identity operator has been subtracted.

The two middle levels represent the targeted transition, separated by the small drive detuning δ_d. The first and last levels correspond to the potential leakage levels $|k-1\rangle$ and $|k+2\rangle$. An important observation is that, due to the weak nonlinearity, the level separation between the two leakage levels, $\Delta_k = \omega_{k-1} - 2\omega_k + \omega_{k+1}$, is approximately equal to the anharmonicity $|\alpha|$ and only increases slightly as the levels rise. This is illustrated in Figure 6.2a, with the base case $\Delta_0 = \alpha$. The difference between them is given by $\delta_{k-1,k+2} = 3\delta_d - \omega_{k-1} + 3\omega_k - 3\omega_{k+1} + \omega_{k+2}$. For a harmonic or Duffing oscillator, it is straightforward to verify that $\delta_{k-1,k+2}$ is zero. However, for a transmon oscillator, $\delta_{k-1,k+2}$ takes a small but nonzero value, as illustrated in Figure 6.2b, which is plotted as a function of E_J/E_c and the anharmonicity. The curve is truncated when the eigenstate's dispersion noise reaches 10^{-3} GHz, where the qudit coherence time drops below a minimum threshold (see appendix 6.6.2). Within this range, the $\delta_{k-1,k+2}$ is much smaller than Δ_k, as depicted in Figure 6.1c.

The off-diagonal coupling term in equation (6.8) shows a similar structure as the Duffing model. The term λ_k denotes the renormalized drive strength between level k and $k+1$, given by $\lambda_k = n_{k,k+1}/|n_{0,1}|$, which equals $\sqrt{k}$ in the Duffing approximation. Therefore, the corresponding four-level system has the structure depicted in Figure 6.1c. This model holds as long as the state remains within the potential well and the eigenenergy's dispersion to charge noise is sufficiently small.

6.3 Recursive DRAG pulse for qudit gates

For the lowest two levels in a transmon, $|0\rangle$ and $|1\rangle$, the system reduces to the well-studied single-qubit gate of the transmon qubit. The research on controlling this simple model led to the development of the widely-used DRAG technique [64, 66, 67, 132], with a particular focus on minimizing leakage to state $|2\rangle$, as illustrated in Figure 6.1b. However, beyond the first two levels, higher-level transitions present different level structure and control errors, as shown in Figure 6.1c.

Similar to the well-studied $|0\rangle \leftrightarrow |1\rangle$ transition, residual couplings between the target subspace $|k\rangle$ and $|k+1\rangle$ and the ancillary levels inevitably lead to control errors such as leakage and Stark shifting, especially when attempting to shorten gate times to reduce decoherence. An overview of the error budget is provided in Figure 6.3a, indicating the leading contributions. Moreover, to incorporate more levels into the qudit, the nonlinearity $\Delta_k \approx \alpha$ needs to be reduced to protect

the state from charge noise (Figure 6.1d), which further complicates the control scheme. To address these challenges, we introduce the recursive DRAG pulse, which accelerates gate speeds while maintaining sufficiently low control errors. In the rest of this chapter, we demonstrate how the DRAG method can be generalized for higher-level transitions and examine its performance.

6.3.1 Single-derivative DRAG and its limitation

The most widely used pulse shape is the single-derivative DRAG pulse [64]

$$\Omega - ia\frac{\dot{\Omega}}{\Delta}, \tag{6.9}$$

where a derivative term is introduced to suppress unwanted off-resonant transitions between two levels separated by Δ. In a semiclassical approximation, this approach can be interpreted as engineering a zero point in the spectrum corresponding to Δ [67]. In practice, a free parameter a is often calibrated to account for imperfect knowledge of the Hamiltonian model and higher level error contributions [64, 66]. In addition, a drive with a constant detuning parameter δ_d has been used to correct phase errors, enabling high-fidelity single-qubit gates [67, 134].

However, when driving the $|k\rangle \leftrightarrow |k+1\rangle$ transition in a qudit as described in equation (6.8), both leakage levels $|k+2\rangle$ and $|k-1\rangle$ must be considered. In this case, the single-derivative DRAG as given in equation (6.9), lacks sufficient degrees of freedom to address all sources of error [65]. For instance, if we choose a specific parameter set where $\delta_{k-1,k+2} = 0$ and $\lambda_{k-1} = \lambda_k = \lambda_{k+1} = 1$, the first derivative term provides no improvement at all, as shown in Figure 6.3b, with the leakage population to state $|j\rangle$ defined by

$$\mathcal{L}_j = \frac{1}{4}\sum_{l\in\{k,k+1\}} \left(|\mathcal{U}_{l,j}|^2 + |\mathcal{U}_{j,l}|^2\right). \tag{6.10}$$

This overconstraining occurs because the energy separations for the two leakage levels have opposite signs, i.e., $E_{|k\rangle} - E_{|k-1\rangle} \approx -(E_{|k+2\rangle} - E_{|k+1\rangle})$. This limitation applies to all ladder transitions with $k \geq 1$ in the nonlinear oscillator because $\delta_{k-1,k+2}$ is typically small compared to the anharmonicity (see Figure 6.2b).

6.3.2 General DRAG correction for a n-photon transition

As shown in Figure 6.3, a single degree of freedom is insufficient to simultaneously manage both leakage transitions. To address this limitation, an effective strategy involves incorporating higher-order derivative terms [67, 136]. This approach can be interpreted as a superadiabatic transformation [218], wherein a second adiabatic frame is derived, enabling the introduction of new time-dependent control functions proportional to the second derivative of the original drive shape. In the presence of multiple leakage levels, DRAG corrections can be tailored for each leakage coupling

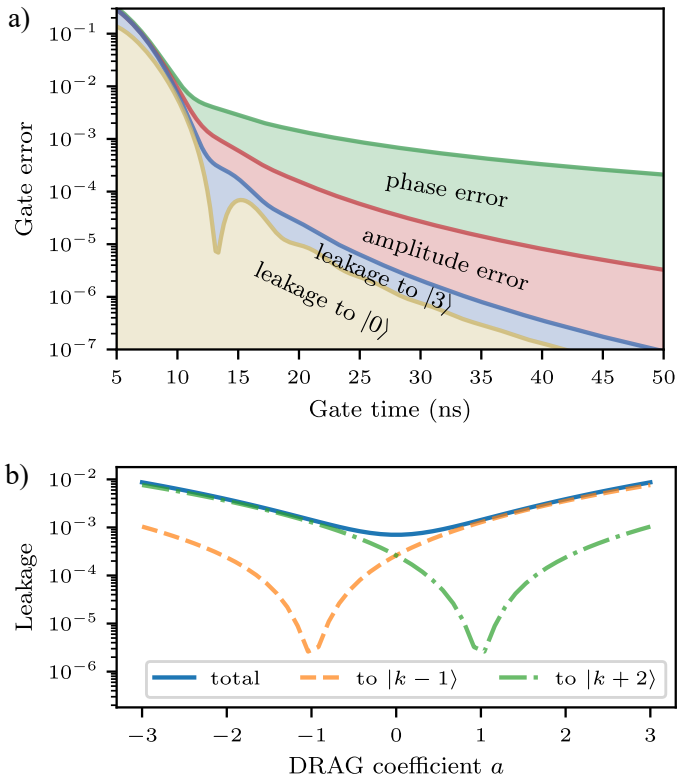


Figure 6.3: Control error in driving ladder transitions in a transmon qudit. a) Estimated error budget of driving a $|1\rangle \leftrightarrow |2\rangle$ π rotation using a Hann pulse with an anharmonicity of $\alpha/(2\pi) = -200$ MHz. The phase and amplitude errors are estimated by optimizing with constant detuning and maximal drive amplitude. Note that the error is plotted on a logarithmic scale, e.g., the two leakage errors are of the same order of magnitude. b) The leakage error calculated via Eq. 6.10 in a regime where the single-derivative DRAG [66] faces limitations and offers no improvement, even with an optimized DRAG coefficient. Parameters used are $\lambda_{k-1} = \lambda_k = \lambda_{k+1} = 1$, $\delta_{k-1,k+2} = 0$ and $t_f = 15$ns. The pulse is a single-derivative DRAG shape $\Omega - a\dot{\Omega}/\Delta_k$, with Ω the standard Hann pulse.

and chained together. In the following, we first present the general formulation and then derive the specific solution to this problem.

For an nth-order coupling $\Omega^n/\Delta_{\text{eff}}^{n-1}$ between two levels separated by Δ, the Hamiltonian in the two-level subspace is given as

$$\hat{H} = -\frac{\Delta}{2}\hat{\sigma}_z + \left(\frac{\Omega^n}{\Delta_{\text{eff}}^{n-1}}\frac{\hat{\sigma}_{jk}^+}{2} + \text{h.c.}\right). \tag{6.11}$$

Assuming $\Omega^n/\Delta_{\text{eff}}^{n-1} \ll \Delta$, we perform a perturbative expansion with the antihermitian generator $\hat{S}(\tilde{\Omega}) = \frac{\tilde{\Omega}^n}{2\Delta\Delta_{\text{eff}}^{n-1}}\hat{\sigma}_{jk}^+ - \text{h.c.}$ The time-dependent frame transformation is given as

$$\hat{H}'(\Omega) = \hat{V}(\tilde{\Omega})\hat{H}(g)\hat{V}^\dagger(\tilde{\Omega}) + i\dot{\hat{V}}(\tilde{\Omega})\hat{V}^\dagger(\tilde{\Omega}) \tag{6.12}$$

with $\hat{V}(\tilde{\Omega}) = e^{\hat{S}(\tilde{\Omega})}$. This transformation yields

$$\begin{aligned}\hat{H}'(\Omega) =& i\dot{\hat{S}}(\tilde{\Omega}) + \hat{H}(\Omega) + [\hat{S}(\tilde{\Omega}), \hat{H}(\Omega)] + \cdots \\ \approx& -\frac{\Delta}{2}\hat{\sigma}_z + \frac{1}{\Delta_{\text{eff}}^{n-1}}\left(\Omega^n - \tilde{\Omega}^n + i\frac{\mathrm{d}}{\mathrm{d}t}\frac{\tilde{\Omega}^n}{\Delta}\right)\frac{\hat{\sigma}_{jk}^+}{2} \\ &+ \text{h.c.},\end{aligned} \tag{6.13}$$

where we keep only the leading-order perturbation. Following from the equation above, the DRAG pulse is given by

$$\Omega^n = \tilde{\Omega}^n - i\frac{\mathrm{d}}{\mathrm{d}t}\frac{\tilde{\Omega}^n}{\Delta}. \tag{6.14}$$

Therefore, we can derive a drive pulse Ω resistant to this error based on the initial shape $\tilde{\Omega}$ and its derivative.

To ensure that the unitary evolution remains consistent under the frame transformation in equation (6.12), it is crucial that the generator $\hat{S}$ vanishes at the beginning and end of the evolution. To achieve this, we use the following initial pulse shape:

$$\Omega_{\text{I}}(t) = \Omega_{\max}\left[\frac{1}{16}\cos\left[6\pi\frac{t}{t_{\text{f}}}\right] - \frac{9}{16}\cos\left[\pi\frac{2t}{t_{\text{f}}}\right] + \frac{1}{2}\right], \tag{6.15}$$

with t_{f} the gate time and $\Omega_{\max}$ the drive amplitude. For comparison, we also define the widely used Hann pulse:

$$\Omega_{\text{Hann}}(t) = \sin\left[\frac{\pi t}{t_{\text{f}}}\right]^2, \tag{6.16}$$

which will be used as a baseline to benchmark the control schemes.

6.3.3 First-order (linearized) solution for qudits

To manage the two different leakage channels with opposite energy gaps as shown in Figure 6.1c and Figure 6.3b, two degrees of freedom are required. The leading-order leakage error in equation (6.8) is associated with the ladder couplings between $|k-1\rangle \leftrightarrow |k\rangle$ and $|k+1\rangle \leftrightarrow |k+2\rangle$. Both of these are first-order transitions with $n = 1$ in equation (6.11). To address the two errors, two DRAG corrections can be introduced recursively [67, 136] as

$$\Omega_0 = \Omega_1 - i\frac{\dot{\Omega}_1}{\Delta_l}, \tag{6.17}$$

$$\Omega_1 = \Omega_2 - i\frac{\dot{\Omega}_2}{\Delta_h}, \tag{6.18}$$

where $\Delta_h = E_{|k+2\rangle} - E_{|k+1\rangle}$ and $\Delta_l = E_{|k\rangle} - E_{|k-1\rangle}$ are the upper and lower adjacent levels, respectively. For Ω_2 we use Ω_I in equation (6.15). The detailed derivation based on perturbation theory is provided in appendix 6.6.4. Each of these expressions is designed to address one leakage pathway, and their order is interchangeable due to the linearity of derivatives.

This is different from the high-order perturbative solution proposed in Ref. [66], where no second derivatives were introduced and the result is only a compromise between different errors. This recursive formulation suppresses both errors simultaneously to the leading order and can be extended with additional correction terms if more ancillary levels are involved [67]. Semiclassically, it can be understood as engineering two zero points on the classical spectrum of the pulse. We refer to this DRAG pulse as the DRAG2 pulse. Notably, for a weakly nonlinear oscillator where $\Delta_l \approx -\Delta_h$, the imaginary part of the correction becomes small, and the real part dominates:

$$\Omega_0 \approx \Omega_\mathrm{I} + \frac{\ddot{\Omega}_\mathrm{I}}{\Delta_l^2} \approx \Omega_\mathrm{I} + \frac{\ddot{\Omega}_\mathrm{I}}{\Delta_h^2}. \tag{6.19}$$

Apart from the leakage error, two other errors, the phase and amplitude errors, must also be addressed to achieve the desired rotation. For a typical qubit operation between $|0\rangle \leftrightarrow |1\rangle$, the phase error comes from both the Stark shift caused by the $|2\rangle$ state and the non-commutativity of the imaginary DRAG correction term. For transitions involving higher levels, the Stark shift is influenced by both the higher and lower adjacent levels. Because the phase accumulation on the states $|k\rangle$ and $|k+1\rangle$ have the same sign, the overall accumulated phase error in this two-level transition is smaller compared to driving $|0\rangle \leftrightarrow |1\rangle$ [214]. Experimentally, this small phase error is often mitigated by applying a constant detuning to the drive [134]. The correction of the drive shape also affects the rotation angle. To compensate for this, a small correction term needs to be added, $\Omega_2 \leftarrow \Omega_2 + \Omega_\mathrm{amp}$. Similar to the phase correction, this amplitude error can also be approximately mitigated by calibrating the maximal drive amplitude Ω_max.

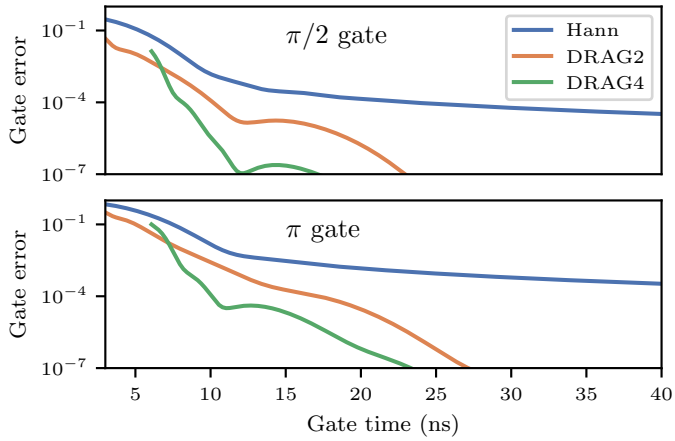


Figure 6.4: Gate infidelity as a function of duration for different drive schemes driving the $|1\rangle \leftrightarrow |2\rangle$ transition, with $\alpha/(2\pi) = -200$ MHz and $\omega_{10}/(2\pi) = 5$ GHz. The DRAG2 pulse is defined in equations (6.17) and (6.18) and the DRAG4 pulse in equations (6.20) and (6.21).

6.3.4 Second-order solution for qudit

Although the two couplings between $|k-1\rangle \leftrightarrow |k\rangle$ and $|k+1\rangle \leftrightarrow |k+2\rangle$ are suppressed by the DRAG2 correction, under a strong drive the higher-order transitions between $|k-1\rangle \leftrightarrow |k+1\rangle$ and $|k\rangle \leftrightarrow |k+2\rangle$ may also play a role. These second-order transitions arise from diagonalizing the direct ladder couplings and are proportional to Ω^2 (see appendix 6.6.4). Following the general DRAG expression in equation (6.14), this leads to the chained second-order correction:

$$\Omega_2 = \sqrt{\Omega_3^2 - i\frac{2\Omega_3\dot{\Omega}_3}{\Delta_h}} \tag{6.20}$$

$$\Omega_3 = \sqrt{\Omega_4^2 - i\frac{2\Omega_4\dot{\Omega}_4}{\Delta_l}} \tag{6.21}$$

where Ω_4 is again taken from Ω_I in equation (6.15). We refer to this pulse, combined with the two corrections in equations (6.17) and (6.18) as the DRAG4 pulse.

The second-order corrections introduced above commute with each other but do not commute with the first-derivative corrections. It is important to note that in the recursive DRAG formulation, the second-order correction is applied first to the initial pulse. This ensures that the dynamics in the final effective frame are governed by the initial pulse. This ordering is the reverse of the order of perturbation.

6.3.5 Performance benchmarking

To demonstrate the performance of the recursive DRAG pulse, we simulate the time evolution for various gate durations and compare different drive schemes. We use the analytically derived DRAG pulse while numerically calibrating a constant detuning δ_d and amplitude $\Omega_{\max}$. The simulation is performed using the full Hamiltonian, truncated at $N_{\max}$, which is much larger than the qudit size. The average gate fidelity is calculated as [174]

$$F[\hat{\mathcal{U}}_Q] = \frac{\mathrm{Tr}\left[\hat{\mathcal{U}}_Q \hat{\mathcal{U}}_Q^\dagger\right]}{d(d+1)} + \frac{\left|\mathrm{Tr}\left[\hat{\mathcal{U}}_Q \hat{\mathcal{U}}_I^\dagger\right]\right|^2}{d(d+1)}, \tag{6.22}$$

with $\hat{U}_Q$ representing the truncated unitary within the two-level subspace for the targeted transition and $\hat{\mathcal{U}}_I$ the ideal π and $\pi/2$ rotation gates. Note that this fidelity only includes deviations in the gate quality within the two-level subspace and error leakages from the target states $|k\rangle$ and $|k+1\rangle$. Error dynamics that may occur on other levels are discussed in Section 6.4.

In Figure 6.4, we compare the gate fidelity between standard Hann pulse, DRAG2 and DRAG4 pulses, for π and $\pi/2$ gates on $|1\rangle \leftrightarrow |2\rangle$. For short gate durations, below 25 ns, each successive correction improves fidelity by one to two orders of magnitude. For longer gate times, the error is primarily dominated by the phase and amplitude error, which are suppressed by the constant detuning and amplitude recalibration. The DRAG4 corrections are effective until the gate time is reduced to below 7 ns, where the ratio $\Omega_{\max}/\Delta$ approaches one and the perturbative assumption breaks. This improvement can also be examined by fixing a target fidelity and examining the minimum gate time required to achieve it. For instance, with a target fidelity of 10^{-4}, the DRAG pulse reduces the gate duration to 10 ns from 30 ns for a $\pi/2$ gate, and from more than 40 ns to 15 ns for a π gate.

Generalizing the analysis to arbitrary $|k\rangle \leftrightarrow |k+1\rangle$ transitions, we apply the same DRAG construction and repeat the benchmarking for different k values. Figure 6.5a shows the fidelity improvement for three different anharmonicities α and gate times. As the anharmonicity decreases, the system more closely resembles a linear oscillator, allowing more levels to be used as quantum registers without significant coupling to environmental noise. However, the energy difference between each level, roughly proportional to the anharmonicity, also decreases. Therefore, we increase the gate time proportionally, inversely to the reduced anharmonicity. The results indicate that the improvements provided by the DRAG corrections for general $|k\rangle$ transitions with varying anharmonicity are analogous to those observed for $|1\rangle \leftrightarrow |2\rangle$ in Figure 6.4. This also verifies that the four-level effective model is well suited for studying the transmon ladder transition across different levels. In addition, we observe that control errors decrease as the level k increases, mainly because the leakage coupling to upper and lower levels becomes more symmetric as the level goes up. As the two leakage couplings $|k-1\rangle \leftrightarrow |k\rangle$ and $|k+1\rangle \leftrightarrow |k+2\rangle$ become more symmetric, the phase error is reduced for higher levels, as also observed in Ref. [214].

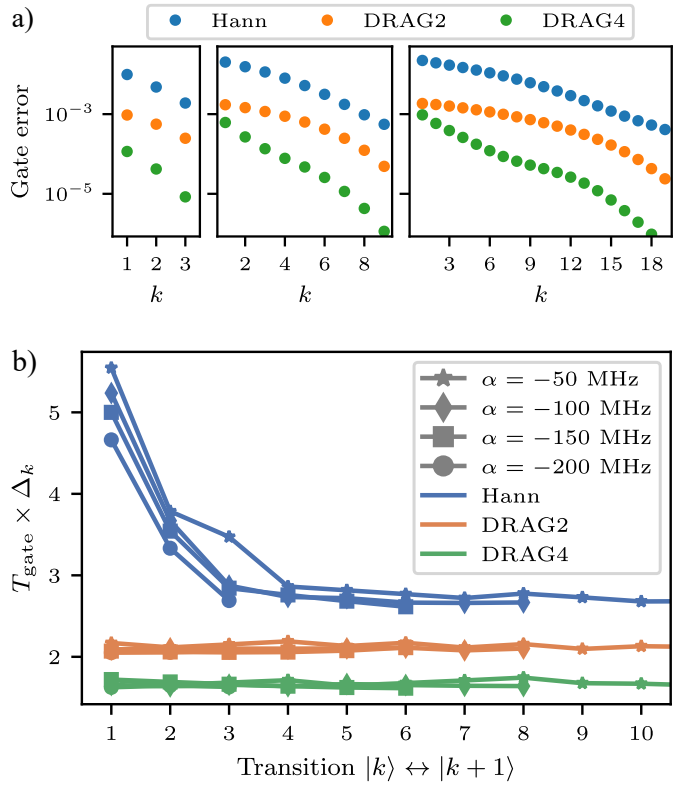


Figure 6.5: Controlling the ladder transitions in a transmon qudit. a) The $\pi/2$ gate error for different ladder transitions $|k\rangle \leftrightarrow |k+1\rangle$ for $\alpha/(2\pi) = -200, -100, -50$ MHz using a fixed gate duration of 8, 15 and 30 ns respectively. These values of anharmonicity correspond to $E_C/E_J \approx$100, 355 and 1331, respectively. b) The minimum gate time required to achieve fidelity of 10^{-4} for different drive schemes and different hardware parameters for a $\pi/2$ gate. The gate duration is multiplied by the corresponding leakage level separation Δ_k, resulting in overlapping outcomes across different hardware parameters and qudit levels.

To further characterize the control of different ladder transitions, we compute the minimal gate duration achievable for a fidelity threshold of 10^{-4}, as depicted in Figure 6.5b. To capture the universality of the pulse solutions, we normalize the time by the energy separation Δ_k for each ladder transition. Remarkably, we see that when comparing different transition indices k, and comparing different values of the anharmonicity, all the values collapse horizontally on the same line for the DRAG family of pulses. When we use DRAG4 pulses instead of DRAG2, we remove two additional weak, 2-photon transitions, and these collapse to a yet shorter minimum time line (related to a quantum speed limit for the particular choice of pulse), with apparently even stronger overlap for different parameters. However, this does not happen for the standard Hann pulse, where there is a strong dependence both on the anharmonicity (or equivalently E_J/E_C) and the chosen level index. We interpret this as evidence that removing the leakage transitions gives an effective qubit model with universal behaviour, independent of the nature of the leakage transitions themselves.

6.4 Error beyond the targeted two-level subspace

In the previous analysis, we focused on the relevant two-level subspace and computed the average gate fidelity of driving a π or $\pi/2$ rotation, using equation (6.22). For a target transition between $|k\rangle \leftrightarrow |k+1\rangle$, this error model includes leakage from the target subspace to $|k-1\rangle$ and $|k+2\rangle$ and the corresponding phase and amplitude error. To use it as the building block for universal qudit computational gates, we also need to study its effect on all the K qudit states.

6.4.1 Phase error beyond the two target levels.

As discussed above, the Stark shift accumulates phases on the affected subspace. The optimized detuning fixes the difference between the phase on $|k\rangle$ and $|k+1\rangle$. However, a phase shift still exists between the subspace and other energy levels. For the target ladder transition, this phase shift is merely a global phase, but for a K-level qudit, it becomes relevant and must be accounted for.

Fortunately, this phase mismatch can be easily calibrated by applying virtual phase gates to each untargeted level [162]. The accumulated phase is calibrated by using a phase-amplification technique. For an operation $\mathrm{RX}_{\pi/2}^{(k,k+1)}$, the state $(|k\rangle+|j\rangle)/\sqrt{2}$ is prepared, where $|j\rangle$ is the state not addressed by the gate. The gate is then applied $8n$ times, followed by a rotation of the system back using $\mathrm{RY}_{\pi/2}^{(k,k+1)}$, similar to a Ramsey experiment. The accumulated phase is then measured on the state $|j\rangle$ and corrected for future use. Virtual phase gate construction is described in Appendix 6.6.12.

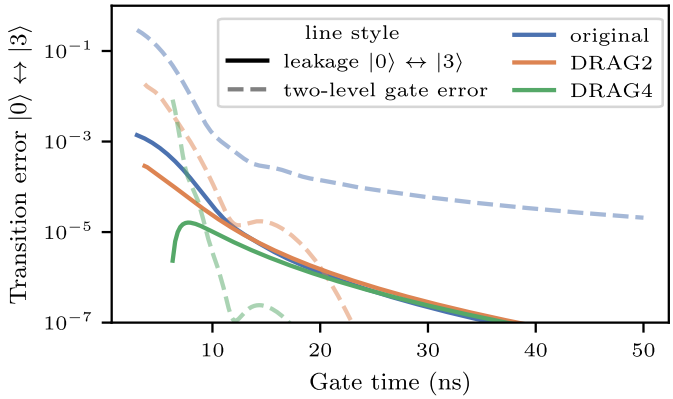


Figure 6.6: The three-photon transition error under different drive schemes. The solid line represents the $|0\rangle \leftrightarrow |3\rangle$ error as a function of the gate time for different drive schemes for a $\pi/2$ gate. The dashed lines are the same as in Figure 6.4 as a reference. The parameters used are also the same.

6.4.2 Leakage on $|k+2\rangle \leftrightarrow |k+3\rangle$

The DRAG pulse we studied primarily targets leakage involving the target subspace, i.e., leakage from the two target levels to the nearest neighbours, which are separated by approximately Δ_k in the rotating frame. Under a very strong drive, a small population transfer may also appear between nearby states such as $|k+2\rangle \leftrightarrow |k+3\rangle$, which are not directly driven. The level separation between them is about $2|\Delta_k|$. Due to this large separation, the unwanted transition is much smaller, however, it might become non-negligible ($> 10^{-4}$) if a DRAG4 pulse is used for a short gate time. Since they are small and do not involve the target states, a weak off-resonant drive can readily be added separately to cancel the small transition.

6.4.3 Three-photo leakage $|k-1\rangle \leftrightarrow |k+2\rangle$

Another small error that we have not discussed is the three-photon transition between $|k-1\rangle \leftrightarrow |k+2\rangle$. As illustrated in Figure 6.1c, this third-order transition is induced by off-resonant ladder couplings and is proportional to Ω^3. Although the absolute value of this error is accordingly small, due to the very small value of $\delta_{k-1,k+2}$ in a nonlinear oscillator, it may still introduce a non-negligible error after DRAG corrections. This error probability, defined by

$$\mathcal{L}_{k-1,k+2} = \frac{1}{4}\left(|\hat{\mathcal{U}}_{k-1,k+2}|^2 + |\hat{\mathcal{U}}_{k+2,k-1}|^2\right) \tag{6.23}$$

is plotted in Figure 6.6 for $k = 1$. Targeting a gate error of 10^{-4}, we see from the plot that this error is generally lower than the two-level gate error calculated in Section 6.3, and hence does not pose a significant obstacle.

6.5 Conclusion and discussion

Using a universal pulse construction, we have shown how coherent error can be drastically suppressed in qudit ladder systems. Moreover, they completely predict a universal behaviour whereby the minimum gate time to achieve 10^{-4} error can be very accurately calculated, irrespective of the details of the exact energy structure of the nearby surrounding levels.

The method adopts a recursive structure to simultaneously suppress multiple leakage errors. Despite the simple form, the performance benchmarking highlights the substantial error reduction achieved, enabling faster gates and consequently reducing decoherence. This method not only improves gate fidelity for the nonlinear oscillator but also offers a framework that can be adapted to other qudit systems beyond the specific model studied here.

Our results offer valuable insights into the relationship between the number of qudit levels that can be utilized for universal computation, and the corresponding gate time required to achieve a specific fidelity threshold using analytical DRAG pulses. For practical implementation with specific parameters, further optimization like frequency engineering can be performed on top of the DRAG correction [70], leading to additional performance gain.

For advanced hardware with high bandwidth waveform generators, where multiplexed frequency is possible, it would be advantageous to implement non-overlapping ladder transitions in parallel, further improving the efficiency and scalability of qudit-based quantum computing.

6.6 Appendix

6.6.1 Universality of the ladder transition

In this section, we show that any K-dimentional qudit unitary $\hat{U}$ can be decomposed into $\pi/2$ gates between adjacent levels (ladder gates) $|k\rangle$, $|k+1\rangle$ and virtual phase gates.

Decomposition of arbitrary unitary to Givens rotation

In the following, we show that arbitrary qudit unitaries $SU(d)$ can be decomposed into a sequence of Givens rotations and a diagonal phase matrix. We follow the QR decomposition similar to Ref. [219, 220] and decompose the unitary by progressively eliminating the left bottom part of the unitary matrix. An upper triangular matrix which is unitary is easily proved to be a diagonal matrix.

A unitary Givens rotation between two levels j, l is defined as

$$\hat{G}(\gamma,\varphi) = \begin{pmatrix} \ddots & & & & \\ & \cos\frac{\gamma}{2} & \dots & -ie^{i\varphi}\sin\frac{\gamma}{2} & \\ & \vdots & \ddots & \vdots & \\ & -ie^{-i\varphi}\sin\frac{\gamma}{2} & \dots & \cos\frac{\gamma}{2} & \\ & & & & \ddots \end{pmatrix}. \tag{6.24}$$

The entries not explicitly defined are filled with identities, i.e., one in the diagonal entries and zero otherwise.

The QR decomposition eliminates each column from left to right and for each row from bottom to the diagonal. This ensures that the eliminated entries will remain zero in later steps. There are in total $M = (K-1)K/2$ Givens rotations. For the m-th Givens rotation $G^{(m)}$ designed to eliminate the entry j, l, the parameters are recursively defined by

$$\tan\gamma_m = 2|U^{(m)}_{j,l}/U^{(m)}_{j-1,l}|, \tag{6.25}$$

$$\varphi_m = \pi/2 + \arg(U^{(m)}_{j-1,l}) - \arg(U^{(m)}_{j,l}), \tag{6.26}$$

where $\hat{U}^{(m)}$ is the remaining unitary after eliminating the first m entries, i.e., $\hat{U}^{(m)} = \hat{G}^{(m-1)}\hat{G}^{(m-2)}\cdots\hat{G}^{(1)}\hat{U}$.

Note that the Givens rotation applied above is always a rotation between two adjacent levels $j, j-1$, which can be directly implemented by ladder coupling. To further simplify the native gate set, it is common to use the ZXZXZ decomposition [175] designed for a qubit, which decomposes an arbitrary Givens rotation into three RZ gates and two $\pi/2$ gates

$$\hat{G}(\gamma,\varphi) = \mathrm{RZ}(-\varphi-\frac{\pi}{2})\mathrm{RX}_{\frac{\pi}{2}}\mathrm{RZ}(\pi-\gamma)\mathrm{RX}_{\frac{\pi}{2}}\mathrm{RZ}(\varphi-\frac{\pi}{2}), \tag{6.27}$$

where the subscript denotes the rotation angle. The diagonal matrix after the QR decomposition can also be easily written as a series of RZ gates. However note that the DRAG2 and DRAG4 pulses derived in the main text can also be used with any rotation and phase angle.

Virtual phase gate in a qudit

The above decomposition resolves arbitrary unitary to $\pi/2$ gates and RZ gates. The former is the main focus of the main text. Here, we show that the RZ gate between two arbitrary qudit levels, defined by

$$\mathrm{RZ}(\lambda) = \begin{pmatrix} e^{-i\lambda/2} & 0 \\ 0 & e^{i\lambda/2} \end{pmatrix}, \tag{6.28}$$

can be implemented by adjusting a constant phase of the drive [175]. This virtual phase implementation will significantly reduce the duration of the circuit. In contrast to the case of qubits, since there are $K-1$ ladder transitions, we need to track the accumulated phase for each drive, which we denote as θ_k.

A rotation between $|k\rangle$ and $|k+1\rangle$ is implemented by a Hamiltonian

$$\hat{H}_\Omega^{(k)}(\theta) = \begin{pmatrix} 0 & \frac{1}{2}e^{i\theta_k}\Omega(t) \\ \frac{1}{2}e^{-i\theta_k}\Omega(t) & 0 \end{pmatrix}, \tag{6.29}$$

where $\Omega(t)$ is the time dependent drive amplitude and θ a constant phase of the drive. This angle θ adjusts the axis in the XY plane, around which the rotation is performed. The corresponding unitary evolution is denoted by $\hat{\mathcal{U}}_\Omega^{(k)}(\theta) = e^{-i\hat{H}_\Omega^{(k)}(\theta)t}$. It is then straightforward to show that

$$\mathrm{RZ}^{(k,k+1)}(\phi)\hat{\mathcal{U}}_\Omega^{(k)}(\theta) = \hat{\mathcal{U}}_\Omega^{(k)}(\theta-\phi)\mathrm{RZ}^{(k,k+1)}(\phi). \tag{6.30}$$

This relation indicates that by phase shifting the drive Ω by $-\phi$, the RZ phase gate can be effectively moved to the end of the gate operation. Since it appears at the end, it can be neglected, as measurements only capture state populations.

The above is enough for qubit operation. For qudit operation, however, the presence of other computational levels has to be taken into consideration. Therefore, we need to consider an RZ gate between arbitrary two levels and obtain the following expressions

$$\mathrm{RZ}^{(j,k)}(\phi)\hat{\mathcal{U}}_\Omega^{(k)}(\theta) = \hat{\mathcal{U}}_\Omega^{(k)}(\theta+\phi/2)\mathrm{RZ}^{(j,k)}(\phi), \tag{6.31}$$

$$\mathrm{RZ}^{(k,l)}(\phi)\hat{\mathcal{U}}_\Omega^{(k)}(\theta) = \hat{\mathcal{U}}_\Omega^{(k)}(\theta-\phi/2)\mathrm{RZ}^{(k,l)}(\phi), \tag{6.32}$$

$$\mathrm{RZ}^{(j,k+1)}(\phi)\hat{\mathcal{U}}_\Omega^{(k)}(\theta) = \hat{\mathcal{U}}_\Omega^{(k)}(\theta-\phi/2)\mathrm{RZ}^{(j,k+1)}(\phi), \tag{6.33}$$

$$\mathrm{RZ}^{(k+1,l)}(\phi)\hat{\mathcal{U}}_\Omega^{(k)}(\theta) = \hat{\mathcal{U}}_\Omega^{(k)}(\theta+\phi/2)\mathrm{RZ}^{(k+1,l)}(\phi), \tag{6.34}$$

with $j<k$ and $l>k+1$. Notice that the adjusted phase is reduced by half because only one of the levels overlaps between RZ and the transition gate.

6.6.2 Transmon circuit in the charge representation

In this appendix, we will discuss a more general description of the transmon circuit that goes beyond the standard Duffing oscillator [6]. The fundamental aspect of this modelling is the representation of both charge and phase operators in equation (6.1). In the charge qubit description, the operator $\hat{n}$ describes the excess of Cooper-Pair on the superconducting islands, while the cosine operator $\cos(\hat{\varphi})$ describes the tunnelling between them along the junction. Explicitly, we write:

$$\hat{n} := \sum_{n\in\mathbb{Z}} n\,|n\rangle\langle n|\,, \tag{6.35}$$

$$\exp\left(i\hat{\phi}\right) := \sum_{n\in\mathbb{Z}} |n\rangle\langle n+1|\,. \tag{6.36}$$

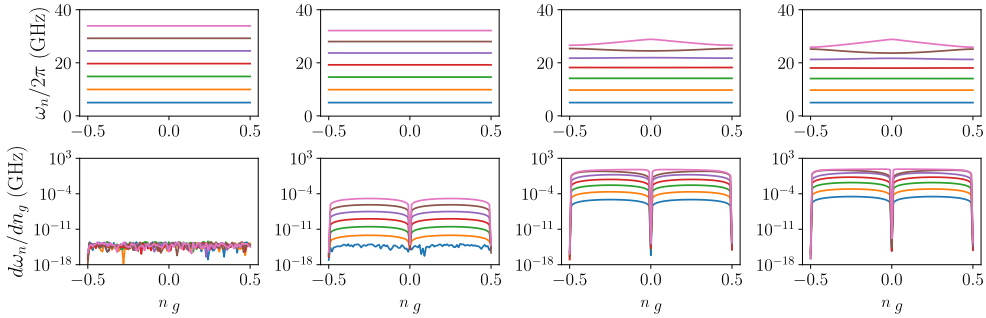


Figure 6.7: Energy spectrum of the transmon circuit as a function of the dimensionless gate voltage n_g for four different values of $\alpha/(2\pi) = \{-50, -100, -200, -300\}$ (MHz). For the numerical simulations, we have fixed E_J for obtaining $\omega_{10}/(2\pi) = 5$ (GHz), and as expected, for increasing anharmonicity, the energy spectrum becomes more sensitive to charge fluctuations.

In this representation, the eigenstates of the transmon are obtained by numerically diagonalizing the Hamiltonian to a subspace spanned by a few charge states. This modifies the control operator $\hat{n}$ in such a basis so that it exhibits selection rules different from the bosonic oscillator defining the Duffing oscillator [see equation (6.6)].

The gate voltage $n_g(t)$ responsible for driving the transitions on the transmon circuit is susceptible to fluctuations which could be thermal, due to wiring circuits and quasiparticle tunnelling through the junction, or non-thermal, due to impedance mismatching with the signal generator. Thus, we need to quantify the fluctuation of the energy levels of the transmon by varying $n_g(t)$. Figure 6.7 shows the low-lying energy spectrum as a function of the gate voltage n_g. We have selected E_C and E_J such that the $\omega_{10}/(2\pi) = (\omega_1 - \omega_0)/(2\pi) = 5$ GHz, and we vary the anharmonicity $\alpha = \omega_{21} - 2\omega_{10}$ to be in the range $\alpha/(2\pi) = (-50, -300)$ (MHz).

We observe increasing charge dispersion for larger values of α. The main reason for the increase of charge dispersion with decreasing E_J relies on always the same frequency; consequently, fewer states are confined in the cosine potential. This feature is more appreciable when we see the variation of the energy spectrum $\partial\omega_{k+1,k}/\partial n_g$ with respect to the gate voltage, where for smaller α the fluctuations are on the order of KHz.

In this scenario, depending on our transmon parameters, we need to carefully select the workable low-lying energy levels for our qudit gates. In our case, we follow a different approach than Ref. [215]; rather than compute the ratio between the deep potential with the energy spacing, we consider the average of the fluctuation over n_g. We set a truncation at $\partial\omega_{N_{\max}}/\partial n_g \approx 10^{-3}$ (GHz) and consider any eigenstates with lower dispersion suitable as a qudit level. This results in the number of levels available in the nonlinear oscillator in Figure 6.1.

This constraint on dispersion also extends to the dephasing time where we have used $1/f$ noise as the most detrimental source of decoherence which can be

estimated by the relation [41, 221]

$$\frac{1}{T_{\phi}^{(k)}} = A_{n_g} \left| \frac{\partial \omega_{k+1,k}}{\partial n_g} \right| \sqrt{2|\ln(\omega_{\text{low}} t_{\text{exp}})|}, \tag{6.37}$$

where $\omega_{k+1,k} = \omega_{k+1} - \omega_k$ and $A_{n_g} = 10^{-4}e$ is the noise strength [222–224] with e being the electron charge. Also, $\omega_{\text{low}} = 2\pi/t_{\text{exp}}$ corresponds to the infrared cutoff due to the finite data acquisition time $t_{\text{exp}} = 10^4$ ns [41]

This is illustrated in Figure 6.8, where one point corresponds to one qudit eigenstate with a specific hardware parameter. As the anharmonicity decreases, more and more levels with a coherence time longer than 100 μs can be included as quantum information registers.

For amplitude damping, we estimate T_1 assuming that the main loss mechanism corresponds to capacitive losses. In such a way, Fermi's golden rules give the relation [225]

$$\frac{1}{T_1^{(k)}} = |\langle k|\hat{n}|k+1\rangle|^2 S(\omega_{k+1,k}), \tag{6.38}$$

where the spectral density for the capacitive losses reads [221, 226]

$$S(\omega_{k+1,k}) = \frac{4\hbar E_C}{Q_{\text{cap}}(\omega_{k+1,k})} \left[\frac{\coth\left(\frac{\hbar|\omega_{k+1,k}|}{2k_B T}\right)}{1 + \exp\left(-\frac{\hbar\omega_{k+1,k}}{k_B T}\right)} \right] . \tag{6.39}$$

with $Q_{\text{cap}}(\omega_{k+1,k}) = 10^6(2\pi \times 6\ \text{GHz}/|\omega_{k+1,k}|)^{0.7}$ [227, 228] the capacitive quality factor per ladder transition. Also, k_B is the Boltzmann constant, and $T = 15$ mK is the temperature. Since this value is not strongly dependent on the levels in our system studied, we do not use it to truncate the qudit level.

For completeness, we plot the T_1 for the different energy levels in Figure 6.8. We should note that improvement of the coherent times could be possible by implementing different fabrication techniques such as surface error mitigation [229, 230], changing the Niobium with Tantalum as the base superconductor [165, 231] or mitigating the micromotion of the circuitry [232], among other techniques. Such shielding on the transmon circuit leads to coherence times nearly in the millisecond scale.

6.6.3 Derivation of the Leakage manifold

Here, we will show that the energy diagram for any qudit gate between the states $(k+1, k)$ is represented as in Figure 6.1c. In other words, if we want to implement this single qudit gate, there appears to be a nearly-resonant transition between the states $|k-1\rangle \leftrightarrow |k+2\rangle$. To do so, let us consider the explicit form of the energy of the kth energy level after the frame transformation in equation (6.7)

$$\tilde{\Delta}_k = \omega_k - k(\omega_{k+1} - \omega_k - \delta_d). \tag{6.40}$$

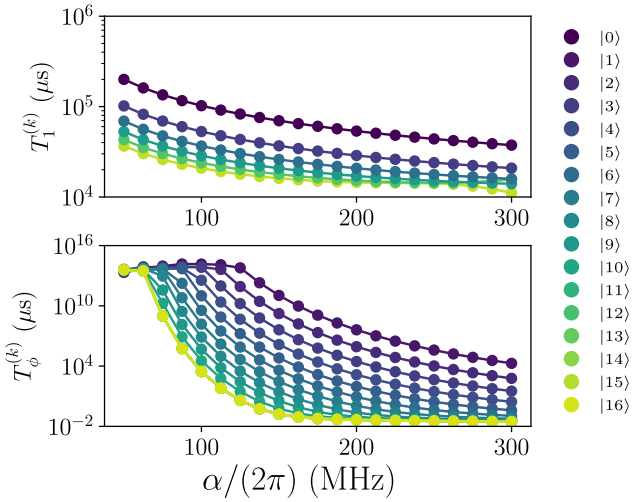


Figure 6.8: Coherence times of the transmon circuit as a function of α for different transition frequency $\omega_{k+1,k}$. We set E_J and E_C such that the transition frequency equal to $\omega_{10}/(2\pi) = 5$ (GHz). For the amplitude damping, we assume capacitive losses and dephasing correspond to charge fluctuations.

For the Dufing oscillator model, we know that $\omega_k = \omega_k - \alpha k(k-1)/2$, where $\omega = \sqrt{8E_C E_J} - E_C$ is the transmon frequency, and $\alpha = -E_C$ is the anharmonicity, respectively. Thus, $\Delta_{k-1} = (k-1)(\alpha(k+2) + 2\delta_d)/2$ while $\Delta_{k+2} = (k+2)(\alpha(k-1) + 2\delta_d)/2$. Thus, the detuning between these energy levels is $\delta_{k-1,k+2} = 3\delta_d$ for all values of k, which is zero if the drive is resonant.

However, such a description of the system Hamiltonian is only valid for larger E_J/E_C. Thus, for obtaining better estimation of the detuning, we consider the eigenenergies of the transmon obtained by numerical diagonalizing equation (6.1). Figure 6.2d shows $\delta_{k-1,k+2}$ as a function of the anharmonicity α for several ladder transitions at $n_g = 0$; from the figure we appreciate an inverse relation between the degeneracy of the leakage state with the anharmonicity, recovering the previous calculation result when $\alpha = -2\pi \times 50$ (MHz). Moreover, we also see an increase of such discrepancy with the qudit manifold to be addressed, this effect is mainly produced by the sensitivity of the energy spectrum to the charge noise (see Figure 6.7).

6.6.4 Derivation of recursive DRAG pulse

Single-photon correction

In the following, we show the derivation of the recursive DRAG pulse shape designed to suppress the two single-photon transitions $|k-1\rangle \leftrightarrow |k\rangle$ and $|k+1\rangle \leftrightarrow |k+2\rangle$. Our general approach is to progressively derive the effective frame and

the corresponding drive shapes to minimize the prevalent error. Throughout the calculation, we keep the perturbative correction up to the second order for all the terms with two exceptions: the matrix entry $(0,3)$, which characterizes a three-photon leakage due to the small energy separation, and the entry $(1,2)$, which describes the pulse amplitude correction. For those two, we keep the terms up to the third order correction.

We start with the rotating frame Hamiltonian in equation (6.8)

$$\hat{H}_0 = \begin{pmatrix} -\Delta_l & \frac{\lambda_1\bar{\Omega}_0}{2} & 0 & 0 \\ \frac{\lambda_1\Omega_0}{2} & \delta_d & \frac{\lambda_2\bar{\Omega}_0}{2} & 0 \\ 0 & \frac{\lambda_2\Omega_0}{2} & 2\delta_d & \frac{\lambda_3\bar{\Omega}_0}{2} \\ 0 & 0 & \frac{\lambda_3\Omega_0}{2} & \Delta_h + 3\delta_d \end{pmatrix}, \tag{6.41}$$

where $\Delta_h = \Delta_k + \delta_{k-1,k+2}$ and $\Delta_l = -\Delta_k$. For ease of notation, we use λ_1, λ_2, λ_3 for λ_{k-1}, λ_k and λ_{k+1} in this section. We define the first transition targeting the single-photon leakage error, $|k+1\rangle \leftrightarrow |k+2\rangle$. For small δ_d, as is typical in the transmon regime, this is the largest leakage source (see Figure 6.2). The frame transformation generator is given by

$$\hat{S}_{0\to1} = \begin{pmatrix} 0 & -\frac{\lambda_1\bar{\Omega}_1}{2\Delta_h} & 0 & 0 \\ \frac{\lambda_1\Omega_1}{2\Delta_h} & 0 & -\frac{\lambda_2\bar{\Omega}_1}{2\Delta_h} & 0 \\ 0 & \frac{\lambda_2\Omega_1}{2\Delta_h} & 0 & -\frac{\lambda_3\bar{\Omega}_1}{2\Delta_h} \\ 0 & 0 & \frac{\lambda_3\Omega_1}{2\Delta_h} & 0 \end{pmatrix}. \tag{6.42}$$

The denominator Δ_h is chosen such that in the effective frame, the matrix entry $(2,3)$ is zero. In addition, $S_{0\to1}$ is chosen to be proportional to the control term in $\hat{H}_0$; this is designed in particular such that there is no derivative term $\dot{\Omega}_1$ in $\hat{H}_1$ [136]. After substituting the expression $\Omega_0 = \Omega_1 - i\frac{\dot{\Omega}_1}{\Delta_h}$, we get $\hat{H}_1$ with the off-diagonal term

$$\hat{H}_1 - \hat{H}_{1,\mathrm{diag}} = \begin{pmatrix} 0 & \frac{1}{2}\epsilon\lambda_{r1}\bar{\Omega}_1 & \frac{-\Delta_l\epsilon^2\lambda_1\lambda_2\bar{\Omega}_1^2}{8\Delta_h^2} & \epsilon^3\bar{\Omega}^{(1)}_{\mathrm{L03}} \\ \frac{1}{2}\epsilon\lambda_{r1}\Omega_1 & 0 & \frac{\epsilon\lambda_2\left(\bar{\Omega}_1+\epsilon^2\bar{\Omega}_c^{(1)}\right)}{2} & \frac{\epsilon^2\lambda_2\lambda_3\bar{\Omega}_1^2}{8\Delta_h} \\ \frac{-\Delta_l\epsilon^2\lambda_1\lambda_2\Omega_1^2}{8\Delta_h^2} & \frac{\epsilon\lambda_2\left(\Omega_1+\epsilon^2\Omega_c^{(1)}\right)}{2} & 0 & 0 \\ \epsilon^3\Omega^{(1)}_{\mathrm{L03}} & \frac{\epsilon^2\lambda_2\lambda_3\Omega_1^2}{8\Delta_h} & 0 & 0 \end{pmatrix}, \tag{6.43}$$

where $\Omega_c^{(1)}$ and $\Omega^{(1)}_{\mathrm{L03}}$ denote the third order error to the drive amplitude in this frame and the three-photon leakage transition, which we do not explicitly use in the following calcualtion. Notice that in the effective frame, we obtain a renormalized leakage rate $\lambda_{r1}\Omega_1$ between $|k-1\rangle$ and $|k\rangle$, with $\lambda_{r1} = \lambda_1(1-\Delta_l/\Delta_h) \approx 2\lambda_1$ in the limit $\delta_{k-1,k+2} \to 0$. This explains why the leakage increases with only one

single derivative DRAG correction in Figure 6.2b. This prefactor also needs to be taken into consideration when making the perturbative assumptions. The diagonal energy terms are given by

$$E_{1,|k-1\rangle} = -\Delta_l - \frac{\lambda_1^2 \,\mathrm{Re}\left(\Omega_0\bar{\Omega}_1\right)}{2\Delta_h} + \frac{\lambda_1^2\Delta_l\left|\Omega_1\right|^2}{4\Delta_h^2}, \tag{6.44}$$

$$E_{1,|k\rangle} = \delta_d + \left(\frac{\lambda_1^2-\lambda_2^2}{2\Delta_h}\right)\mathrm{Re}\left(\Omega_0\bar{\Omega}_1\right) - \frac{\lambda_1^2\Delta_l\left|\Omega_1\right|^2}{4\Delta_h^2}, \tag{6.45}$$

$$E_{1,|k+1\rangle} = 2\delta_d + \left(\frac{\lambda_2^2-\lambda_3^2}{2\Delta_h}\right)\mathrm{Re}\left(\Omega_0\bar{\Omega}_1\right) + \frac{\lambda_3^2\left|\Omega_1\right|^2}{4\Delta_h}, \tag{6.46}$$

$$E_{1,|k+2\rangle} = 3\delta_d + \Delta_h + \frac{\lambda_3^2\,\mathrm{Re}\left(\Omega_0\bar{\Omega}_1\right)}{2\Delta_h} - \frac{\lambda_3^2\left|\Omega_1\right|^2}{4\Delta_h}. \tag{6.47}$$

Secondly, we target the single photon leakage between state $|k-1\rangle$ and $|k\rangle$, with the frame transformation generator

$$\hat{S}_{1\to2} = \begin{pmatrix} 0 & -\frac{\epsilon\lambda_{\mathrm{r}1}\bar{\Omega}_2}{2\Delta_l} & 0 & 0 \\ \frac{\epsilon\lambda_{\mathrm{r}1}\Omega_2}{2\Delta_l} & 0 & -\frac{\epsilon\lambda_2\bar{\Omega}_2}{2\Delta_l} & 0 \\ 0 & \frac{\epsilon\lambda_2\Omega_2}{2\Delta_l} & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}. \tag{6.48}$$

This, together with the substitution $\Omega_1 = \Omega_2 - i\frac{\dot{\Omega}_2}{\Delta_l}$, results in the suppression of the transition and gives $\hat{H}_2$

$$\hat{H}_2 - \hat{H}_{2,\mathrm{diag}} = \begin{pmatrix} 0 & 0 & -\frac{\Delta_l\epsilon^2\lambda_2\bar{\Omega}_1^2}{8\Delta_h^2} - \frac{\epsilon^2\lambda_2\lambda_{\mathrm{r}1}\bar{\Omega}_2^2}{8\Delta_l} & \bar{\Omega}_{\mathrm{L03}}^{(1)} \\ 0 & 0 & \frac{\epsilon\lambda_2\left(\bar{\Omega}_2+\epsilon^2\bar{\Omega}_{\mathrm{c}}^{(2)}\right)}{2} & \frac{\epsilon^2\lambda_2\lambda_3\bar{\Omega}_1^2}{8\Delta_h} \\ -\frac{\Delta_l\epsilon^2\lambda_2\Omega_1^2}{8\Delta_h^2} - \frac{\epsilon^2\lambda_2\lambda_{\mathrm{r}1}\Omega_2^2}{8\Delta_l} & \frac{\epsilon\lambda_2\left(\Omega_2+\epsilon^2\Omega_{\mathrm{c}}^{(2)}\right)}{2} & 0 & 0 \\ \Omega_{\mathrm{L03}}^{(1)} & \frac{\epsilon^2\lambda_2\lambda_3\Omega_1^2}{8\Delta_h} & 0 & 0 \end{pmatrix}. \tag{6.49}$$

In addition to the leakage error, the phase error and the amplitude renormalization also need to be considered to get the desired rotation. The time-dependent phase correction is given by

$$\begin{aligned} \delta_d = &-\mathrm{Re}\left(\Omega_1\bar{\Omega}_2\right)\left(\frac{\lambda_2^2}{\Delta_l} - \frac{\lambda_{\mathrm{r}1}^2}{2\Delta_l}\right) \\ &-\mathrm{Re}\left(\Omega_0\bar{\Omega}_1\right)\left(-\frac{\lambda_1^2}{2\Delta_h} + \frac{\lambda_2^2}{\Delta_h} - \frac{\lambda_3^2}{2\Delta_h}\right) \\ &-\left|\Omega_1\right|^2\left(\frac{\Delta_l\lambda_1^2}{4\Delta_h^2} + \frac{\lambda_3^2}{4\Delta_h}\right) - \frac{\left|\Omega_2\right|^2\lambda_{\mathrm{r}1}^2}{4\Delta_l}, \end{aligned} \tag{6.50}$$

where $\lambda_{r1} = \lambda_1(1 - \Delta_l/\Delta_h)$.

Apart from that, the correction on the drive shape also slightly affects the rotation angle. A small correction term needs to be added $\Omega_2 \leftarrow \Omega_2 + \Omega_{\text{amp}}$. The analytical formula of the amplitude correction is written as

$$\begin{aligned}
\Omega_{\text{amp}} = {} & \Omega_0 \left|\Omega_1\right|^2 \left(\frac{\lambda_1^2}{8\Delta_h^2} + \frac{\lambda_3^2}{8\Delta_h^2} - \frac{\lambda_2^2}{4\Delta_h^2}\right) \\
& + \Omega_1 \left|\Omega_2\right|^2 \left(\frac{\lambda_{\text{r1}}^2}{8\Delta_l^2} - \frac{\lambda_2^2}{4\Delta_l^2}\right) \\
& + \Omega_2 \left|\Omega_1\right|^2 \left(-\frac{\lambda_1^2}{4\Delta_h^2} - \frac{\lambda_3^2}{4\Delta_l\Delta_h}\right) \\
& + \Omega_2 \,\text{Re}\left(\Omega_0\bar{\Omega}_1\right) \left(\frac{\lambda_1^2}{2\Delta_l\Delta_h} + \frac{\lambda_3^2}{2\Delta_l\Delta_h} - \frac{\lambda_2^2}{\Delta_l\Delta_h}\right) \\
& + \Omega_1^2\bar{\Omega}_0 \left(\frac{\lambda_1^2}{8\Delta_h^2} + \frac{\lambda_3^2}{8\Delta_h^2} - \frac{\lambda_2^2}{4\Delta_h^2}\right) \\
& + \delta_d \left(-\frac{\Omega_1}{\Delta_h} - \frac{\Omega_2}{\Delta_l}\right) \\
& + \Omega_1^2\bar{\Omega}_1 \left(-\frac{\Delta_l\lambda_1^2}{8\Delta_h^3} - \frac{\lambda_3^2}{8\Delta_h^2}\right) \\
& + \Omega_2^2\bar{\Omega}_1 \left(\frac{\lambda_{\text{r1}}^2}{8\Delta_l^2} - \frac{\lambda_2^2}{4\Delta_l^2}\right) \\
& - \frac{\lambda_1\Omega_1^2\bar{\Omega}_2\lambda_{\text{r1}}}{8\Delta_h^2} - \frac{\Omega_2^2\bar{\Omega}_2\lambda_{\text{r1}}^2}{8\Delta_l^2}.
\end{aligned} \tag{6.51}$$

In our investigation, we neglect the time-dependence and numerically optimize a fixed correction of the detuning and the amplitude.

Two-photon correction

The first two transitions yield the effective Hamiltonian described in equation (6.49), where the desired transition between $|k\rangle \leftrightarrow |k+1\rangle$ is preserved, with a renormalized effective coupling strength. The diagonalization of the single-photon coupling introduces new two-photon transitions, $|k-1\rangle \leftrightarrow |k+1\rangle$ and $|k\rangle \leftrightarrow |k+2\rangle$, with the coupling strength proportional to Ω^2. This is also obtained for qubit driving in a nonlinear oscillator, as discussed in [67]. For very strong drive amplitude, these two-photon transitions become the dominant source of error once the single-photon transitions are sufficiently suppressed.

For simplicity, we do not repeat the full calculation as in the last subsection but note the following properties. We can treat Ω^2 as the new coupling g, then derive the same expression to suppress the two leakages as in equation (6.18) but

with the coupling g. Moreover, any perturbative diagonalization of the two-photon transitions will introduce corrections only in the order of ϵ^3 or smaller, which is negligible relative to the truncation order considered. By substituting g back into Ω, we obtain the expression in equation (6.21).

Three-photon correction

In principle, based on the DRAG2 pulse, we can follow a similar strategy and use a recursive DRAG design to suppress the transition error between state $|0\rangle$ and $|3\rangle$:

$$\Omega_4 = \sqrt[3]{\Omega_5^3 - i\frac{3\Omega_5^2\dot{\Omega}_5}{\delta_{k-1,k+2}}}. \tag{6.52}$$

However, due to the small gap between $|k-1\rangle$ and $|k+2\rangle$ in the transmon regime, the imaginary DRAG correction term is much larger and a constant detuning may not suffice to compensate for the phase error. Nevertheless, even with a not perfectly aligned phase, a $\pi/2$ gate can be implemented with the help of virtual phase gates.

Alternatively, one could explore the direct coupling between the states $|k-1\rangle$ and $|k+2\rangle$ instead of relying on the multi-photon process. However, this would require microwave drive generators with a frequency approximately three times that of the qubit frequency.

7 Quantum crosstalk suppression with DRAG

7.1 Introduction

As quantum devices scale up, incorporating more qubits onto a single superconducting chip, new challenges arise beyond controlling individual or pairs of qubits. Ideally, the qubits' couplings are precisely engineered and controlled, with unwanted interactions fully eliminated. However, in practice, residual couplings between qubits often persist due to fabrication imperfections, even between qubit pairs that are not intentionally connected by couplers [233]. These unintended couplings can lead to errors, and as the number of qubits increases, the likelihood of defects and unexpected behaviours rises, affecting the yield in chip fabrication. Thus, alongside improvements in fabrication processes and chip design, it is crucial to develop quantum control techniques that mitigate these errors.

In this chapter, we examine a simple model of quantum crosstalk, in which an unintended flipping interaction occurs between two qubits. Similar to classical crosstalk, which can be mitigated through compensation in the control lines [234] or pulse shaping [65], we treat the quantum crosstalk problem also as off-resonant population error. We investigate the performance of DRAG pulses in suppressing these errors during single-qubit gate operations, demonstrating their effectiveness in mitigating quantum crosstalk.

7.2 Model of quantum crosstalk

We first define a simplified model of quantum crosstalk mediated by a constant coupling strength g. The drift Hamiltonian, $\hat{H}_{\mathrm{d}}$, is given by

$$\hat{H}_{\mathrm{d}} = \Delta\hat{\Pi}_2 + \epsilon_1 g\left(\hat{\sigma}_1^+\hat{\sigma}_2^- + \hat{\sigma}_1^-\hat{\sigma}_2^+\right) = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & \Delta & g\epsilon_1 & 0 \\ 0 & g\epsilon_1 & 0 & 0 \\ 0 & 0 & 0 & \Delta \end{pmatrix}, \tag{7.1}$$

which is written in the rotating frame with respect to the first qubit. The symbol $\Delta = \omega_2 - \omega_1$ denotes the frequency difference. The control Hamiltonian is given as

$$\hat{H}_{\mathrm{c}} = \frac{\epsilon_2}{2}\left(\mathrm{Re}[\Omega_{\mathrm{c}}]\hat{X}\hat{I} + \mathrm{Im}[\Omega_{\mathrm{c}}]\hat{Y}\hat{I}\right) = \begin{pmatrix} 0 & 0 & \frac{1}{2}\bar{\Omega}_{\mathrm{c}}\epsilon_2 & 0 \\ 0 & 0 & 0 & \frac{1}{2}\bar{\Omega}_{\mathrm{c}}\epsilon_2 \\ \frac{\Omega_{\mathrm{c}}\epsilon_2}{2} & 0 & 0 & 0 \\ 0 & \frac{\Omega_{\mathrm{c}}\epsilon_2}{2} & 0 & 0 \end{pmatrix}, \tag{7.2}$$

where Ω_{c} is the time-dependent complex control function.

In this chapter, we continue to use the notation as described in Section 4.2 for the Hamiltonians and transformation generators (see Section 4.1 for an introduction to the perturbative frame transformation). Different from the problems previously studied, we introduce two small parameters. We use ϵ_1 to denote the small parameter with respect to g/Δ and ϵ_2 for Ω/Δ. Typically, ϵ_1 is much smaller than ϵ_2, i.e., the crosstalk coupling is much smaller than the detuning Δ. Therefore, we restrict our analysis to first-order errors of ϵ_1, which scale linearly with g. The parameter ϵ_2, on the other hand, depends on the strength of the drive or, equivalently, the gate duration.

Under this weak g coupling, the frequency of the qubits is renormalized, given in a so-called dressed frame. The dressed frame transformation that diagonalizes the g coupling between the two qubits is written as

$$\hat{S}_{0\to 1} = \epsilon_1 g\left(\hat{\sigma}_1^+\hat{\sigma}_2^- - \hat{\sigma}_1^-\hat{\sigma}_2^+\right). \tag{7.3}$$

With a pulse $\Omega_{\mathrm{c}} = \Omega(t)$, this transformation gives the effective Hamiltonian

$$\hat{H}_{0,1} = \exp\left(\hat{S}\right)\hat{H}_{\mathrm{d}}\exp\left(-\hat{S}\right) = \frac{\Omega\epsilon_2}{2}\hat{X}\hat{I} + \Delta\hat{\Pi}_2 + \frac{g\Omega\epsilon_1\epsilon_2}{2\Delta}\hat{Z}\hat{X} + \hat{H}_{\mathrm{S}} + \mathcal{O}(\epsilon_1^2). \tag{7.4}$$

Here we discard all the terms smaller or equal to ϵ_1^2. Also, we remove the Stark shift term $\hat{H}_{\mathrm{S}} = \frac{g^2}{\Delta}\hat{Z}\hat{I} - \frac{g^2}{\Delta}IZ$ by adjusting the drive frequency $\Delta \leftarrow \Delta - \frac{g^2}{\Delta}$ and rotating frame accordingly. The dressed Hamiltonian below is the starting point of the crosstalk analysis

$$\begin{aligned} \hat{H}_{1,1} &= \frac{\Omega\epsilon_2}{2}\hat{X}\hat{I} + \Delta\hat{\Pi}_2 + \frac{g\Omega\epsilon_1\epsilon_2}{2\Delta}\hat{Z}\hat{X} \\ &= \begin{pmatrix} 0 & \frac{g\Omega\epsilon_1\epsilon_2}{2\Delta} & \frac{\Omega\epsilon_2}{2} & 0 \\ \frac{g\Omega\epsilon_1\epsilon_2}{2\Delta} & \Delta & 0 & \frac{\Omega\epsilon_2}{2} \\ \frac{\Omega\epsilon_2}{2} & 0 & 0 & -\frac{g\Omega\epsilon_1\epsilon_2}{2\Delta} \\ 0 & \frac{\Omega\epsilon_2}{2} & -\frac{g\Omega\epsilon_1\epsilon_2}{2\Delta} & \Delta \end{pmatrix}. \end{aligned} \tag{7.5}$$

In the dressed frame, we observe the crosstalk term in the form of $\frac{g\Omega}{2\Delta}\hat{Z}\hat{X}$. This Hamiltonian is very similar to the Cross-Resonance Hamiltonian studied in Chapter 5. A key distinction in the current configuration is that the drive is resonant with the first qubit, whereas the second qubit, though undriven, is influenced by an entangling Hamiltonian ZX, renormalized by a factor g/Δ. Notice that the quantum crosstalk (entanglement) studied here is different from the classical crosstalk which is in the form of $\Omega_x\hat{I}\hat{X} + \Omega_y\hat{I}\hat{Y}$. In the latter case, an exact solution exists [65, 67].

7.3 First-derivative DRAG pulse

Similar to the derivation presented in Chapter 4, we define a generator that perturbatively removes the crosstalk term

$$\hat{S}_{1\to 2} = -i\frac{g\Omega\epsilon_1\epsilon_2}{2\Delta^2}\hat{Z}\hat{Y}, \tag{7.6}$$

which gives the effective Hamiltonian

$$\hat{H}_{1,2} = \frac{\Omega}{2}\hat{X}\hat{I} + \Delta\hat{\Pi}_2 + \frac{ig\dot{\Omega}}{2\Delta^2}\hat{Z}\hat{Y} + \frac{g\Omega^2}{2\Delta^2}\hat{Y}\hat{Y} = \begin{pmatrix} 0 & -\frac{ig\dot{\Omega}}{2\Delta^2} & \frac{\Omega}{2} & -\frac{g\Omega^2}{2\Delta^2} \\ \frac{ig\dot{\Omega}}{2\Delta^2} & \Delta & \frac{g\Omega^2}{2\Delta^2} & \frac{\Omega}{2} \\ \frac{\Omega}{2} & \frac{g\Omega^2}{2\Delta^2} & 0 & \frac{ig\dot{\Omega}}{2\Delta^2} \\ -\frac{g\Omega^2}{2\Delta^2} & \frac{\Omega}{2} & -\frac{ig\dot{\Omega}}{2\Delta^2} & \Delta \end{pmatrix}. \tag{7.7}$$

In the above and subsequent calculations, we keep the terms up to ϵ_2^2 and omit the explicit symbols ϵ_1 and ϵ_2 in the Hamiltonians for clarity. It is worth noting that the second-order YY error exists only in the quantum crosstalk. For classical crosstalk, the perturbative DRAG pulse effectively suppresses all the errors up to the third order.

To determine the DRAG correction, we also derive the control Hamiltonian in this frame, with a free parameter Ω_c to be determined

$$\hat{H}_{c,2}(\Omega_c) = \begin{pmatrix} 0 & \frac{g\bar{\Omega}_c}{2\Delta} & \frac{\bar{\Omega}_c}{2} & -\frac{g\Omega\bar{\Omega}_c}{2\Delta^2} \\ \frac{g\Omega_c}{2\Delta} & 0 & \frac{g\Omega\bar{\Omega}_c}{2\Delta^2} & \frac{\bar{\Omega}_c}{2} \\ \frac{\Omega_c}{2} & \frac{g\Omega\Omega_c}{2\Delta^2} & 0 & -\frac{g\bar{\Omega}_c}{2\Delta} \\ -\frac{g\bar{\Omega}\Omega_c}{2\Delta^2} & \frac{\bar{\Omega}_c}{2} & -\frac{g\Omega_c}{2\Delta} & 0 \end{pmatrix}. \tag{7.8}$$

By comparing the crosstalk elements, it is then clear that the DRAG correction is given as $\Omega_c = -i\dot{\Omega}/\Delta$. The result reads

$$\begin{aligned} \hat{H}_{2,2} &= \hat{H}_{1,2} + \hat{H}_{c,2}(-i\frac{\dot{\Omega}}{\Delta}) = \frac{\Omega}{2}\hat{X}\hat{I} + \frac{\dot{\Omega}}{2\Delta}\hat{Y}\hat{I} + \Delta\hat{\Pi}_2 + \frac{g\Omega^2}{2\Delta^2}\hat{Y}\hat{Y} + \frac{g\Omega\dot{\Omega}}{2\Delta^3}\hat{X}\hat{Y} \\ &= \begin{pmatrix} 0 & 0 & \frac{\Omega}{2} + \frac{i\dot{\Omega}}{2\Delta} & -\frac{g\Omega^2}{2\Delta^2} - \frac{ig\Omega\dot{\Omega}}{2\Delta^3} \\ 0 & \Delta & \frac{g\Omega^2}{2\Delta^2} + \frac{ig\Omega\dot{\Omega}}{2\Delta^3} & \frac{\Omega}{2} + \frac{i\dot{\Omega}}{2\Delta} \\ \frac{\Omega}{2} - \frac{i\dot{\Omega}}{2\Delta} & \frac{g\Omega^2}{2\Delta^2} - \frac{ig\Omega\dot{\Omega}}{2\Delta^3} & 0 & 0 \\ -\frac{g\Omega^2}{2\Delta^2} + \frac{ig\Omega\dot{\Omega}}{2\Delta^3} & \frac{\Omega}{2} - \frac{i\dot{\Omega}}{2\Delta} & 0 & \Delta \end{pmatrix}. \end{aligned} \tag{7.9}$$

Same as the single-Transmon DRAG in Section 4.3, a Y error is introduced by the correction, which can be transformed into a Z error with

$$\hat{S}_{2\to 3} = -i\frac{\Omega\epsilon_2}{2\Delta}\hat{I}\hat{Y}. \tag{7.10}$$

The effective Hamiltonian is given by

$$\hat{H}_{2,3} = \left(\frac{\Omega}{2} - \frac{\Omega^3}{4\Delta^2}\right)\hat{X}\hat{I} + \Delta\hat{\Pi}_2 + \frac{g\Omega^2}{2\Delta^2}\hat{Y}\hat{Y} + \frac{g\Omega\dot{\Omega}}{2\Delta^3}\hat{X}\hat{Y} - \frac{\Omega^2}{2\Delta}\hat{Z}\hat{I}. \tag{7.11}$$

From the above formula, focusing on the subspace of the first qubit, we extract the required detuning $-\frac{\Omega^2}{2\Delta}\hat{Z}\hat{I}$. By including third-order terms, we also obtain a renormalization of the pulse shape $\Omega + \frac{\Omega^3}{2\Delta^2}$. Notice that although it is also proportional to $\frac{\Omega^3}{2\Delta}$, the sign is different compared to the solution in Section 4.7 because the other one also has contributions from the $|0\rangle \leftrightarrow |2\rangle$ coupling. Including the phase ramping (see Section 4.4), the final pulse shape is given by

$$\Omega_{\text{DRAG-1}} = \left(\Omega + \frac{\Omega^3}{2\Delta^2} - i\frac{\dot{\Omega}}{\Delta}\right)\exp\left(i\int_0^T \frac{\Omega^2}{\Delta}\mathrm{d}t\right). \tag{7.12}$$

7.4 Second-derivative DRAG pulse

The first-derivative DRAG solution provides adequate suppression of crosstalk. However, unlike leakage error, the frequency difference between two qubits can be as small as a few tens of MHz due to fabrication inhomogeneity [121, 157]. This proximity in frequency can result in a significant Y error, which is proportional to $i\dot{\Omega}/\Delta$, as shown in equation (7.9), challenging the calibration procedure. In this case, the second-derivative DRAG correction can prove useful, because it provides a real correction and no detuning is required to correct the phase error.

This is motivated by classical frequency engineering. In the context of frequency engineering, the derivation of perturbative linear DRAG correction is equivalent to an integral by parts [67]. Under the assumption of weak coupling, an off-resonant excitation error like crosstalk can be estimated as

$$E = \left|\int_0^T \mathrm{e}^{i\Delta t}\Omega(t)\mathrm{d}t\right|^2 = \left|\int_0^T \mathrm{e}^{i\Delta t}\frac{\mathrm{d}^n}{\mathrm{d}^n t}\frac{\Omega(t)}{\Delta^n}\mathrm{d}t\right|^2, \tag{7.13}$$

where Ω represents the coupling and Δ the energy separation. In addition, the boundary condition $\frac{\mathrm{d}^n\Omega(t)}{\mathrm{d}^n t}|_{t=0} = \frac{\mathrm{d}^n\Omega(t)}{\mathrm{d}^n t}|_{t=T} = 0$ needs to be satisfied for all n. Therefore, for a pulse shape that is n times differentiable and the derivatives at the boundaries are zero, there exist n different but equivalent (up to the first-order perturbation) linear DRAG solutions. Specifically, the DRAG pulse takes the form $\Omega - i^n\frac{\mathrm{d}^n}{\mathrm{d}^n t}\frac{\Omega}{\Delta^n}$, with corrections alternating between imaginary and real values depending on whether n is odd or even.

In the following, we formally derive the second-derivative DRAG solution for the crosstalk. As shown later in the simulation, for this particular crosstalk problem, the DRAG strength optimization of the second-order DRAG provides significant improvement due to destructive interference. It is important to highlight that this second-derivative solution is (perturbatively) equivalent to the first-derivative solution. It still aims to suppress the first-order crosstalk coupling without addressing higher-order effects. However, due to the increased frequency bandwidth associated with the second-derivative DRAG, its performance in suppressing leakage may not match that of the first-derivative DRAG, as will be demonstrated in the simulation results in later sections.

We start from the effective Hamiltonian after the first perturbation, $\hat{H}_{1,2}$, in equation (7.7) and define the generator

$$\hat{S}_{2\to4} = \frac{ig\epsilon_1\epsilon_2\dot{\Omega}}{2\Delta^3}\hat{Z}\hat{X}, \tag{7.14}$$

which results in the Hamiltonian

$$\hat{H}_{1,4} = \frac{\Omega}{2}\hat{X}\hat{I} + \Delta\hat{\Pi}_2 - \frac{g\ddot{\Omega}}{2\Delta^3}\hat{Z}\hat{X} + \frac{g\Omega^2}{2\Delta^2}\hat{Y}\hat{Y} - \frac{g\Omega\dot{\Omega}}{2\Delta^3}\hat{Y}\hat{X}. \tag{7.15}$$

The imaginary error term $\hat{Z}\hat{Y}$ transforms into a real one, $-\frac{g\ddot{\Omega}}{2\Delta^3}\hat{Z}\hat{X}$. There is no phase mismatch or imaginary part on the first qubit. The control Hamiltonian in this frame reads similarly as

$$\hat{H}_{c,4} = \begin{pmatrix} 0 & \frac{g\bar{\Omega}_c}{2\Delta} & \frac{\bar{\Omega}_c}{2} & -\frac{g\Omega\bar{\Omega}_c}{2\Delta^2} + \frac{ig\bar{\Omega}_c\dot{\Omega}}{2\Delta^3} \\ \frac{g\Omega_c}{2\Delta} & 0 & \frac{g\Omega\bar{\Omega}_c}{2\Delta^2} + \frac{ig\bar{\Omega}_c\dot{\Omega}}{2\Delta^3} & \frac{\bar{\Omega}_c}{2} \\ \frac{\Omega_c}{2} & \frac{g\Omega\Omega_c}{2\Delta^2} - \frac{ig\Omega_c\dot{\Omega}}{2\Delta^3} & 0 & -\frac{g\bar{\Omega}_c}{2\Delta} \\ -\frac{g\Omega\Omega_c}{2\Delta^2} - \frac{ig\Omega_c\dot{\Omega}}{2\Delta^3} & \frac{\Omega_c}{2} & -\frac{g\Omega_c}{2\Delta} & 0 \end{pmatrix}. \tag{7.16}$$

With its DRAG correction $\Omega_c = \frac{\ddot{\Omega}}{\Delta^2}$, the effective Hamiltonian reads as follows

$$\hat{H}_{3,4} = \left(\frac{\Omega}{2} + \frac{\ddot{\Omega}}{\Delta^2}\right)\hat{X}\hat{I} + \Delta\hat{\Pi}_2 + \left(\frac{g\Omega^2}{2\Delta^2} + \frac{g\Omega\ddot{\Omega}}{2\Delta^4}\right)\hat{Y}\hat{Y} + \left(-\frac{g\Omega\dot{\Omega}}{2\Delta^3} - \frac{g\dot{\Omega}\ddot{\Omega}}{2\Delta^5}\right)\hat{Y}\hat{X}. \tag{7.17}$$

The resulting final expression for the second-derivative DRAG solution aligns with that derived from semiclassical pulse engineering:

$$\Omega_{\text{DRAG-2}} = \Omega + \frac{\ddot{\Omega}}{\Delta^2}. \tag{7.18}$$

The amplitude of the drive Ω may need a recalibration to achieve the desired rotation angle. The remaining dominant error manifests as the swap types of coupling between the qubits.

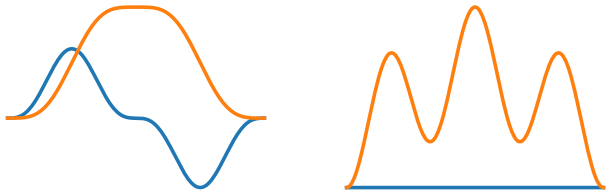


Figure 7.1: Examples of the analytical pulse shape for a π rotation with DRAG-1 and DRAG-2 corrections with the parameters $\Delta = 40$ MHz and gate time $T = 30$ ns, based on the initial pulse given in equation (7.19). Orange and blue represent the real and imaginary parts of the pulse. Due to the small detuning chosen, the change of the DRAG corrections to the initial pulses is quite significant. The final shape also varies with different choices of initial pulse, especially for DRAG-2.

7.5 Numerical simulation

To test the crosstalk suppression between two qubits, we perform simulations involving an undesired small interaction, where two qubits are positioned closely in frequency space due to either fabrication inhomogeneity or frequency collision in the design. The parameters used are $\Delta = 40$ MHz (representing the energy separation between qubit states) and $g = 2$ MHz (indicating the strength of the qubit-qubit coupling). This emulates a situation where neighbouring qubits experience crosstalk due to their proximity in frequency.

We use the initial pulse shape defined in equation (5.13) with $m = 3$, given by

$$\Omega_{\text{initial}}(t) = \mathcal{I}_{\max}\left(\frac{1}{16}\cos\left(6\pi\frac{t}{T}\right) - \frac{9}{16}\cos\left(\pi\frac{2t}{T}\right) + \frac{1}{2}\right). \tag{7.19}$$

The normalization $\mathcal{I}_{\max}$ ensures that the total area of the integral of $\Omega_{\text{initial}}(t)$ from 0 to T is the desired rotation angle, which is a π rotation in the following calculation.

In Figure 7.2, we show the measured leakage error as a function of different gate time, with $\int_0^T \Omega_{\text{initial}}(t)\mathrm{d}t = \pi$ and the analytical drive shape defined in equations (7.12) and (7.18). The crosstalk error is defined as

$$L = \frac{1}{2}\sum_k \left|\langle k,0|\,\hat{U}\,|k,1\rangle\right|^2 + \left|\langle k,1|\,\hat{U}\,|k,0\rangle\right|^2, \tag{7.20}$$

where $\hat{U}$ denotes the unitary propagator. The performance is limited for gate duration below 20 ns due to the small detuning Δ, which challenges the perturbative assumptions underlying the analytical solutions. For instance, with a gate duration set at 20 ns, the maximum drive amplitude needed is 50 MHz. As the gate time increases, the perturbative DRAG solutions demonstrate improved performance.

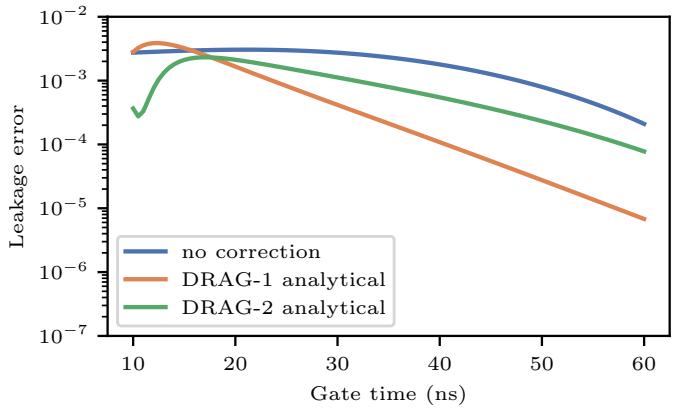


Figure 7.2: Leakage error for a π rotation as a function of gate time using the analytical DRAG pulse.

In addition, one observes that the analytical second-derivative DRAG solution, though expected to be equivalent to the first-derivative one in classical engineering, performs less well because of the increased bandwidth. Nevertheless, subsequent analyses reveal that with an optimized DRAG strength, the second-derivative DRAG can significantly outperform its first-derivative counterpart in specific parameter regimes.

To study the optimization of DRAG parameters, we define modified versions of the two DRAG solutions parameterized by additional free parameters, which can be optimized to improve performance. Specifically, for the DRAG-1 pulse, we write

$$\Omega'_{\text{DRAG-1}} = \Omega_{\max}\left(\Omega - ia\frac{\dot{\Omega}}{\Delta}\right)\exp\left(ia_\delta\int_0^T\frac{\Omega^2}{\Delta}\mathrm{d}t\right), \tag{7.21}$$

with a, a_δ and $\Omega_{\max}$ represent the DRAG strength, the detuning strength and the drive strength. All three are constants and parameters that can be calibrated in experiments. It is important to note that the detuning in this context must be time-dependent. Due to the large Y error introduced by the DRAG correction, a constant detuning, as typically used for single Transmon leakage suppression [134], is insufficient for most parameter regimes in this study.

For each specified gate duration, we optimize the DRAG parameters to maximize the gate fidelity defined as

$$F = \max_\theta\left|\mathrm{tr}\left[\mathrm{e}^{i\theta\hat{\Pi}_2}\hat{U}\hat{U}^\dagger_{\text{ideal}}\right]\right|. \tag{7.22}$$

where the additional Z rotation on the second qubit is employed to counteract phase accumulation during the gate operation. The ideal unitary is defined as $\hat{U}_{\text{ideal}} = \hat{X}\hat{I}$, i.e., an X gate on the first qubit. The optimization results, illustrated

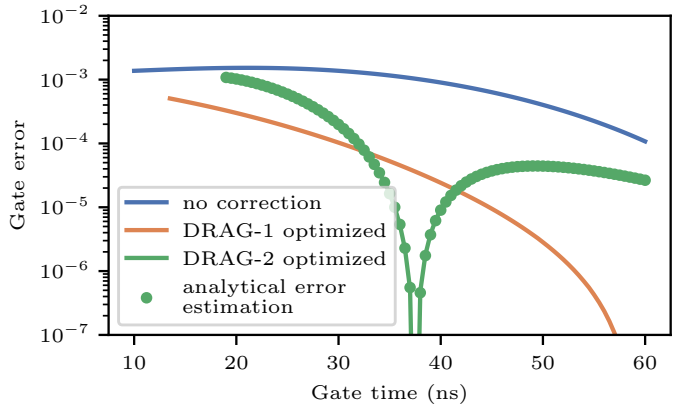


Figure 7.3: Infidelity fora π rotation as a function of gate time using parameterized DRAG pulses, with optimized DRAG parameters.

in Figure 7.3, reveal that for gate times up to approximately 50 ns, the performance of the optimized pulse closely aligns with the analytical pulse shape.

For the second-derivative DRAG pulse, we define the parameterized pulse as follows:

$$\Omega'_{\text{DRAG-2}} = \Omega_{\max} \left(\Omega + a \frac{\ddot{\Omega}}{\Delta^2} \right). \tag{7.23}$$

This DRAG solution does not include a phase correction and therefore has fewer parameters compared to other DRAG pulse variants. Following the same optimization procedure, we evaluated the gate fidelity depicted in Figure 7.3. Surprisingly, around 37 ns, we observe a minimum in the gate error due to destructive interference of the crosstalk dynamics. This presents a particularly favourable scenario for experimental implementation. The existence and location of this minimum leakage error depend on the properties of the initial pulse shape. Different choices of initial pulse shapes can alter the characteristics of this minimum. Furthermore, it is important to point out that in the Transmon regime, additional DRAG corrections are necessary to address leakage to higher levels, where a small constant detuning is still required.

7.6 Analytical expression for the gate error

In the numerical simulation above we showed that optimization of the DRAG coefficient reveals significant performance improvement at specific gate duration. In this section, we aim to understand this by deriving an analytical error estimation for the crosstalk error. We begin our analysis with the Hamiltonian provided in equation (7.15), i.e., the effective Hamiltonian after a second diagonalization. Instead of fully removing the ZX crosstalk term, we only partially compensate for

the error, with a DRAG parameter a:

$$\hat{H}_{4,4} = \left(\frac{\Omega}{2} + \frac{a\ddot{\Omega}}{2\Delta^2}\right)\hat{X}\hat{I} + \Delta\hat{\Pi}_2 + \hat{H}_{\text{error},4}. \tag{7.24}$$

We consider the first two terms as the desired dynamics (up to a recalibration of gate time) and the rest of them as errors:

$$\hat{H}_{\text{error},4} = \Omega_{ZX}\hat{Z}\hat{X} + \Omega_{YY}\hat{Y}\hat{Y} + \Omega_{YX}\hat{Y}\hat{X}. \tag{7.25}$$

The amplitude of terms in $\hat{H}_{\text{error},4}$ is assumed to be much smaller than the main dynamics Ω_{XI} and Δ

$$\Omega_{ZX} = \left(-\frac{g\ddot{\Omega}}{2\Delta^3} + a\frac{g\ddot{\Omega}}{2\Delta^3}\right), \tag{7.26}$$

$$\Omega_{YY} = \left(\frac{g\Omega^2}{2\Delta^2} + a\frac{g\Omega\ddot{\Omega}}{2\Delta^4}\right), \tag{7.27}$$

$$\Omega_{YX} = \left(-\frac{ig\Omega\dot{\Omega}}{2\Delta^3} - a\frac{ig\dot{\Omega}\ddot{\Omega}}{2\Delta^5}\right). \tag{7.28}$$

When analyzing unwanted transition errors relative to idling dynamics, it is often sufficient to estimate the transition probability by integrating the coupling strength, modulated with the eigenfrequency [65, 67, 89]. This approach corresponds to a first-order Magnus expansion. However, applying the same method to study leakage is challenging because the dominant dynamic implements a π rotation. Under this assumption, the condition of Magnus expansion does not converge because $\int |\hat{H}|\mathrm{d}t \approx \pi$, i.e., desired dynamics flips the state of the first qubit. Therefore, a simple integral of the leakage term does not accurately capture the leakage dynamics.

To isolate the error term, we must first transform the Hamiltonian to the interaction picture with respect to the desired dynamics. We define

$$\hat{H}_{X1} = \left(\frac{\Omega}{2} + a\frac{\ddot{\Omega}}{2\Delta^2}\right)\hat{X}\hat{I} \tag{7.29}$$

$$, \tag{7.30}$$

and the corresponding frame transformation

$$\hat{V}_{4\to5} = \exp\left(i\int \hat{H}_{XI}\mathrm{d}t\right) = \exp\left(i\frac{\theta(t)}{S}2\hat{X}\hat{I}\right), \tag{7.31}$$

with $\theta(t) = \int\left(\Omega + a\frac{\ddot{\Omega}}{\Delta^2}\right)\mathrm{d}t$.

The above transformation transforms the error into the interaction picture, the remaining terms are all in the second order, considerably small compared to Δ and Ω. We have

$$\hat{H}_{4,5} = \Delta\hat{\Pi}_2 + \hat{H}_{\text{error},5}, \tag{7.32}$$

with

$$\begin{aligned}\hat{H}_{\text{error},5} = &\left(-\Omega_{YX}\sin(\theta t) + \Omega_{ZX}\cos(\theta t)\right)\hat{Z}\hat{X} - \Omega_{YY}\sin(\theta t)\hat{Z}\hat{Y} \\ &+ \left(\Omega_{YX}\cos(\theta t) + \Omega_{ZX}\sin(\theta t)\right)\hat{Y}\hat{X} + \Omega_{YY}\cos(\theta t)\hat{Y}\hat{Y}. \end{aligned} \tag{7.33}$$

Because $\hat{H}_{\text{error},5}$ is small, we can now use the first-order Magnus expansion and obtain two types of errors: a Cross-Resonance type of coupling

$$C_1 = \int \mathrm{e}^{it\Delta}\left(-\sin(\theta t)\Omega_{YX} - i\sin(\theta t)\Omega_{YY} + \cos[t\theta]\Omega_{ZX}\right)\mathrm{d}t, \tag{7.34}$$

and a SWAP type

$$C_2 = \int i\mathrm{e}^{it\Delta}\left(\cos(\theta t)\Omega_{YX} + i\cos(\theta t)\Omega_{YY} + \sin[t\theta]\Omega_{ZX}\right)\mathrm{d}t. \tag{7.35}$$

The total error is given as

$$E = 1 - \frac{1}{4}\left|\mathrm{Tr}\left[\exp\left(-i\int \mathrm{e}^{i\Delta\hat{\Pi}_2}\hat{H}_{\text{error},5}\mathrm{d}t\right)\right]\right| \tag{7.36}$$

$$\approx \frac{1}{4}\left|\mathrm{Tr}\left[-\frac{1}{2}\left(\int \mathrm{e}^{i\Delta\hat{\Pi}_2}\hat{H}_{\text{error},5}\mathrm{d}t\right)^2\right]\right| \tag{7.37}$$

$$= \frac{1}{2}\left(|C_1|^2 + |C_2|^2\right), \tag{7.38}$$

where in the approximation above we expand the exponential and use the fact that $\hat{H}_{\text{error},5}$ is traceless.

This error metric includes all the gate errors (captured up to the second-order perturbation) except for the rotation angle error. Instead of optimizing using the fidelity of the time evolution propagator, one can optimize this specific error metric. The minimized gate error is shown in Figure 7.3 as green dots, which agrees very well with the numerically optimized value.

7.7 Discussion

In this chapter, we derive two DRAG solutions to suppress quantum crosstalk for single-qubit drives, demonstrating that even under conditions of relatively small detuning, DRAG correction still provides adequate improvement in gate fidelity. Additionally, we show that calibration of the DRAG strength reveals a zero point of leakage error for the second-derivative DRAG correction and provide an analytical formula for error estimation.

The quantum crosstalk model examined here is a simplified one, focusing on achieving a perfect π rotation on one qubit while minimizing leakage to the neighbouring qubit. Despite the simplicity of the model, similar principles apply to

more complex quantum crosstalk scenarios, such as the spectator qubits problem with Cross-Resonance gates [81]. Following the recursive DRAG formulation introduced in Chapter 5, the solution here can also be combined with other DRAG corrections, including those addressing leakage to higher energy levels.

One particular observation is that the DRAG correction under this quantum crosstalk model is independent of crosstalk strength g and only relies on the drive strength Ω and detuning Δ. This can also be explained from the perspective of frequency engineering. The first perturbative explanation of DRAG involves engineering the Fourier properties of the pulse such that the unwanted frequency component is minimized, which is independent of the magnitude of the coupling strength. This imposes a limit on the DRAG correction in cases of very small detuning: Regardless of the leakage coupling strength, the same amount of DRAG correction needs to be applied, complicating phase and rotation angle calibration.

Another challenge not addressed here is that the DRAG correction does not affect the hybridization between the two qubits. In particular, if the second qubit is subject to another drive, the calibration of this drive may depend on the DRAG frame of the first qubit. For parallel execution of gates on both qubits, this dependence needs to be analytically characterized to provide parameterized pulse shapes for experimental calibration.

8

Pulse-level quantum circuit simulation with QuTiP

The study of the impact of noise on quantum circuits is especially relevant to guide the progress of Noisy Intermediate-Scale Quantum (NISQ) computing. In this chapter, we address the pulse-level simulation of noisy quantum circuits with the Quantum Toolbox in Python (QuTiP). We introduce new tools in `qutip-qip`*, QuTiP's quantum information processing package. These tools simulate quantum circuits at the pulse level, leveraging QuTiP's quantum dynamics solvers and control optimization features. We show how quantum circuits can be compiled on simulated processors, with control pulses acting on a target Hamiltonian that describes the unitary evolution of the physical qubits. Various types of noise can be introduced based on the physical model, e.g., by simulating the Lindblad density-matrix dynamics or Monte Carlo quantum trajectories. In particular, the user can define environment-induced decoherence at the processor level and include noise simulation at the level of control pulses. We illustrate how the Deutsch-Jozsa algorithm is compiled and executed on a superconducting-qubit-based processor, on a spin-chain-based processor and using control optimization algorithms. We also show how to easily reproduce experimental results on cross-talk noise in an ion-based processor, and how a Ramsey experiment can be modeled with Lindblad dynamics. Finally, we illustrate how to integrate these features with other software frameworks.*

This chapter has been published, with minor changes, as B. Li, S. Ahmed, S. Saraogi, N. Lambert, F. Nori, A. Pitchford, and N. Shammah, *Pulse-level noisy quantum circuits with QuTiP*, Quantum 6, 630 (2022). [235]. The thesis author contributed significantly to the code base, composed all the figures and made significant contributions in writing the manuscript.

8.1 Introduction

Quantum computation and quantum algorithms are deemed to be able to complete tasks that would be harder or impossible to achieve with classical resources. However, noise on quantum hardware significantly influences its performance, limiting large-scale applications. Currently, we are in the so-called noisy intermediate-scale quantum (NISQ) computing era [236]. Before we reach the regime of quantum error correction (QEC) [3], quantum algorithms will suffer from quantum and classical noise, e.g., decoherence and noise in classical control signals. Both types of noise lead to errors in the computation and therefore determine the performance of a quantum algorithm. Hence, a realistic simulation of a quantum algorithm needs to incorporate these different types of noise, which can depend strongly on the type of qubit technology [237].

A modern quantum algorithm typically includes both classical and quantum parts [238]. The former can include classical variational subroutines, while the latter is usually represented by a quantum circuit, consisting of a number of gates applied on a quantum state. Many software projects provide the simulation of such circuits including PyQuil [239, 240], Qiskit [241], Cirq [242], ProjectQ [243], and PennyLane [244], among others [245, 246]. However, within these approaches, noise is usually modelled as an additional layer on top of ideal quantum gates, e.g., probabilistically inserting random Pauli gates or a list of Kraus operators to describe a noisy quantum channel.

To improve the performance of a quantum circuit on noisy hardware, it is useful to also perform optimization at the level of control pulses based on the quantum dynamics of the underlying hardware. For this purpose, open-source software packages have been developed to map quantum circuits to control pulses on hardware, allowing for fine-tuning and calibration of the control pulses, such as `qiskit.pulse` [161], `qctrl-open-controls` [247] and `Pulser` [248]. Recently, Qiskit also launched the project `qiskit-dynamics` to support solving time-dependent quantum systems, connected with `qiskit.pulse`. The project is still in the early stages of development.

In the realm of simulation, one of the earliest, and most widely used Python packages to simulate quantum dynamics is the Quantum Toolbox in Python, QuTiP [27, 249]. QuTiP provides useful tools for handling quantum operators and simplifies the simulation of a quantum system under a noisy environment by providing a number of solvers, such as the Lindblad master equation solver. An ecosystem of software tools for quantum technology is growing around it [24, 29, 248, 250–255]. Hence, it is a natural base to start connecting the simulation of quantum circuits and the time evolution of the quantum system representing the circuit registers. At the cost of more computing resources, simulation at the level of time evolution allows noise based on the physical model to be included in the realistic study of quantum circuits.

In this chapter, we illustrate how the new tools in `qutip-qip`[1] can be used

[1] `https://github.com/qutip/qutip-qip`

to bridge the gap between the gate-level circuit simulation and the simulation of quantum dynamics following the master equation for various hardware models. While a quantum circuit representation and a few specific Hamiltonian models have been available in QuTiP for some time, in this chapter, we bridge them with QuTiP solvers and build a pulse-level simulation framework, allowing the simulation of noisy circuits.

Provided a Hamiltonian model and a map between the quantum gate and control pulses, we show how these new tools in `qutip-qip` can be used to compile the circuit into the native gates of a given hardware, how to generate the physical model described by control pulses and how to use QuTiP's dynamical solvers to obtain the full-state time evolution, as shown in Figure 8.1.

A number of example hardware models are available in the software package – a spin qubit processor, a cavity-QED device, a superconducting qubit model – while in general the users are provided with the freedom to define their own devices of choice. In addition to a predefined map between gates and pulses for each model, optimal control algorithms in QuTiP can also be used to generate control pulses. Moreover, we demonstrate how various types of noise, including decoherence induced by the quantum environment and classical control noise, can be introduced at different layers of the simulation. Thanks to a modular code design, one can quickly extend the toolkit with customized hardware and noise models.

This chapter is organized as follows: In Section 8.2, information about the software installation and specifics is given. In Section 8.3, we briefly present the background concepts of quantum circuits at the gate level, the continuous-time pulse-level description for circuits, open quantum systems theory and the tools present in `qutip` and `qutip-qip` to represent and simulate open quantum systems. Section 8.4 contains the main novel results and new software features: therein, we illustrate in detail the novel architecture of the pulse-level quantum-circuit simulation framework in `qutip-qip` and the available modelling of quantum devices and noise. In Section 8.5, we show how these features can be integrated with other software by importing external quantum circuits using the QASM format. We conclude in Section 8.6.

The Appendices include self-contained code examples: Section 8.8.1 contains the full code for the Deutsch-Jozsa algorithm simulation; Section 8.8.2 presents the simulation of a 10-qubit quantum Fourier transform (QFT) algorithm using the spin chain model; Section 8.8.3 shows how to customize the physical model of a processor with noise. More examples can also be found in QuTiP tutorials[2].

8.2 Software information

The tools described here are part of the QuTiP project [27, 249]. The `qutip-qip` package builds upon what was once a module of QuTiP, `qutip.qip`. Usage and installation has not significantly changed for the end user, who can easily install

[2] `http://qutip.org/tutorials.html` under the section Quantum Information Processing

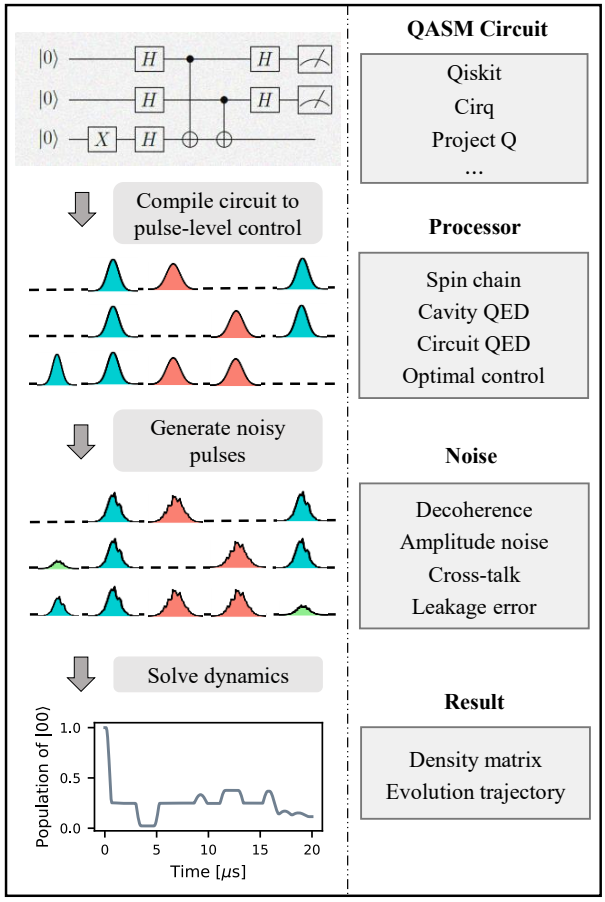


Figure 8.1: Illustration of the workflow of the pulse-level noisy quantum circuit simulation. It starts from a quantum circuit defined in QuTiP or imported from other libraries through the QASM format. Based on the hardware of interest, the circuit is then compiled to control pulse signals for each control Hamiltonian (blue for single-qubit gates and red for two-qubit gates in the figure). Next, a representation of the time evolution, including various types of noise, is generated under the description of the master equation. In the last step, the QuTiP solver is employed to solve the dynamics. The solver returns the final result as well as the intermediate state information on demand. Both the final and the intermediate quantum states can be recorded, as illustrated by the plot showing the population of the $|00\rangle$ state, with the third qubit traced out. This plot is the same as Section 8.8.1 and will be explained later in detail. The control signals in the figure are for illustration purposes only while the real compiled pulses on a few predefined hardware models are shown in Figure 8.3.

the package from the Python package index (PyPI) distribution with

Code Listing 8.1: Installing qutip-qip

```
pip install qutip-qip
```

The `qutip-qip` package has the core `qutip` package as its main dependency. This means that it also builds upon the wider Python scientific open source software stack, including NumPy [256] and SciPy [257], and optionally Matplotlib [258] and Cython [259]. `qutip-qip` is a software developed by many contributors [260].

The `qutip-qip` package is developed with the best practices of open-source software development and scientific software. The codebase is hosted on GitHub and new code contributions are reviewed by the project maintainers. The license is the BSD three-clause license (also known as BSD 2.0 or New BSD). The code is thoroughly unit-tested, with tests for most objects also running on the cloud in continuous integration, on multiple operating systems. The documentation, whose code snippets and API documentation are also unit tested, is hosted online on Read The Docs (`https://qutip-qip.readthedocs.io/`); the documentation can also be generated locally by contributors with Sphinx by forking the `QuTiP/qutip-qip` Github repository.

8.3 Quantum circuits and open quantum dynamics

In this section, we briefly review the theory of quantum circuits and their modelling on actual devices that are subject to noise. We introduce the formalism for the gate-level representation of quantum circuits, then describe the Hamiltonian description at the pulse level, and finally the open-quantum dynamics of a realistic system.

8.3.1 Quantum circuits and gate-level simulation

A quantum circuit is a model for quantum computation, where the quantum dynamics is abstracted and broken down into unitary matrices (quantum gates), which can be applied to all or only a few circuit registers. Inherited from classical computing, the circuit registers are most often two-level systems, referred to as qubits. The execution of a circuit on a quantum state is then given by

$$|\psi_f\rangle = \hat{U}_K \hat{U}_{K-1} \cdots \hat{U}_2 \hat{U}_1 |\psi_i\rangle , \tag{8.1}$$

where $|\psi_i\rangle$ and $|\psi_f\rangle$ are the initial and final state and $\hat{U}_k$ with $k \in \{1, 2, \cdots, K\}$ the quantum gates.

Often, the simulation of quantum circuits is implemented by representing the unitaries and quantum states as complex matrices and vectors. The execution of a circuit is then described as matrix-vector multiplication. We refer to this as gate-level quantum circuit simulation. The gate-level quantum circuit description is a representation of a quantum algorithm at an abstract level before considering any

physical realization to implement the algorithm [3]. More general representations of hybrid quantum algorithms include the integration of classical and quantum subroutines, as for variational quantum algorithms [238], their compilation and execution [240, 244, 261–263]. In `qutip-qip`, this gate-level simulation can be performed with the `QubitCircuit` class, which is the Python object used to represent a quantum circuit.

In order to introduce the effects of noise, quantum states can be most generally represented by a density matrix and the idea of a quantum channel is introduced, where noise can be characterized by a set of non-unitary Kraus operators acting on the quantum states. Many well-known channel representations of noise have been implemented in circuit simulation, such as depolarising, dephasing, amplitude damping and erasure channels. Although the channel description is very general, noisy gate simulation based on it has two shortcomings.

First, in most implementations, noise is applied after the ideal gate unitaries, while in reality they are not separated. Second, although quantum channels describe the most general evolution that a quantum system can undergo, finding the representation of realistic noise in this channel form is not a trivial task. Usually, a noise channel implemented in simulators only describes single-qubit decoherence and cannot accurately capture the complicated noisy evolution that the system undergoes. Hence, to study the execution of circuits on noisy hardware in more detail, one needs to turn to the quantum dynamics of the hardware platform.

8.3.2 Continuous time evolution and pulse-level description

Down to the physical level, quantum hardware, on which a circuit is executed, is described by quantum theory. The dynamics of the system that realizes a unitary gate in equation (8.1) is characterized by the time evolution of the quantum system. For isolated or open quantum systems, we consider both unitary time evolution and open quantum dynamics. The latter can be simulated either by solving the master equation or sampling Monte Carlo trajectories. Here, we briefly describe those methods as well as the corresponding solvers available in QuTiP.

Unitary time evolution

For a closed quantum system, the dynamics is determined by the Hamiltonian and the initial state. From the perspective of controlling a quantum system, the Hamiltonian is divided into the non-controllable drift $\hat{H}_\mathrm{d}$ (which may be time-dependent) and controllable terms combined as $\hat{H}_\mathrm{c}$ to give the full system Hamiltonian

$$\hat{H}(t) = \hat{H}_\mathrm{d}(t) + \hat{H}_\mathrm{c}(t) = \hat{H}_\mathrm{d}(t) + \sum_j c_j(t)\hat{H}_j, \tag{8.2}$$

where the $\hat{H}_j$ describes the effects of available physical controls on the system that can be modulated by the time-dependent control coefficients $c_j(t)$, by which one drives the system to realize the desired unitary gates.

The unitary $\hat{U}$ that is applied to the quantum system driven by the Hamiltonian $\hat{H}(t)$ is a solution to the Schrödinger operator equation

$$i\hbar\frac{\partial\hat{U}(t)}{\partial t} = \hat{H}(t)\hat{U}(t). \tag{8.3}$$

By choosing $\hat{H}(t)$ that implements the desired unitaries (the quantum circuit) we obtain a pulse-level description of the circuit in the form of equation (8.2). The choice of the solver depends on the parametrization of the control coefficients $c_j(t)$. The parameters of $c_j(t)$ may be determined through theoretical models or automated through control optimisation, as described later in Section 8.4.

Open quantum system dynamics

In reality, a quantum system is never perfectly isolated; hence, a unitary evolution is often only an approximation. To consider possible interaction with the environment, one can introduce a larger Hilbert space, or reduce the overhead by effectively limiting the description to the system Hilbert space and using super-operators inducing non-unitary dynamics (i.e., on an open system). One way to describe the evolution of an open quantum system is by the Lindblad master equation. It can be solved either by solving a differential equation (`qutip.mesolve`) or by Monte Carlo sampling of quantum trajectories (`qutip.mcsolve`). Both can be chosen as a simulation back-end for the pulse-level circuit simulator.

These solvers provide an efficient simulation of open-system quantum dynamics. They can describe noise models derived under the Born-Markov Secular (BMS) approximations [43, 264], and more general Lindbladians, including those with time-dependent rates. For most hardware implementations these noise models are powerful and flexible enough to capture the most salient environmental noise effects.

Density-matrix master equation solver. The function `qutip.mesolve` can solve general open dynamics that can be cast in the form

$$\frac{\partial\hat{\rho}(t)}{\partial t} = \mathcal{L}\hat{\rho}(t), \tag{8.4}$$

where the dynamics of the "system" density matrix $\hat{\rho}(t)$ evolves under the action of a superoperator $\mathcal{L}$. The user can decide to provide directly the full superoperator $\mathcal{L}$, or divide the dynamics in the Hamiltonian part [(8.2)] and noise terms provided by a set of collapse operators (`c_ops`) with related rates, and `qutip.mesolve` will effectively solve Eq. (8.4) behind the scenes. The structure of Eq. (8.4) can be quite generic, including the possibility for time-dependent rates and collapse operators, beyond the Born-Markov and secular (BMS) approximation, however, one of the most straightforward approaches is to simulate a Lindblad master equation. A

common example of a quantum circuit consisting of N qubits experiencing relaxation and dephasing would be the following Lindblad master equation,

$$\frac{\partial\hat{\rho}(t)}{\partial t} = -i\left[\hat{H}(t),\hat{\rho}(t)\right] + \sum_{j=0}^{N-1}\gamma_j\mathcal{D}[\hat{\sigma}_j^-]\hat{\rho}(t) + \sum_{j=0}^{N-1}\frac{\gamma_j^D}{2}\mathcal{D}[\hat{\sigma}_j^z]\hat{\rho}(t), \tag{8.5}$$

where $\hat{H}$ is the system Hamiltonian, γ_j is the relaxation rate of qubit j, γ_j^D the pure dephasing rate of qubit j, $\mathcal{D}[\Gamma_n]X = \Gamma_n X\Gamma_n^\dagger - \frac{1}{2}\Gamma_n^\dagger\Gamma_n X - \frac{1}{2}X\Gamma_n^\dagger\Gamma_n$ is the Lindblad dissipator for a generic jump operator Γ_n acting on a density matrix X, and $\hat{\sigma}_j^\alpha$ are Pauli operators, with $\alpha = x, y, z, +, -$.

This approach allows us to model the coexistence of pulse-level control, in the coherent Hamiltonian part, and the influence of noise. However, the density matrix description of the system introduces a quadratic overhead in memory size. If this becomes a limiting factor for a given simulation, progress can be made by employing the Monte-Carlo quantum trajectory solver, `qutip.mcsolve`.

Monte-Carlo quantum trajectories. A popular method that is alternative to the full master equation simulation is the Monte Carlo sampling with quantum trajectories. Noise is included in an effective non-Hermitian Hamiltonian, and a stochastic term is added by pseudo-random sampling. An effective Hamiltonian is continuously applied to the system, integrating the part of equation (8.5) with Lindblad dissipators,

$$\hat{H}_{\text{eff}} = \hat{H}(t) - \frac{i}{2}\sum_n \hat{\Gamma}_n^\dagger\hat{\Gamma}_n, \tag{8.6}$$

while the second part is determined stochastically, checking if a random number is greater than the norm of the unnormalized wave function. If that is the case, the quantum jump is applied, ensuring the renormalization of the wavefunction,

$$|\psi(t+\delta t)\rangle = \frac{\hat{\Gamma}_n|\psi(t+\delta t)\rangle}{\sqrt{\langle\psi(t)|\hat{\Gamma}_n^\dagger\hat{\Gamma}_n|\psi(t)\rangle}}. \tag{8.7}$$

The advantage of the quantum trajectory approach over the density-matrix master equation solution is that one needs to handle a computational space of dimension N equal to the Hilbert space, instead of its square. Additionally, the quantum-trajectory approach allows simulating the dynamics of single executions instead of the averaged dynamics from a density-matrix simulation using the master equation, which can provide further insight in processes that may be washed out when looking only at the statistical averages [265, 266]. A trade-off is present in the number of trajectories that need to be run to evaluate a mean path with a small standard deviation. However, the trajectories can be computed in parallel. QuTiP uses Python's multiprocessing module to benefit from multi-core computing platforms.

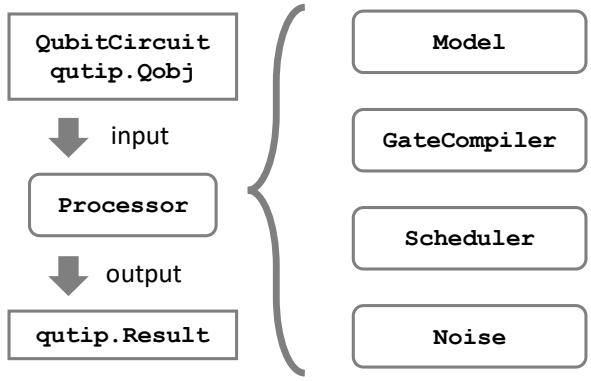


Figure 8.2: The structure of the simulation framework. The main interface is implemented in the class `Processor`. An instance of `Processor` emulates a quantum processor that takes a circuit and an initial quantum state as input and outputs the result as a `qutip.Result` object. From the result, one can inspect the final state of the physical qubits, as well as intermediate results during the time evolution. The `Processor` has a modular design that allows for arbitrary specifications of the underlying hardware model, compilation, scheduling gates and noise models.

Other dynamical solvers. QuTiP also provides solvers for other noise models and dynamics, such as the (secular and non-secular) Bloch-Redfield equation [43], the (non-Markovian) hierarchical equation of motion (HEOM) [252, 267], and stochastic master equations. These are not currently supported for the pulse-level circuit simulation of `qutip-qip`.

8.4 Pulse-level quantum-circuit simulation framework

In this section, we describe the architecture of the simulation framework. The framework aims at simplifying the simulation of noisy quantum circuits through the explicit time evolution of physical qubits using QuTiP solvers. As illustrated in Figure 8.2, the simulation is designed around a `Processor` class, which consists of several different components. An instance of `Processor` emulates the behaviour of a quantum processor that takes a quantum circuit (`QubitCircuit`) as well as an initial quantum state (`qutip.Qobj`) and produces the final state as a (`qutip.Result`) object. As discussed further below in this section, the key improvements in the new `qutip-qip` package are the `Model`, `GateCompiler`, `Scheduler` and the `Noise` classes that allow a modular and flexible design of realistic quantum processors for simulations.

We illustrate our new framework here with an example simulating a 3-qubit Deutsch-Jozsa algorithm on a chain of spin qubits. We will work through this

example and explain briefly the workflow and all the main modules. We then describe each module in detail in the subsequent subsections. The simulation of a more complicated circuit, a 10-qubit QFT algorithm, is presented in Section 8.8.2.

In `qutip-qip`, a quantum circuit is represented by an instance of the `QubitCircuit` class. The following code defines a circuit of a 3-qubit Deutsch-Jozsa algorithm (see Figure 8.3a)[3]:

```
qc = QubitCircuit(3)
qc.add_gate("X", targets=2)
qc.add_gate("SNOT", targets=0)
qc.add_gate("SNOT", targets=1)
qc.add_gate("SNOT", targets=2)

# Oracle function f(x)
qc.add_gate(
    "CNOT", controls=0, targets=2)
qc.add_gate(
    "CNOT", controls=1, targets=2)

qc.add_gate("SNOT", targets=0)
qc.add_gate("SNOT", targets=1)
```

The Deutsch-Jozsa algorithm consists of an oracle constructed using two CNOT gates. The first two qubits in our circuit take a binary input and will be measured at the end while the last qubit is an ancillary qubit that stores the result of the oracle. The goal is to test if the oracle function is balanced or constant. A constant function returns all 0 or 1 for any input, while a balanced function returns 0 for half of the input combinations and 1 for the other half.

Among the four different classical inputs ($\{00, 01, 10, 11\}$), for half of them, the oracle returns 0 while for the other half it returns 1. Hence it is a balanced function and, without noise, the measurement of the first two qubits will never be both 0. One can run the gate-level simulation in the following way:

```
init_state = basis([2,2,2], [0,0,0])
final_state = qc.run(init_state)
```

where we first initialize the state as $|000\rangle$ using `qutip.basis` and then run the circuit simulation. By checking the final state, one will see that it has no overlap with $|000\rangle$ or $|001\rangle$.

The above simulation is at the gate level and is computed by matrix-vector products of the gate operators and the input quantum state. We now describe how to simulate the circuit at the pulse level using `Processor`.

8.4.1 Processor

The `Processor` class handles the routine of a pulse-level simulation. It first compiles the circuit into a pulse-level description and then simulates the time evolution of

[3]The code examples present in the main text and the Appendices are available at github.com/boxili/qutip-qip-paper. The code is compatible with `qutip-qip==0.2`.

the underlying physical system using QuTiP solvers. For different hardware models and compiling methods, the same circuit can be compiled into different pulses, as shown in Figures 8.3b to 8.3d. Because of different noise models, the final state also differs from that of the ideal gate-level simulation.

In the following, we choose the spin chain model as an example of the underlying physical system and give an overview of the simulation procedure. We start by initializing a specific type of processor, a subclass of `Processor` called `LinearSpinChain`:

```
processor = LinearSpinChain(
    num_qubits=3, sx=0.25)
```

where we provide the number of qubits and the $\hat{\sigma}_x$ drive strength 0.25MHz. The other parameters, such as the interaction strength, are set to be the default value. The decoherence noise can also be added by specifying the coherence times (T_1 and T_2) which we discuss hereafter.

By initializing this processor with the hardware parameters, a Hamiltonian model for a spin chain system is generated, including the drift and control Hamiltonians ($\hat{H}_\mathrm{d}$, $\hat{H}_\mathrm{c}$). The Hamiltonian model is represented by the `Model` class and is saved as an attribute of the initialized processor. We provide different predefined models and discuss them more in Section 8.4.2. In addition, the `Processor` can also hold simulation configurations such as whether to use a cubic spline interpolation for the pulse coefficients. Such configurations are not directly part of the model but nevertheless could be important for the pulse-level simulation.

Next, we provide the circuit to the processor through the method `load_circuit`:

```
processor.load_circuit(qc)
```

The processor will first decompose the gates in the circuit into native gates that can be implemented directly on the specified hardware model. Each gate in the circuit is then mapped to the control coefficients and driving Hamiltonians according to the `GateCompiler` defined for a specific model. A `Scheduler` is used to explore the possibility of executing several pulses in parallel. The compiler and scheduler classes will be explained in detail in Sections 8.4.3 and 8.4.4.

In addition to the standard compiler, optimal control algorithms in QuTiP can also be used to generate the pulses, which are implemented in `OptPulseProcessor` (Section 8.4.5).

With a pulse-level description of the circuit generated and saved in the processor, we can now run the simulation by

```
t_max = processor.get_full_tlist()[-1]
tlist = np.linspace(0, t_max, 300)
result = processor.run_state(
    init_state, tlist=tlist)
```

The `run_state` method first builds a Lindblad model including all the defined noise models (none in this example, but options are discussed below) and then calls a QuTiP solver to simulate the time evolution. One can pass solver parameters as

keyword arguments to the method, e.g., `tlist` (time sequence for intermediate results), `e_ops` (measurement observables) and `options` (solver options). In the example above, we record the intermediate state at the time steps given by `tlist`. The returned result is a `qutip.Result` object, which, depending on the solver options, contains the final state, intermediate states and the expectation value. This allows one to extract all information that the solvers in QuTiP provide.

As for the simulation of noise, simple decoherence noise can be included in the `Processor` by specifying T_1, T_2, e.g.,

```
LinearSpinChain(num_qubits=3, t2=30)
```

More advanced noise models can be represented by the `Noise` class and added with the method `Processor.add_noise`. The following code is an equivalent way of defining a T_2 noise:

```
processor.add_noise(
    RelaxationNoise(t2=30))
```

In general, the `Noise` class can be used to represent both decoherence and coherent noise sources. The former is defined by time-dependent or independent collapse operators and the latter by additional Hamiltonian terms in equation (8.2), with which distortion in the control coefficients or cross-talk can be represented. In particular, one can define noise that is correlated with the compiled ideal control coefficients through the `Pulse` class. They are explained in detail with examples in Sections 8.4.6 and 8.4.7.

Overall, the framework is designed in a modular way so that one can add custom Hamiltonian models, compilers and noise models. We describe in Section 8.4.8 how this can be done by defining new subclasses.

8.4.2 Model

The pulse-level simulation depends strongly on the modelling of the physical qubits. In the framework, the physical model is saved as an instance of the `Model` class in an initialized processor. A `Model` object contains the information regarding the specific quantum hardware, including the drift Hamiltonian that cannot be controlled, the available control Hamiltonians and possible noise in the system. A concrete physical model such as `SpinChainModel` is defined as a subclass of `Model`.

For convenience of use, a `Model` object is automatically generated while initializing a specific `Processor`, as in the example at the beginning of this section. To offer more flexibility, `qutip-qip` provides an equivalent way for the user to define a model and pass it to a `Processor` object, e.g.,

```
model = SpinChainModel(
    num_qubits=3, setup="circular", g=1)
processor = Processor(model=model)
```

One can inquire about the properties of a control Hamiltonian through

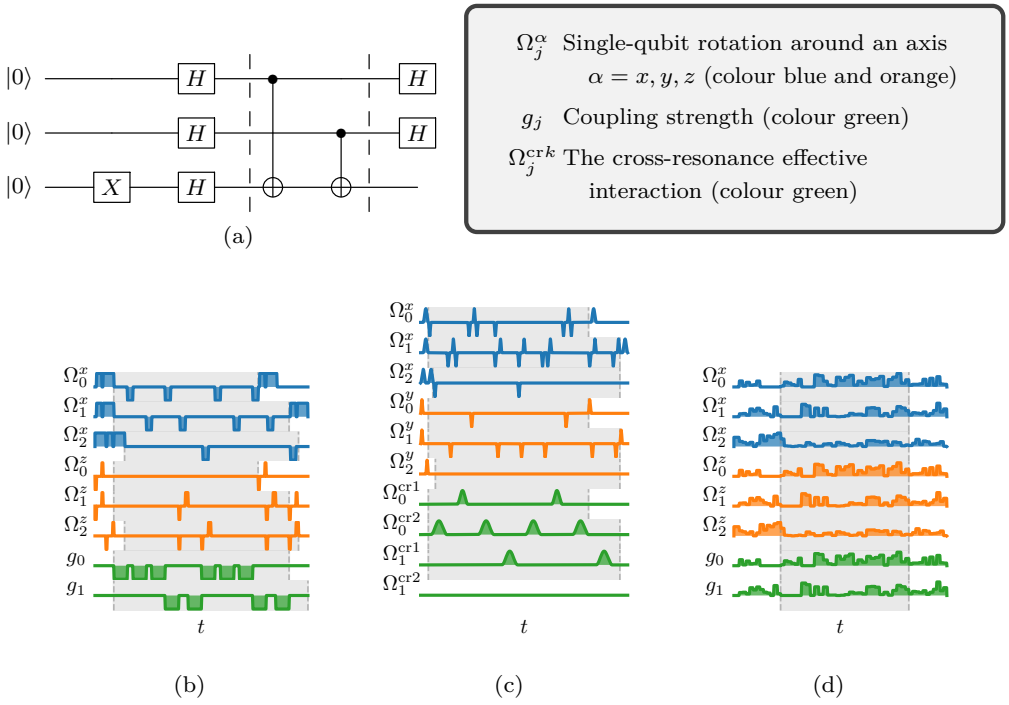


Figure 8.3: **(a).** Control pulses generated for a three-qubit Deutsch-Jozsa algorithm (a), where two CNOT gates implement the oracle, which is a balanced function. The pulses are compiled using (b) the spin chain model [equation (8.8)], (c) the superconducting qubits [equation (8.10)] and (d) the optimal control algorithm (using GRAPE with the same control Hamiltonian as the spin chain model in equation (8.8)). The symbols for pulse coefficients are defined in the corresponding equations. The blue and orange colours denote the two single-qubit control pulses, while green is used for the qubit-qubit interaction. For the spin chain and superconducting qubits, the interaction exists only between neighbouring qubits, hence SWAP gates are added to implement the CNOT between the first and third qubits and decomposed into the native gates. The grey background marks the pulse duration for the two CNOT gates, where the effect of ASAP scheduling is evident. The strength of the compiled pulses, $|c_j(t)|$, is normalized for plotting and should not be compared between different control Hamiltonians. Code examples generating these plots are shown in Section 8.8.1.

```
model.get_control(label="sx0")
```

which returns a tuple consisting of the Hamiltonian as a `qutip.Qobj` and the indices of the target qubits. For the predefined models, all available control Hamiltonians can be obtained by

```
model.get_control_labels()
```

The same interface is also provided in `Processor` (e.g., `Processor.get_control`) for convenience.

In predefined models, these control Hamiltonian terms are simply defined in a dictionary, equivalent to the following code:

```
controls = {}
for m in range(num_qubits):
    op = 2 * np.pi * sigmax()
    controls["sx"+str(m)] = (op, m)
```

which will be accessed by the model object. Notice that, in general, a model can be correctly recognized by the processor if the method `Model.get_control(label)` returns the results in the expected format, regardless of the internal implementation. For instance, in Section 8.8.3, we define it in a different way. This will be helpful, for instance, in an all-to-all connected system, e.g., using ions or neutral atoms, for which listing all the available combinations of target qubits is tedious.

Models allow one to simulate the physical qubits and their interaction in a more realistic way, e.g., using resonator-induced coupling and including leakage levels. This is demonstrated by a few predefined models that are implemented as subclasses of `Model`: the spin chain model, the qubits-resonator model and the fixed-frequency superconducting qubit model. Custom Hamiltonian models can be defined as subclasses as detailed in Sections 8.4.8 and 8.8.3. In the following, we illustrate the characteristics of the predefined hardware models in detail.

Spin Chain model

The spin-exchange interaction exists in many quantum systems and is one of the earliest types of interaction used in quantum information processing, e.g., in Refs. [237, 268, 269]. Our predefined `SpinChainModel` implements a system of a few spin qubits with the exchange interaction arranged in a one-dimensional chain layout with either open ends or closed ends.

The interaction is only possible between adjacent qubits. For the spin model, the single-qubit control Hamiltonians are $\hat{\sigma}_j^x$, $\hat{\sigma}_j^z$, while the interaction is realized by the exchange Hamiltonian $\hat{\sigma}_j^x\hat{\sigma}_{j+1}^x + \hat{\sigma}_j^y\hat{\sigma}_{j+1}^y$. The control Hamiltonian is given by

$$\hat{H} = \sum_{j=0}^{N-1} \Omega_j^x(t)\hat{\sigma}_j^x + \Omega_j^z(t)\hat{\sigma}_j^z + \sum_{j=0}^{N-2} g_j(t)(\hat{\sigma}_j^x\hat{\sigma}_{j+1}^x + \hat{\sigma}_j^y\hat{\sigma}_{j+1}^y), \tag{8.8}$$

where Ω^x, Ω^y and g are the time-dependent control coefficients and N is the number of qubits.

Qubit-resonator model

In some experimental implementations, interactions are realized by a quantum bus or a resonator connecting different qubits. The qubit-resonator model describes a system composed of a single resonator and a few qubits connected to it. The coupling is kept small so that the resonator is rarely excited but acts only as a mediator for entanglement generation. The single-qubit control Hamiltonians used are $\hat{\sigma}_x$ and $\hat{\sigma}_y$. The dynamics between the resonator and the qubits is captured by the Tavis-Cummings Hamiltonian, $\propto \sum_j \hat{a}^\dagger \hat{\sigma}_j^- + \hat{a}\hat{\sigma}_j^+$, where $\hat{a}$, $\hat{a}^\dagger$ are the destruction and creation operators of the resonator, while $\hat{\sigma}_j^-$, $\hat{\sigma}_j^+$ are those of each qubit. The control of the qubit-resonator coupling depends on the physical implementation, but in the most general case we have single and multi-qubit control in the form,

$$\hat{H} = \sum_{j=0}^{N-1} \Omega_j^x(t)\hat{\sigma}_j^x + \Omega_j^y(t)\hat{\sigma}_j^y + g_j(t)(\hat{a}^\dagger\hat{\sigma}_j^- + \hat{a}\hat{\sigma}_j^+) \,. \tag{8.9}$$

In the numerical simulation, the resonator Hamiltonian is truncated to finite levels. The user can find a predefined `CavityQEDModel` implementing equation (8.9).

Superconducting qubit model

Superconducting-circuit qubits have been harnessed to provide artificial atoms for quantum simulation and quantum computing [4, 5, 237, 270, 271]. In our model, defined by the `SCQubitsModel` class, each qubit is simulated by a multi-level Duffing model, in which the qubit subspace is provided by the ground state and the first excited state. By default, the creation and annihilation operators are truncated at the third level, which can be adjusted, if desired, by the user. The multi-level representation can capture the leakage of the population out of the qubit subspace during single-qubit gates. The single-qubit control is generated by two orthogonal quadratures $(\hat{a}_j^\dagger + \hat{a}_j)$ and $i(\hat{a}_j^\dagger - \hat{a}_j)$. Similar to the spin chain model, the interaction is possible only between adjacent qubits. Although this interaction is mediated by a resonator, for simplicity, we replace the complicated dynamics among two superconducting qubits and the resonator with a two-qubit effective Hamiltonian derived in [80].

As an example, we choose the cross resonance interaction in the form of $\hat{\sigma}_j^z\hat{\sigma}_{j+1}^x$, acting only on the two-qubit levels, which is widely used, e.g., in fixed-frequency superconducting qubits. We can write the Hamiltonian as

$$\hat{H} = \hat{H}_\mathrm{d} + \sum_{j=0}^{N-1} \Omega_j^x(\hat{a}_j^\dagger + \hat{a}_j) + \Omega_j^y i(\hat{a}_j^\dagger - \hat{a}_j) + \sum_{j=0}^{N-2} \Omega_j^{\mathrm{cr1}}\hat{\sigma}_j^z\hat{\sigma}_{j+1}^x + \Omega_j^{\mathrm{cr2}}\hat{\sigma}_j^x\hat{\sigma}_{j+1}^z, \tag{8.10}$$

where the drift Hamiltonian $\hat{H}_\mathrm{d}$ is defined by the anharmonicity α_j of the second excited state,

$$\hat{H}_\mathrm{d} = \sum_{j=0}^{N-1} \frac{\alpha_j}{2} \hat{a}_j^\dagger \hat{a}_j^\dagger \hat{a}_j \hat{a}_j. \tag{8.11}$$

The coefficients $\Omega^{\mathrm{cr}1}$ and $\Omega^{\mathrm{cr}2}$ are computed from the qubit-resonator detuning and coupling strength [80]. With additional single-qubit gates, a CNOT gate can be realized using this type of interaction [123]. Using this effective Hamiltonian significantly reduces the size of the Hilbert space in the simulation and allows us to include more qubits. This flexibility in choosing different levels of detail in the modelling is one of the biggest advantages of this framework, in particular for noise simulation (as illustrated in more detail in Section 8.4.6).

8.4.3 Compiler

A compiler converts the quantum circuit to the corresponding pulse-level controls $c_j(t)\hat{H}_j$ on the quantum hardware. In the framework, it is defined as an instance of the `GateCompiler` class. The compilation procedure is achieved through the following steps.

First, each quantum gate is decomposed into the native gates (e.g., rotation over x, y axes and the CNOT gate), using the existing decomposition scheme in QuTiP. If a gate acts on two qubits that are not physically connected, like in the chain model and superconducting qubit model, SWAP gates are added to match the topology before the decomposition. Currently, only 1-dimensional chain structures are supported.

Next, the compiler maps each quantum gate to a pulse-level control description. It takes the hardware parameter defined in the Hamiltonian model and computes the pulse duration and strength to implement the gate. For continuous pulses, the pulse shape can also be specified using SciPy window functions (`scipy.signal.windows`). A pulse scheduler is then used to explore the possibility of executing multiple quantum gates in parallel, which is explained in detail in Section 8.4.4.

In the end, the compiler returns a time-dependent pulse coefficient $c_j(t)$ for each control Hamiltonian $\hat{H}_j$ [see equation (8.2)]. They contain the full information to implement the circuit and are saved in the processor. The coefficient $c_j(t)$ is represented by two NumPy arrays, one for the control amplitude and the other for the time sequence. For a continuous pulse, a cubic spline is used to approximate the coefficient. This allows the use of compiled Cython code in QuTiP to achieve better performance.

For the predefined physical models described in the previous subsection, the corresponding compilers are also included and they will be used when calling the method `Processor.load_circuit`. As an example, we compile the three-qubit Deutsch-Jozsa algorithm, shown in Figure 8.3a, while the compiled pulses on three different models are plotted in Figures 8.3b to 8.3d. From the plots, it is evident that the same circuit is compiled to completely different pulse-level controls:

- For the spin chain model (Figure 8.3b), SWAP gates are added between and after the first CNOT gate, swapping the first two qubits (coefficient g_0). The SWAP gate is decomposed into three iSWAP gates, while the CNOT is decomposed into two iSWAP gates plus additional single-qubit corrections. Both the Hadamard gate and the two-qubit gates need to be decomposed to native gates (iSWAP and rotation on the x and z axes). The compiled coefficients are square pulses and the control coefficients on $\hat{\sigma}_z$ and $\hat{\sigma}_x$ are also different, resulting in different gate times.

- For the superconducting-qubit processor (Figure 8.3c), the compiled pulses have a Gaussian shape. This is crucial for superconducting qubits because the second excited level is only slightly detuned from the qubit transition energy. A smooth pulse usually prevents leakage to the non-computational subspace. Similar to the spin chain, SWAP gates are added to switch the zeroth and first qubit and one SWAP gate is compiled to three CNOT gates. The control $\Omega_1^{\mathrm{cr}2}$ [defined in equation (8.10)] is not used because there is no CNOT gate that is controlled by the second qubit and acts on the first one.

- For the optimal control model (Figure 8.3d), we use the GRAPE algorithm, where control pulses are piece-wise constant functions. We provide the algorithm with the same control Hamiltonian model used for the spin chain model, Eq. (8.8). In the compiled optimal signals, all controls are active (non-zero pulse amplitude) during most of the execution time. We note that for identical gates on different qubits (e.g., Hadamard), each optimized pulse is different, demonstrating that the optimized solution is not unique, and there are further constraints one could apply, such as adaptions for the specific hardware.

As a demonstration of the capability of the simulator, we also compile a 10-qubit QFT algorithm using `LinearSpinChain`, as shown in Section 8.8.2.

To end this subsection, we mention that the gate decomposition is not fully optimized in QuTiP. Circuit optimization at the level of quantum gates, such as for an optimal number of two-qubit gates, depends on the hardware of interest and is still an open research topic [272–275]. The same holds for mapping the circuit to the topology of the qubits' connectivity [276–278]. Because the focus of this simulator is the simulation of the circuit at the physics level, we leave more advanced optimization and scheduling techniques at the gate level for future work. Instead, we offer the possibility to import quantum circuits defined in other libraries into QuTiP in the QASM format (see Section 8.5). This allows possible optimizations elsewhere and then exporting the optimized circuits in QuTiP for a pulse-level simulation.

8.4.4 Scheduler

The scheduling of a circuit consists of an important part of the compilation. Without it, the gates will be executed one by one and many qubits will be idling during the

circuit execution, which increases the execution time and reduces the fidelity. In the framework, the scheduler is used after the control coefficient of each gate is computed. It runs a scheduling algorithm to determine the starting time of each gate while keeping the result correct.

The heuristic scheduling algorithm we provide offers two different modes: ASAP (as soon as possible) and ALAP (as late as possible). In addition, one can choose whether permutation among commuting gates is allowed to achieve a shorter execution time. The scheduler implemented here does not take the hardware architecture into consideration and assumes that the connectivity in the provided circuit matches with the hardware at this step.

In predefined processors, the scheduler runs automatically when loading a circuit and hence there is no action necessary from the side of the user. To help explain the scheduling algorithm, we provide here two examples of directly using the `Scheduler` class.

For gate scheduling, one can use

```
Scheduler("ASAP").schedule(qc)
```

which, for the 3-qubit Deutsch-Jozsa example (Figure 8.3a), returns a list

```
[0, 0, 0, 1, 2, 3, 3, 4]
```

This list denotes the gate cycle of each gate in the circuit. Here, all gates are assumed to have the same duration. One can see that, e.g., the second CNOT and the last Hadamard on the first qubit are grouped together in cycle 3.

For pulse scheduling, one needs to use the `Instruction` class, which includes information about a specific implementation of a gate on the hardware, e.g., the duration of a gate. If we assume that all single-qubit gates take a time duration of 1 unit while the CNOT takes a time duration of 2 units, we can rewrite it as

```
inst_list = []
for gate in qc.gates:
    if gate.name in ("SNOT", "X"):
        inst_list.append(
            Instruction(gate, duration=1
            )
        )
    else:
        inst_list.append(
            Instruction(gate, duration=2
            )
        )
Scheduler("ALAP").schedule(inst_list)
```

Notice that now we use the ALAP scheduling. This returns a different list

```
[0, 3, 1, 1, 4, 2, 6, 6]
```

with the starting time of each gate. In this result, the two CNOT gates (starting time 4 and 2) are exchanged, so that the first Hadamard on the zeroth qubit only needs to start at time step 3.

In the following, we describe our implementation of the pulse scheduler. The implementation is similar to Ref. [279, 280]. However, we omit the hardware-dependent part but allow gates to have different durations, generalizing it to a pulse scheduler. We focus on the ASAP scheduling while for the ALAP mode, the circuit is reversed before it is passed to the algorithm and then reversed back after the scheduling.

We first represent the dependency among quantum gates in a quantum circuit as a directed acyclic graph. Each gate is represented by a node and the dependency by arrows. Gate A is considered dependent on gate B if A has to be executed after B. This also means that A needs to be executed after all the gates that B depends on. Hence, there is no loop in the graph. Next, all gates are divided into different cycles (ignoring the gate duration) according to the dependency graph. A priority is then assigned to each quantum gate, determined by the time required to execute all the gates that depend on it. The more time it takes to execute the gates after it, the higher priority is assigned to this gate. In the end, from the dependency graph and the priority, a list-scheduling algorithm is used to determine the order of the execution and the starting time of each operation.

Unlike scheduling classical gates, a scheduler of quantum gates needs to take the commutation relation into account. For instance, if two CNOT gates are controlled by the same qubit, but act on two different target qubits, they can be exchanged. Exploring this flexibility may reduce the total execution time, as shown in the example above. This is included in the process of building the dependency graph. All commuting gates are added to the same cycle when computing the priority and the one with the highest priority will be executed first. In general, more advanced techniques need to be applied to optimize the commuting gates, for instance as discussed in Ref. [280]. However, this becomes more complicated when gates have different execution times. For simplicity, we omit these advanced techniques in this implementation.

8.4.5 Optimal control

Apart from using compilers with predefined gate-to-pulse maps, one can also use the optimal control algorithm in QuTiP to find optimized control pulses. The algorithm can take arbitrary control Hamiltonians as input and uses quantum control function optimisation, based on open-loop quantum control theory [281] to find the best pulses. For a set of given control Hamiltonians $\hat{H}_j$, the optimal control module uses classical algorithms to optimize the control function $c_j(t)$ in equation (8.2). Parameters of control pulses for realizing individual gates, sequences and hence complete circuits, are generated automatically through multi-variable optimization targeting maximum fidelity with the evolution described by the circuit.

The optimal control module in QuTiP supports both the GRAPE [282, 283] and the CRAB algorithms [30, 284]. The interface to use these algorithms in `qutip-qip` is implemented via the `OptPulseProcessor` class. One first provides the available control Hamiltonians that characterize the physical controls on the system, which, e.g., can be provided as an instance of the `Model` class, such as the `SpinChainModel`.

Upon loading the quantum circuit, each quantum gate is expanded to a unitary acting on the full Hilbert space and passed to the optimal control algorithm as the desired target. The returned pulses that drive this are concatenated to complete a full circuit simulation of the physical control sequences. An example of optimized pulses is shown in Figure 8.3d and the code can be found in Section 8.8.1.

8.4.6 Noise

The noise module allows one to add control and decoherence noise following the Lindblad description of open quantum systems [equation (8.5)]. Compared to the gate-based simulator (Section 8.3.1), this provides a more practical and straightforward way to describe the noise. In the current framework, noise can be added at different layers of the simulation, allowing one to focus on the dynamics of the dominant noise, while representing other noise, such as single-qubit relaxation, as collapse operators for efficiency. Depending on the problem studied, one can devote the computing resources to the most relevant type of noise.

Apart from imperfections in the Hamiltonian model and circuit compilation, the `Noise` class in the current framework defines deviations of the real physical dynamics from the compiled one. It takes the compiled pulse-level description of the circuit (see also Section 8.4.7) and adds noise elements to it, which allows defining noise that is correlated to the compiled pulses. In the following, we detail the three different noise models already available in the current framework.

Noise in the hardware model. The Hamiltonian model defined in the `Model` class may contain intrinsic imperfections of the system and hence the compiled ideal pulse does not implement the ideal unitary gate. Therefore, building a realistic Hamiltonian model usually already introduces noise to the simulation. An example is the superconducting-qubit processor model (Section 8.4.2), where the physical qubit is represented by a multi-level system. Since the second excitation level is only weakly detuned from the qubit transition frequency, the population may leak out of the qubit subspace. Another example is an always-on ZZ-type cross-talk induced by interaction with higher levels of the physical qubits [107], which is also implemented for the superconducting qubit model.

Control noise. The control noise, as the name suggests, arises from imperfect control of the quantum system, such as distortion in the pulse amplitude or frequency drift. The simplest example is the random amplitude noise on the control coefficient $c_j(t)$ in equation (8.2).

As a demonstration of control noise, we simulate classical cross-talk-induced decoherence between two neighbouring ion trap qubits described in [285]. We build a two-qubit `Processor`, where the second qubit is detuned from the first one by $\delta = 1.852$ MHz. A sequence of π-pulses with Rabi frequency of $\Omega = 20$ KHz and random phases are applied to the first qubit. We define noise such that the same pulse also applies to the second qubit. Because of the detuning, this pulse

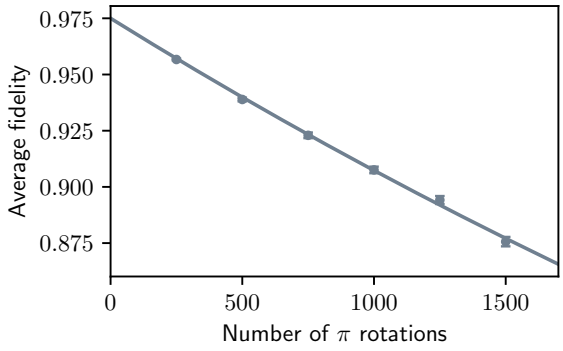


Figure 8.4: An example of simulated classical cross-talk-induced decoherence between neighbouring qubits in an ion trap system. The randomized benchmarking protocol is adopted from Piltz *et al.* [285] and the figure reproduces the measured fidelity decay in Figure 3a of that work. We build a custom `Processor` and `Noise` object to define classical cross-talk noise and perform our simulations. It shows the average fidelity of the qubit when a sequence of single-qubit π rotations with random phase is applied to its direct neighbour. The cross-talk is simulated by adding control pulses to the neighbouring qubits with a strength proportional to that of the target qubit and detuned by the difference of the qubit transition frequency. Each point is sampled from 1600 repetitions. We set the detuning $\delta = 1.852$ MHz, the Rabi frequency $\Omega = 20$ KHz and the cross-talk ratio $\lambda = 1$.

does not flip the second qubit but subjects it to a diffusive behaviour, so that the average fidelity of the second qubit with respect to the initial state decreases. This decreasing fidelity is shown experimentally in Figure 3a of Ref. [285].

Here, we reproduce these results with our two-qubit `Processor` in Figure 8.4. We start with an initial state of fidelity 0.975 and simulate the Hamiltonian

$$\hat{H} = \Omega(t)(\hat{\sigma}_0^x + \lambda\hat{\sigma}_1^x) + \delta\hat{\sigma}_1^z, \tag{8.12}$$

where λ is the ratio between the cross-talk pulse's amplitudes. The plot in Figure 8.4 shows a similar fidelity decay curve as the experimental result, but includes only the contribution of cross-talk, while in the experimental result, other noise sources may exist. This kind of simulation provides a way to identify noise contributions from different sources. The code is described in detail in Section 8.8.3, as an example of a custom noise model.

Lindblad noise. The Lindblad noise originates from the coupling of the quantum system with the environment (e.g., a thermal bath) and leads to loss of information. It is simulated by collapse operators and results in non-unitary dynamics [43, 264].

The most commonly used type of Lindblad noise is decoherence, characterized by the coherence time T_1 and T_2 (dephasing). For the sake of convenience, one

only needs to provide the parameter `t1`, `t2` to the processor and the corresponding operators will be generated automatically. Both can be either a number that specifies one coherence time for all qubits or a list of numbers, each corresponding to one qubit.

For T_1, the operator is defined as $a/\sqrt{T_1}$ with a as the destruction operator. For T_2, the operator is defined as $a^\dagger a\sqrt{2/T_2^*}$, where T_2^* is the pure dephasing time given by $1/T_2^* = 1/T_2 - 1/(2T_1)$. In the case of qubits, i.e., a two-level system, the destruction operator a is truncated to a two-level operator and is consistent with equation (8.5). Constant T_1 and T_2 can be provided directly when initializing the `Processor`. Custom collapse operators, including time-dependent ones, can be defined through `DecoherenceNoise`. For instance, the following code defines a collapse operator using `qutip.sigmam()` and increases linearly as time:

```
tlist = np.linspace(0, 30., 100)
coeff = tlist * 0.01
noise = DecoherenceNoise(
    sigmam(), targets=0,
    coeff=coeff, tlist=tlist)
proc.add_noise(noise)
```

Similar to the control noise, the Lindblad noise can also depend on the control coefficient.

In order to demonstrate the simulation of decoherence noise, we build an example that simulates a Ramsey experiment as a quantum circuit run on a noisy `Processor`. The Ramsey experiment consists of a qubit that is initialized in the excited state, undergoes a $\pi/2$ rotation around the x axis, idles for a time t, and is finally measured after another $\pi/2$ rotation:

```
amp = 0.1
f = 0.5
t2 = 10 / f

# Define a processor.
proc = LinearSpinChain(
    num_qubits=1, sx=amp/2, t2=t2)
ham_idle = 2*pi * sigmaz()/2 * f
resonant_sx = 2*pi * sigmax() - \
    ham_idle / (amp/2)
proc.add_drift(ham_idle, targets=0)
proc.add_control(
    resonant_sx, targets=0, label="sx0")

# Define a Ramsey experiment.
def ramsey(t, proc):
    qc = QubitCircuit(1)
    qc.add_gate("RX", 0, arg_value=pi/2)
    qc.add_gate("IDLE", 0, arg_value=t)
    qc.add_gate("RX", 0, arg_value=pi/2)
    proc.load_circuit(qc)
    result = proc.run_state(
```

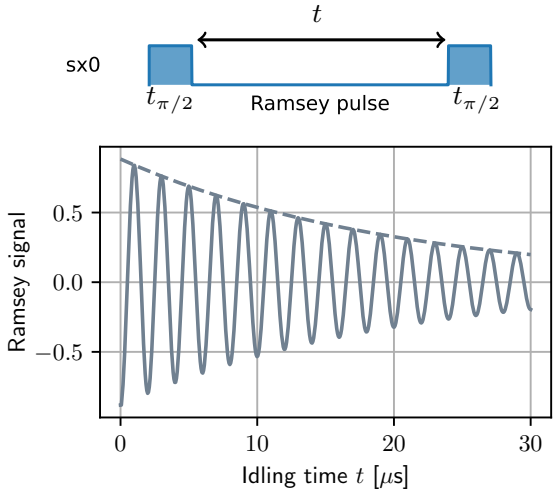


Figure 8.5: The Ramsey pulse and the simulated measurement results. The quantum system is subjected to a rotation around the z axis and a T_2 decoherence. The Ramsey pulse consists of two $\pi/2$ rotations separated by an idling time t. The expectation value of the measurement for different idling times is recorded. The solid line represents the measured expectation value. The dashed line is the fitted exponential decay. Due to the imperfect preparation of the superposed state, the envelope does not start from one.

```
        init_state=basis(2, 0),
        e_ops = sigmaz()
    )
    return result.expect[0][-1]
```

In the above block, we use the linear spin chain processor just for its compiler and do not use any of its default Hamiltonians. Instead, we define an always-on drift Hamiltonian $\hat{\sigma}^z$ with frequency $f = 0.5$ MHz, an on-resonant $\hat{\sigma}^x$ drive with an amplitude of $0.1/2$ MHz and the coherence time $T_2 = 10/f$. For different idling times t, we record the expectation value with respect to the observable $\hat{\sigma}^z$, which is plotted in Figure 8.5 as the solid curve. As expected, the envelope follows an exponential decay characterized by T_2 (dashed curve). Notice that, because $\pi/2$-pulses are simulated as a physical process, the fitted decay does not start from 1. This demonstrates a way to include state preparation errors in the simulation.

8.4.7 Pulse

As discussed before, in this simulation framework, we compile the circuit into pulse-level controls $c_j(t)\hat{H}_j$ [equation (8.2)] and simulate the time evolution of the physical qubits. In this subsection, we describe how the dynamics is represented internally in the workflow of `qutip-qip`, which is useful for understanding the

simulation process as well as defining custom pulse-dependent noise.

A control pulse, together with the noise associated with it, is represented by a class instance of `Pulse`. When an ideal control is compiled and returned to the processor, it is saved as an initialized `Pulse` object, equivalent to the following code:

```
coeff = np.array([1.])
tlist = np.array([0., np.pi])
pulse = Pulse(
    sigmax()/2, targets=0, tlist=tlist,
    coeff=coeff, label="pi-pulse")
```

This code defines a π-pulse implemented using the term $\hat{\sigma}_x$ in the Hamiltonian that flips the zeroth qubit specified by the argument `targets`. The pulse needs to be applied for the duration π specified by the variable `tlist`. The parameters `coeff` and `tlist` together describe the control coefficient $c(t)$. Together with the provided Hamiltonian and target qubits, an instance of `Pulse` determines the dynamics of one control term.

With a `Pulse` initialized with the ideal control, one can define several types of noise, including the Lindblad or control noise as described in Section 8.4.6. An example of adding a noisy Hamiltonian as control noise through the method `add_control_noise` is given below:

```
pulse.add_control_noise(
    sigmaz(), targets=[0], tlist=tlist,
    coeff=coeff * 0.05)
```

The above code snippet adds a Hamiltonian term $\hat{\sigma}_z$, which can, for instance, be interpreted as a frequency drift. Similarly, collapse operators depending on a specific control pulse can be added by the method `Pulse.add_lindblad_noise`.

In addition to a constant pulse, the control pulse and noise can also be provided as continuous functions. In this case, both `tlist` and `coeff` are given as NumPy arrays and a cubic spline is used to interpolate the continuous pulse coefficient. This allows using the compiled Cython version of the QuTiP solvers that have a much better performance than using a Python function for the coefficient. The option is provided as a keyword argument `spline_kind="cubic"` when initializing `Pulse`. Similarly, the interpolation method can also be defined for `Processor` using the same signature.

8.4.8 Adding custom hardware models

As it is impractical to include every physical platform, we provide an interface that allows one to customize the simulators. In particular, the modular architecture allows one to conveniently overwrite existing modules for customization.

To define a customized hardware model, the minimal requirements are a set of available control Hamiltonians $\hat{H}_j$, and a compiler, i.e., the mapping between native gates and control coefficients c_j. One can either modify an existing subclass

or write one from scratch by creating a subclass of the two parent classes `Model` and `GateCompiler`. Since different subclasses share the same interface, different models and compilers can also be combined to build new processors.

Moreover, this customization is not limited to Hamiltonian models and compiler routines. In principle, measurement can be defined as a customized quantum gate and the measurement statistics can be extracted from the obtained density matrix. A new type of noise can also be implemented by defining a new `Noise` subclass, which takes the compiled ideal `Pulse` and adds noisy dynamics on top of it.

An example of building a customized `Model` and `GateCompiler`, with custom types of noise, is provided in Section 8.8.3.

8.5 Importing and exporting circuits in QASM format

As pointed out in Section 8.4.3, it is impractical to include all the advanced techniques for circuit optimization and scheduling. To allow integration with other packages, we support the import and export of circuits in the intermediate Quantum Assembly Language (QASM) format [261]. While there are different intermediate representations for quantum programs, and more specifically quantum circuits, including cQASM [286], `qutip-qip` provides support for OpenQASM. OpenQASM is an imperative programming language that can be used to describe quantum circuits in a back-end agnostic manner.

QuTiP includes a module to import and export quantum circuits compatible with the OpenQASM 2.0 standard [261]. OpenQASM 2.0 allows concise quantum circuit definitions including useful features like custom unitaries and defining groups of qubits over which a common gate can be applied simultaneously. Due to compatibility with multiple libraries such as Qiskit and Cirq, it is an easy way to transfer quantum circuits between these libraries and `qutip-qip`.

As an example, we use again the 3-qubit Deutsch-Jozsa circuit (Figure 8.3a). The following block defines the same circuit in the QASM format:

```
OPENQASM 2.0;
include "qelib1.inc";

qreg q[3];
x q[2];
h q;
cx q[0], q[2];
cx q[1], q[2];
h q[0];
h q[1];
```

It can be saved as a `.qasm` file (such as `"deutsch-jozsa.qasm"` in our example below).

Every QASM file imported to `qutip-qip` requires the two header statements at the beginning of the file. The line `OPENQASM 2.0` declares that the file adheres to

the OpenQASM 2.0 standard. The keyword `include` processes a file that contains definitions of some QASM gates. It is available in the OpenQASM repository (as a standard file) and is included with the QASM file exported by `qutip-qip` (and also by Qiskit/Cirq). This circuit can be easily imported into `qutip-qip` using the `read_qasm` method in the following manner:

```
from qutip_qip.qasm import read_qasm
qc = read_qasm("deutsch-jozsa.qasm")
```

Furthermore, using the `strmode` option for `read_qasm` function, we can import the circuit described in a string object. Once a quantum circuit is defined, we can also export it to the QASM format and save it as a file using the `save_qasm` method:

```
from qutip_qip.qasm import save_qasm
save_qasm(qc,"deutsch-jozsa-qutip.qasm")
```

The circuit can then be simulated with other packages. It is also possible to output the circuit as a string using `circuit_to_qasm_str` or print it out using `print_qasm`.

8.6 Conclusion

In this work, we presented a framework for pulse-level quantum circuit simulation that can be used to study noisy quantum devices simulated on classical computers. This framework builds on existing solvers and the quantum circuit model offered by QuTiP. We expanded the noise modelling capabilities with ad-hoc features for the simulation of controls in noisy quantum circuits, such as providing the option to inject coherent noise in pulses.

We provided a few predefined quantum hardware models, compiling and scheduling routines, as well as noise models, which can be adjusted to devote limited computing resources to the most relevant physical dynamics during the study of noise. We showed the simulation capabilities by illustrating how results obtained on cross-talk noise characterization for an ion-trap-based quantum processor can be easily replicated with this toolbox. Moreover, we provided an example of the simulation of Lindblad noise for a Ramsey experiment.

Due to the modular design, the framework introduced here can be integrated with more hardware models, gate decomposition and optimization schemes. In particular, the simulation of processors supporting bosonic models for quantum information processing, including quantum error correction schemes, is especially suitable within the current framework. Represented as customized gates, state preparation and measurement can also be simulated as a noisy physical process.

Pulse-level simulation could be helpful in quick verification of experimental results, developing quantum algorithms, such as variational quantum algorithms [238, 287–289], and testing compiling and scheduling schemes [290] with realistic noise models [291]. Through hardware simulation and noise simulation, quantum error correction code and quantum mitigation protocols can also be

studied, for example, simulating pulse-level and digital zero-noise extrapolation [292–294].

Moreover, the noise characterization in model devices [295] and the impact of non-Markovian types of noise could be further evaluated [252, 296]. Future development in QuTiP aims at providing a unified interface to the open system solvers, which would enable a simpler integration with `qutip-qip`. This approach also has the potential to be integrated with other quantum control software such as `qupulse` [297] and `C3` [298]. In particular, the features here introduced may be a useful tool to investigate from a novel perspective many-body dynamical properties of quantum circuits, such as for measurement-induced phase transitions [299], chaotic dynamics and information scrambling [207].

Planned developments in `qutip` and `qutip-qip` will enable the use of alternative quantum control optimization algorithms, that is options other than the GRAPE and CRAB algorithms that are currently supported. Most immediately Krotov-type algorithm support could be added through integration with `qucontrol-krotov` [29], which is already closely aligned with QuTiP. Further opportunities for development and integration with the main QuTiP package include the development of an implementation of the GOAT algorithm [35], in which `qutip`'s solvers of various kinds can be used effectively. This could then also be available for optimization of circuit controls to simulate universal gate operations [300, 301].

Another direction of development is the integration with other software frameworks, in the ecosystem of quantum open source software, where considerable duplication exists. Even with respect to the quantum intermediate representation of quantum circuits, standards are not yet solidified. For example, we have connected `qutip-qip` with OpenQASM 2.0, thus providing an access point to any major framework. More sophisticated features are expected in the upcoming OpenQASM 3.0 standard [262], including classical computation specifications and the option for pulse-level definitions for gates. Extending QuTiP support to OpenQASM 3.0 will be an important step in cross-package compatibility with respect to pulse-level quantum circuit simulation and their integration with real hardware.

The use of `qutip-qip` for open quantum hardware is an especially intriguing direction of research and development. One could envision this framework as the backbone for API interconnectivity between simulation and hardware control in research labs with different technologies [297].

8.7 Data availability

The code examples present in the main text and the Appendices are available at github.com/boxili/qutip-qip-paper. A version compatible with the latest distribution of `qutip-qip` can be found at github.com/qutip/qutip-qip.

8.8 Appendix

8.8.1 Simulating the Deutsch-Jozsa algorithm

In this section, we show the code example of simulating the 3-qubit Deutsch-Jozsa algorithm on three different hardware models: the spin chain model, the superconducting qubits, and the optimal control model:

```
from qutip_qip.device import (
    OptPulseProcessor, LinearSpinChain, SCQubits, SpinChainModel)
from qutip_qip.circuit import QubitCircuit
from qutip import sigmaz, sigmax, identity, tensor, basis

# Deutsch-Josza algorithm
dj_circuit = QubitCircuit(num_qubits)
dj_circuit.add_gate("X", targets=2)
dj_circuit.add_gate("SNOT", targets=0)
dj_circuit.add_gate("SNOT", targets=1)
dj_circuit.add_gate("SNOT", targets=2)

# Oracle function f(x)
dj_circuit.add_gate("CNOT", controls=0, targets=2)
dj_circuit.add_gate("CNOT", controls=1, targets=2)

dj_circuit.add_gate("SNOT", targets=0)
dj_circuit.add_gate("SNOT", targets=1)

# Spin chain model
spinchain_processor = LinearSpinChain(num_qubits=num_qubits, t2=30)
                                      # T2 = 30
spinchain_processor.load_circuit(dj_circuit)
initial_state = basis([2, 2, 2], [0, 0, 0])  # 3 qubits in the 000
                                      state
t_record = np.linspace(0, 20, 300)
result1 = spinchain_processor.run_state(initial_state, tlist=t_record
                                      )

# Superconducting qubits
scqubits_processor = SCQubits(num_qubits=num_qubits)
scqubits_processor.load_circuit(dj_circuit)
initial_state = basis([3, 3, 3], [0, 0, 0])  #  3-level
result2 = scqubits_processor.run_state(initial_state)

# Optimal control model
setting_args = {"SNOT": {"num_tslots": 6, "evo_time": 2},
                "X": {"num_tslots": 1, "evo_time": 0.5},
                "CNOT": {"num_tslots": 12, "evo_time": 5}}
opt_processor = OptPulseProcessor(
    num_qubits=num_qubits, model=SpinChainModel(3, setup="linear"))
opt_processor.load_circuit(  # Provide parameters for the algorithm
    dj_circuit, setting_args=setting_args, merge_gates=False,
    verbose=True, amp_ubound=5, amp_lbound=0)
initial_state = basis([2, 2, 2], [0, 0, 0])
```

```
result3 = opt_processor.run_state(initial_state)
```

In the above code block, we first define the Deutsch-Jozsa algorithm, the same as the circuit shown in Figure 8.3. We then run the circuit on various hardware models. For the spin model and superconducting qubits, a Hamiltonian model and a compiler are already predefined and one only needs to load the circuit and run the simulation. Hardware parameters, such as the T_1 and T_2 times, qubit frequencies and coupling strength, can be given as parameters to initialize the processor. For optimal control, we use the control Hamiltonians of the spin chain model and provide a few parameters for the optimization routine in QuTiP, such as the maximal pulse amplitude and the number of time slots for each gate. For details, please refer to the QuTiP documentation (http://qutip.org/docs/latest/index.html).

The generated control pulses are shown in Figure 8.3 and can be obtained by the method:

```
processor.plot_pulses()
```

Because we are doing a simulation, we have access both to the final states as a density matrix and the information of the states during the evolution. We demonstrate this in Figure 8.6. By construction, the measured result of the first two qubits of a perfect Deutsch-Jozsa algorithm with a balanced oracle should not overlap with the state $|00\rangle$. This agrees with the small population of the state $|00\rangle$ in the Hinton diagram (Section 8.8.1). The population is not exactly zero because we define a T_2 decoherence noise. In addition, we can also extract information during the circuit execution, e.g., the population as a function of time (Section 8.8.1).

8.8.2 Compiling and simulating a 10-qubit Quantum Fourier Transform (QFT)

In this section, we simulate a 10-qubit Quantum Fourier Transform (QFT) algorithm. The QFT algorithm is one of the most important quantum algorithms in quantum computing [3]. It is, for instance, part of the Shor algorithm for integer factorization. The following code defines a 10-qubit QFT algorithm using CNOT and single-qubit rotations and runs the simulation both at the gate level and at the pulse level.

```
import numpy as np
from qutip import basis, fidelity
from qutip_qip.device import LinearSpinChain
from qutip_qip.algorithms import qft_gate_sequence

num_qubits = 10
# The QFT circuit
qc = qft_gate_sequence(num_qubits, swapping=False, to_cnot=True)
# Gate-level simulation
state1 = qc.run(basis([2]*num_qubits, [0]*num_qubits))
# Pulse-level simulation
processor = LinearSpinChain(num_qubits)
```

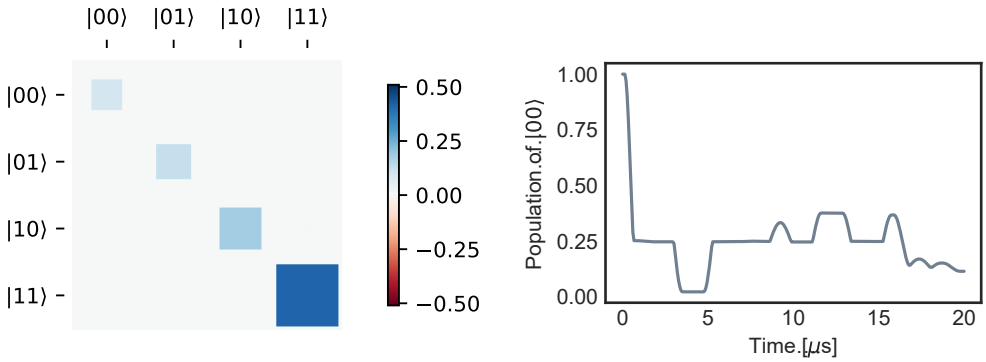


Figure 8.6: The Hinton diagram of the final density matrix (Section 8.8.1) and the population of the $|00\rangle$ state during the circuit execution (Section 8.8.1) for the first two qubits in the circuit shown in Figure 8.3. The Hinton diagram is a visual representation of the complex-valued density matrix. The shade and size of the blocks are determined by the absolute value of the density matrix element and the colour blue (red) denotes whether the real part of the density matrix is positive (negative). For an ideal Deutsch-Jozsa algorithm with a balanced oracle. The first two qubits should end up having no overlap with the ground state. This is not exactly the case in the plot because we define a finite T_2 time.

```
processor.load_circuit(qc)
state2 = processor.run_state(basis([2]*num_qubits, [0]*num_qubits)).
                                        states[-1]

assert(abs(1 - fidelity(state1, state2)) < 1.e-4)
```

We plot the compiled pulses and perform a study of the simulation time in the top and bottom panels of Figure 8.7, respectively. The top panel of Figure 8.7 shows the control pulses $\hat{\sigma}_x^i$ (blue curves), $\hat{\sigma}_y^i$ (orange curves) and g_i (green curves) for the spin chain model processor (Section 8.4.2), where $i = 0, ..., 9$ counts the qubits. The pulses plotted implement the QFT algorithm represented in the native gates of the spin chain model, with single-qubit gates marked by rotations over the x- and z-axes and the iSWAP gate implemented through the spin-spin exchange interaction, marked by g_i. While the sign for single-qubit drive denotes the phase of the control pulse, the negative sign in the coupling strengths g_i is only a result of the convention used in the definition of the interaction, defined in equation (8.8).

In the bottom panel of Figure 8.7, we study the time it takes to simulate the dynamics for the QFT algorithm on the spin chain processor, from 1 to 10 qubits. We divide the simulation between compilation and solution of the dynamical equation. The compilation of the algorithm (blue squares in the bottom panel of Figure 8.7) includes native-gate gate decomposition, scheduling, and mapping

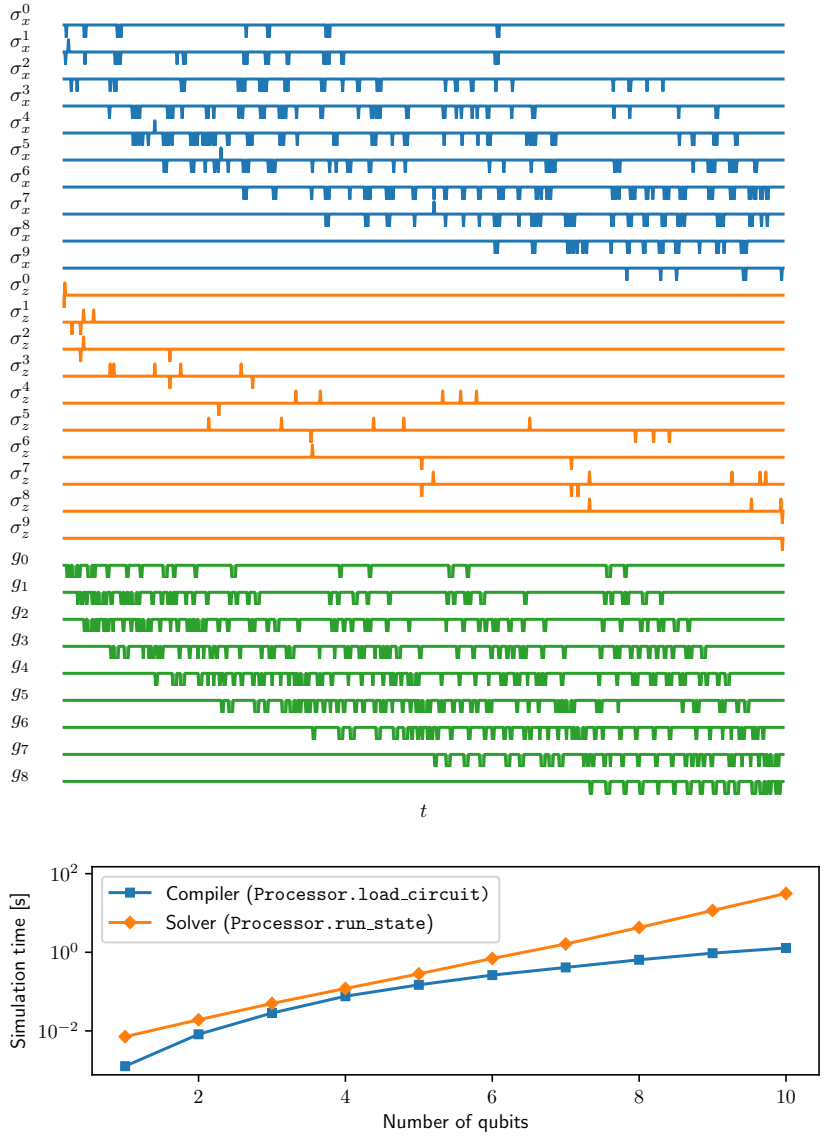


Figure 8.7: **Top:** Compiled pulses for a 10-qubit QFT circuit using the linear spin chain model (see Figure 8.3b and Section 8.4.2). The colours and notation used are the same as in Figure 8.3. The blue and orange colours denote the single-qubit control while the green colour the exchange interactions. **Bottom:** Simulation time of the QFT algorithm using the spin chain model as a function of the number of qubits, $N = 1, 2, ..., 10$, on a commercial CPU with a single thread. We plot both the compilation time (`Processor.load_circuit`) and the time used to solve the dynamics (`Processor.run_state`).

to control pulses (as shown in the top panel). For 10 qubits, the compilation takes about one second, whereas the overall simulation time takes about half a minute on a commercial CPU (Intel i7 8700 with Max Turbo Frequency 4.60 GHz) with a single thread. Indeed, the overall simulation time is dominated by the task of solving the Schrödinger equation: this increases linearly with the circuit depth and exponentially with the size of the Hilbert space (orange diamonds in the bottom panel of Figure 8.7). The proportion of time used for the compilation with respect to the total simulation time decreases as the number of qubits in the QFT algorithm grows. As expected, we find that the bottleneck for the simulation of larger processors lies in the solution of the dynamics.

Note that, because of the pulse-level nature of the simulation, the overall simulation time also depends on the typical frequency characterizing the dynamics. In the above simulation, the maximum frequency in the Hamiltonian is about 1 MHz while the time scale of the quantum circuit is about 2 ms. No collapse operators are included. The simulation time may increase if decay or high-frequency coherent noise are included.

8.8.3 Customizing the physical model and noise

In the following, we show a minimal example of constructing Hamiltonian models and compilers:

```
import numpy as np
from qutip import sigmax, sigmay, sigmaz, basis, qeye, tensor, Qobj,
                                    fock_dm
from qutip_qip.circuit import QubitCircuit, Gate
from qutip_qip.device import ModelProcessor, Model
from qutip_qip.compiler import GateCompiler, Instruction
from qutip import Options
from qutip_qip.noise import Noise


class MyModel(Model):
    """A custom Hamiltonian model with sigmax and sigmay control."""
    def get_control(self, label):
        """
        Get an available control Hamiltonian.
        For instance, sigmax control on the zeroth qubits is labeled
                                            "sx0".

        Args:
            label (str): The label of the Hamiltonian

        Returns:
            The Hamiltonian and target qubits as a tuple (qutip.Qobj,
                                                  list).
        """
        targets = int(label[2:])
        if label[:2] == "sx":
            return 2 * np.pi * sigmax() / 2, [targets]
```

```
        elif label[:2] == "sy":
            return 2 * np.pi * sigmay() / 2, [targets]
        else:
            raise NotImplementError("Unknown control.")

class MyCompiler(GateCompiler):
    """Custom compiler for generating pulses from gates using the
                                        base class
    GateCompiler.

    Args:
        num_qubits (int): The number of qubits in the processor
        params (dict): A dictionary of parameters for gate pulses
                                            such as
                       the pulse amplitude.
    """

    def __init__(self, num_qubits, params):
        super().__init__(num_qubits, params=params)
        self.params = params
        self.gate_compiler = {
            "ROT": self.rotation_with_phase_compiler,
            "RX": self.single_qubit_gate_compiler,
            "RY": self.single_qubit_gate_compiler,
        }

    def generate_pulse(self, gate, tlist, coeff, phase=0.0):
        """Generates the pulses.

        Args:
            gate (qutip_qip.circuit.Gate): A qutip Gate object.
            tlist (array): A list of times for the evolution.
            coeff (array): An array of coefficients for the gate
                                                pulses
            phase (float): The value of the phase for the gate.

        Returns:
            Instruction (qutip_qip.compiler.instruction.Instruction):
                                                An instruction
            to implement a gate containing the control pulses.
        """
        pulse_info = [
            # (control label, coeff)
            ("sx" + str(gate.targets[0]), np.cos(phase) * coeff),
            ("sy" + str(gate.targets[0]), np.sin(phase) * coeff),
        ]
        return [Instruction(gate, tlist=tlist, pulse_info=pulse_info)
                                            ]

    def single_qubit_gate_compiler(self, gate, args):
        """Compiles single-qubit gates to pulses.

        Args:
```

```
            gate (qutip_qip.circuit.Gate): A qutip Gate object.

        Returns:
            Instruction (qutip_qip.compiler.instruction.Instruction):
                                                An instruction
            to implement a gate containing the control pulses.
        """
        # gate.arg_value is the rotation angle
        tlist = np.abs(gate.arg_value) / self.params["pulse_amplitude
                                                "]
        coeff = self.params["pulse_amplitude"] * np.sign(gate.
                                                arg_value)
        if gate.name == "RX":
            return self.generate_pulse(gate, tlist, coeff, phase=0.0)
        elif gate.name == "RY":
            return self.generate_pulse(gate, tlist, coeff, phase=np.
                                                pi / 2)

    def rotation_with_phase_compiler(self, gate, args):
        """Compiles gates with a phase term.

        Args:
            gate (qutip_qip.circuit.Gate): A qutip Gate object.

        Returns:
            Instruction (qutip_qip.compiler.instruction.Instruction):
                                                An instruction
            to implement a gate containing the control pulses.
        """
        # gate.arg_value is the pulse phase
        tlist = self.params["duration"]
        coeff = self.params["pulse_amplitude"]
        return self.generate_pulse(gate, tlist, coeff, phase=gate.
                                                arg_value)


# Define a circuit and run the simulation
num_qubits = 1

circuit = QubitCircuit(1)
circuit.add_gate("RX", targets=0, arg_value=np.pi / 2)
circuit.add_gate("Z", targets=0)

myprocessor = ModelProcessor(model=MyModel(num_qubits))
myprocessor.native_gates = ["RX", "RY"]

mycompiler = MyCompiler(num_qubits, {"pulse_amplitude": 0.02})

myprocessor.load_circuit(circuit, compiler=mycompiler)
result = myprocessor.run_state(basis(2, 0))
```

In this example, we first build a Hamiltonian model called `MyModel`. For simplicity, we only include two single-qubit control Hamiltonians: $\hat{\sigma}_x$ and $\hat{\sigma}_y$. We

then define the compiling routines for the two types of rotation gates RX and RY. In addition, we also define a rotation gate with mixed X and Y quadrature, parameterized by a phase ϕ, $\cos(\phi)\hat{\sigma}_x + \sin(\phi)\hat{\sigma}_y$. This will be used later in the example of custom noise.

We then initialize a `ModelProcessor` with this model. In the `ModelProcessor`, the default simulation workflow is already defined, such as the `load_circuit` method. Since rotations around the x and y axes are the native gates of our hardware, we define them in the attribute `native_gates`. Providing this native gates set, rotation around z axis will be automatically decomposed into rotations around x and y axes. We define a circuit consisting of $\pi/2$ rotation followed by a Z gate. The compiled pulses are shown in Figure 8.8, where the Z gate is decomposed into rotations around x and y axes.

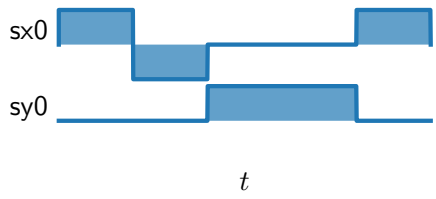


Figure 8.8: The compiled pulse of a $\pi/2$ pulse followed by a Z gate for the customized processor defined in Section 8.8.3. The Z gate is decomposed into rotations over the x and y axes.

Next, we show an example of defining customized noise and simulating classical cross-talk:

```
class ClassicalCrossTalk(Noise):
    def __init__(self, ratio):
        self.ratio = ratio

    def get_noisy_dynamics(self, dims=None, pulses=None,
                                        systematic_noise=None):
        """Adds noise to the control pulses.

        Args:
            dims: Dimension of the system, e.g., [2,2,2,...] for
                                                qubits.
            pulses: A list of Pulse objects, representing the
                                                compiled pulses.
            systematic_noise: A Pulse object with no ideal control,
                                                used to represent
            pulse-independent noise such as decoherence (not used in
                                                this example).
        Returns:
            pulses: The list of modified pulses according to the
                                                noise model.
```

```
            systematic_noise: A Pulse object (not used in this
                                                example).
        """
        for i, pulse in enumerate(pulses):
            if "sx" not in pulse.label and "sy" not in pulse.label:
                continue  # filter out other pulses, e.g. drift
            target = pulse.targets[0]
            if target != 0:  # add pulse to the left neighbour
                pulses[i].add_control_noise(
                    self.ratio * pulse.qobj,
                    targets=[target - 1],
                    coeff=pulse.coeff,
                    tlist=pulse.tlist,
                )
            if target != len(dims) - 1:  # add pulse to the right
                                                neighbour
                pulses[i].add_control_noise(
                    self.ratio * pulse.qobj,
                    targets=[target + 1],
                    coeff=pulse.coeff,
                    tlist=pulse.tlist,
                )
        return pulses, systematic_noise


def single_crosstalk_simulation(num_gates):
    """ A single simulation, with num_gates representing the number
                                        of rotations.

    Args:
        num_gates (int): The number of random gates to add in the
                                            simulation.

    Returns:
        result (qutip.solver.Result): A qutip Result object obtained
                                            from any of the
                                        solver methods such as mesolve.
    """
    num_qubits = 2  # Qubit-0 is the target qubit. Qubit-1 suffers
                                        from crosstalk.
    myprocessor = ModelProcessor(model=MyModel(num_qubits))
    # Add qubit frequency detuning 1.852MHz for the second qubit.
    myprocessor.add_drift(2 * np.pi * (sigmaz() + 1) / 2 * 1.852,
                                        targets=1)
    myprocessor.native_gates = None  # Remove the native gates
    mycompiler = MyCompiler(num_qubits, {"pulse_amplitude": 0.02, "
                                        duration": 25})
    myprocessor.add_noise(ClassicalCrossTalk(1.0))
    # Define a randome circuit.
    gates_set = [
        Gate("ROT", 0, arg_value=0),
        Gate("ROT", 0, arg_value=np.pi / 2),
        Gate("ROT", 0, arg_value=np.pi),
```

```
        Gate("ROT", 0, arg_value=np.pi / 2 * 3),
    ]
    circuit = QubitCircuit(num_qubits)
    for ind in np.random.randint(0, 4, num_gates):
        circuit.add_gate(gates_set[ind])
    # Simulate the circuit.
    myprocessor.load_circuit(circuit, compiler=mycompiler)
    init_state = tensor(
        [Qobj([[init_fid, 0], [0, 0.025]]), Qobj([[init_fid, 0], [0,
                                                  0.025]])]
    )
    options = Options(nsteps=10000)  # increase the maximal allowed
                                     steps
    e_ops = [tensor([qeye(2), fock_dm(2)])]  # observable

    # compute results of the run using a solver of choice with custom
                                     options
    result = myprocessor.run_state(init_state, solver="mesolve",
        options=options, e_ops=e_ops)
    result = result.expect[0][-1]  # measured expectation value at
                                     the end
    return result
```

In the code block above, we first define a custom `ClassicalCrossTalk` noise object that uses the `Noise` class as the base. The `get_noisy_dynamics` method will be called during the simulation to generate the noisy Hamiltonian model. Here, we define a noise model that adds the same driving Hamiltonian to its neighbouring qubits, with a strength proportional to the control pulse strength applied to it. The detuning of the qubit transition frequency is simulated by adding a $\hat{\sigma}_z$ drift Hamiltonian to the processor, with a frequency of 1.852 MHz.

Second, we define a random circuit consisting of a sequence of π rotation pulses with random phases. The driving pulse is a π pulse with a duration of 25 μs and Rabi frequency 20 KHz. As described in [285], this randomized benchmarking protocol allows one to study the classical cross-talk-induced decoherence on the neighbouring qubits. The two qubits are initialized in the $|00\rangle$ state with a fidelity of 0.975. After the circuit, we measure the population of the second qubit. If there is no cross-talk, it will remain perfectly in the ground state. However, cross-talk induces a diffusive behaviour of the second qubit and the fidelity decreases. This simulation is repeated 1600 times to obtain the average fidelity, as shown in Figure 8.4 in the main text.

9 Summary and Outlook

9.1 Summary and conclusions

In this thesis, we have developed and implemented analytical methods for the efficient modelling and control of quantum systems. As a specific physical platform, we focus on enhancing the precision and efficiency of quantum operations for superconducting qubits. Through rigorous theoretical development, comprehensive simulations, and experimental validation, this research has addressed several challenges associated with control errors in quantum operation. The thesis focuses on three primary areas: efficient system modelling, the design of robust control schemes, and performance benchmarking through simulation.

Rather than working with a fully dense Hamiltonian and attempting to implement arbitrary desired unitary operations, our study focuses on well-established physical systems that exhibit specific structures and symmetries. These distinctive characteristics have already positioned these systems as prime platforms for quantum information processing over the past two decades. Their Hamiltonian sparsity and inherent structure enable significant simplification in modelling and allow for the derivation of analytical methods.

The recursive diagonalization methods introduced in Chapter 3 are designed with this sparsity and structure in mind. With only a limited number of error sources relatively isolated from each other, the system's dynamics can be effectively addressed using perturbative techniques. This approach gives rise to the recursive Schrieffer-Wolff method, where perturbative diagonalization is systematically applied to eliminate unwanted Hamiltonian terms in layers: first removing direct couplings, then treating the obtained Hamiltonian as a new problem and addressing the cross terms resulting from higher-order commutators between the errors and the ideal dynamics, as well as among the error terms themselves. Additionally,

we also consider that some error sources, such as off-resonant leakage errors, may be substantial enough to break the perturbative assumption. In these cases, the Givens rotation method is employed. Although the Givens rotation DRAG is only exactly derived for two-level systems, recursively applying them to dominant error sources guides the system toward a diagonalized form.

By applying a recursive approach that simplifies complex problems step by step, this method offers an efficient tool for deriving analytical, closed-form expressions for effective Hamiltonians. Our results show that the NPAD method significantly simplifies the modelling and simulation of superconducting qubits, providing a reliable predictive framework for designing drive schemes and analyzing system behaviour.

This recursive formulation also lays out the foundation of the analytical control schemes presented in Chapters 5 to 7, where we study the dynamics of the quantum system. While introducing time dependence into the analysis significantly increases the complexity, this same time dependence can be leveraged to engineer more effective quantum operations. Following the principle of the DRAG techniques, pulse shapes can be carefully engineered to suppress specific transition errors. These techniques have been continuously refined to minimize qubit operation errors and leakage, which are common problems for superconducting qubits. We demonstrate that integrating the recursive approach with DRAG provides a practical approach for simultaneously addressing multiple errors.

In Chapter 5, the problem manifests itself as the population errors on a three-level Transmon under off-resonant drive. Here, our method extends the traditional application of DRAG in two aspects. First, while the native control term only couples the adjacent levels, strong drives generate two-photon terms that must also be addressed. The recursive application of DRAG frame transformations and corrections leads to an intuitive and efficient expression. Moreover, combining it with the nonperturbative diagonalization methods from Chapter 3 results in DRAG pulses that offer another order magnitude improvement compared to the perturbative version. Although this Givens rotation DRAG is derived exactly only in a two-level system, in many practical scenarios, the impact of cross-terms is small, making this method broadly applicable. This improvement remains robust even with the recursive chain of DRAG corrections.

In Chapters 6 and 7, the application is extended beyond qubit gates to qudit systems and crosstalk. The presence of multiple leakage levels in a qudit system naturally leads to the use of recursive DRAG expressions. Our study also reveals that the same model and control pulse Ansatz are universal and can be applied to drive any ladder operation in a nonlinear oscillator. Interestingly, the crosstalk problem, the unwanted excitation on the neighbouring qubits during a single-qubit drive, is also described by a similar model. Although the energy separation could be quite small in this case, the DRAG correction still demonstrates significant suppression of neighbouring qubit excitation.

The above applications still adhere to the conventional use of DRAG: defining an adiabatic frame and introducing DRAG correction to ensure that the adiabaticity is

kept at the end of the operation. This concept is generalized to correct dynamical errors that do not commute with the ideal dynamics in the same subspace when studying the two-qubit ZZ error in Chapter 5. By transforming the Hamiltonian into the eigenbasis of the ideal drive, we can design DRAG corrections within the eigenframe. Through this approach, a DRAG pulse can be tailored to convert a ZZ interaction term into the desired ZX interaction.

To further validate the practicality of these methods, we tested the proposed control schemes in experimental settings. Using an online platform with remote pulse-level access, we calibrate custom Cross-Resonance CNOT gates based on the proposed DRAG pulse on fixed frequency Transmon qubits, demonstrating a threefold reduction of the gate error in Chapter 5. Notably, in contrast to the most common applications on single-qubit Transmon gates, if all the dominant errors are considered in the model, the designed DRAG pulses only depend on the eigenstates and do not require fine-detuning. As another proof of practicality, the calibration and benchmarking are repeated on several qubit pairs with different frequencies, showing uniform improvement across different pairs. This flexibility is crucial for scaling up quantum chips, as each qubit pair may have different characteristics due to inhomogeneity in both design and fabrication.

Finally, to smooth the testing and verification of new control and calibration methods, a new software package, `qutip-qip`, was developed. Based on the QuTiP package, these tools incorporate realistic control errors and simulate quantum circuits at the level of time evolution. The package includes a full-stack workflow from the definition of quantum operations, the compilation and scheduling of control pulses and a framework for modelling quantum hardware under the Lindblad model. It offers an efficient framework for the design, testing, and implementation of quantum control techniques in realistic scenarios.

9.2 Future work

Scalable control methods for multi-qubit quantum chips In general, analytical techniques such as DRAG provide a framework for parameterized pulse shapes, taking advantage of a deep understanding of the dynamics of the quantum system. In this sense, it is more efficient than traditional pulse Ansatz, such as those based on Fourier or Sigmoid shapes, originating from classical signal engineering.

This thesis primarily focuses on DRAG methodologies and their application in single and two-qubit systems. However, as quantum computing architectures evolve towards larger scales, the complexities of interactions and the potential for errors expand exponentially. The recursive structure presented in this thesis offers a convenient way to integrate various corrections for leakage and crosstalk within a unified framework. Future research could explore extending analytical methods developed in this thesis to design and optimize error suppression techniques specifically for multi-qubit operations and larger quantum circuits, such as spectator qubits. Investigating the scalability of these methods could provide crucial insights into their practicality for large-scale quantum chips.

Control optimization on low-quality qubits In the pursuit of scalable quantum computing architectures, a significant challenge remains in the variability of qubits' quality across a chip. Due to the fabrication inhomogeneity, the property of each qubit cannot be precisely engineered. The variation can be tens to a hundred MHz [38, 39], posing a challenge for uniform system performance. Although the majority of the qubits fall within acceptable performance thresholds, the presence of those outliers leads to performance bottlenecks and limits the fabrication yield [121]. While enhancing the performance of the best-performing qubits, targeted control strategies should also be developed simultaneously for those weak nodes. By developing specialized control strategies aimed at compensating for the deficiencies of these qubits, it is possible to enhance the overall reliability and uniformity of solid-state quantum devices.

Automatic system characterization and control optimization In this thesis, we extensively explored analytical control methods, which play an important role in experimental calibration by directly linking control parameters to measurable physical properties, such as the DRAG correction strength to the leakage rate. These calibration routines are fundamentally model-based, relying on a priori knowledge of the system's characteristics and behaviour. Techniques such as error amplification circuits are carefully designed based on this strategy to optimize system performance [155, 302].

Recently, there has been significant interest in developing automatic system characterization techniques, commonly referred to as Hamiltonian learning [298, 303]. These methods automate the process of system analysis by using probe control pulses to interact with the system and analyzing the resultant measurement outcomes to infer model characteristics. However, a limitation of these approaches is their inefficiency. Often, the rate at which useful information is extracted from experimental data is frustratingly slow.

The future may see the emergence of novel techniques that integrate traditional analytical control methods with automatic system characterization. This hybrid approach would aim to maintain the efficiency of information acquisition in carefully designed error amplification techniques while leveraging the adaptability and automation offered by Hamiltonian learning. This approach could lead to more robust quantum control where control strategies evolve based on empirical data, reducing the reliance on initial system models and accommodating the inevitable drifts in system parameters over time. A practical strategy in the short term might involve establishing a calibration routine based on meticulously designed algorithms that capture the essential dynamics and errors of the quantum system. Then, machine-learning-based optimization methods could be used to iteratively adjust and fine-tune the drive parameters. By integrating both manual expertise in algorithm design and the adaptive capabilities of machine learning, future research could unlock more efficient and adaptable control strategies for quantum systems.

Multi-frequency control schemes The modelling and control methods detailed in this thesis are built under the rotating wave approximation, where the pulse is modulated by oscillations resonant with the qubit's frequency, typically several GHz. This frequency is significantly higher, more than tenfold, than the system's dynamic range, allowing for the simplification where counter-rotating terms, which occur at twice the frequency of the base oscillation, are typically neglected. This approximation may not hold for other superconducting qubits architectures such as Fluxonium [304, 305]. Moreover, recent advancements have introduced multi-frequency drive techniques that use multiple dominant frequencies to drive quantum system dynamics [143, 306, 307]. This requires a generalization of the current methods, in particular, accurately identifying and quantifying the contribution of each frequency component within the system, together with their cross-effect on each other.

Robust control against parameter drifts The practical deployment of quantum control methods crucially depends on the long-term stability of system parameters, which can be significantly impacted by parameter drifts over time [308, 309]. While robust control techniques have been developed to mitigate these effects, they often come at the cost of longer operation times to fulfil adiabatic conditions [310] The control methodologies discussed in this thesis, however, are specifically designed to accelerate quantum operations to outpace the effects of decoherence. This approach opens up the possibility of integrating these fast control methods with robust control techniques to create hybrid strategies that not only counteract immediate errors but also anticipate and compensate for slow environmental changes, which could enhance the operational lifetime and stability of quantum computing devices.

Application in other quantum systems While the thesis successfully applies NPAD and DRAG methodologies to superconducting qubits, their applicability to other types of quantum systems, such as trapped ions and Rydberg atoms, presents a vast area for further research. Some physical systems are distinct from superconducting qubits in that they exhibit lower system inhomogeneity because their intrinsic properties are defined by nature. Despite this advantage, similar crosstalk and leakage errors may occur as the operation infidelity is pushed towards 0.1%. Exploring these applications could not only broaden the impact of the developed methods but also contribute to the universal toolkit of quantum control strategies across different platforms.

References

[1] S. Wehner, D. Elkouss, and R. Hanson, *Quantum internet: A vision for the road ahead*, Science **362**, eaam9288 (2018).

[2] C. L. Degen, F. Reinhard, and P. Cappellaro, *Quantum sensing*, Reviews of Modern Physics **89**, 035002 (2017).

[3] M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information* (Cambridge University Press, 2000).

[4] X. Gu, A. F. Kockum, A. Miranowicz, Y.-x. Liu, and F. Nori, *Microwave photonics with superconducting quantum circuits*, Physics Reports **718–719**, 1–102 (2017).

[5] P. Krantz, M. Kjaergaard, F. Yan, T. P. Orlando, S. Gustavsson, and W. D. Oliver, *A quantum engineer's guide to superconducting qubits*, Applied Physics Reviews **6**, 021318 (2019).

[6] A. Blais, A. L. Grimsmo, S. M. Girvin, and A. Wallraff, *Circuit Quantum Electrodynamics*, Reviews of Modern Physics **93**, 025005 (2021), arXiv:2005.12667 [quant-ph] .

[7] P. V. Klimov, J. Kelly, Z. Chen, M. Neeley, A. Megrant, *et al.*, *Fluctuations of Energy-Relaxation Times in Superconducting Qubits*, Physical Review Letters **121**, 090502 (2018).

[8] D. Dong and I. R. Petersen, *Quantum control theory and applications: A survey*, IET Control Theory & Applications **4**, 2651–2671 (2010), arXiv:0910.2350 [math-ph, physics:quant-ph] .

[9] H. Mabuchi and N. Khaneja, *Principles and applications of control in quantum systems*, International Journal of Robust and Nonlinear Control **15**, 647–667 (2005).

[10] W. S. Warren, H. Rabitz, and M. Dahleh, *Coherent Control of Quantum Dynamics: The Dream Is Alive*, Science **259**, 1581–1589 (1993).

[11] H. Rabitz, R. de Vivie-Riedle, M. Motzkus, and K. Kompa, *Whither the Future of Controlling Quantum Phenomena?*, Science **288**, 824–828 (2000).

[12] L. M. K. Vandersypen and I. L. Chuang, *NMR techniques for quantum control and computation*, Reviews of Modern Physics **76**, 1037–1069 (2005).

[13] S. A. Rice, *New Ideas for Guiding the Evolution of a Quantum System*, Science **258**, 412–413 (1992).

[14] C. H. Bennett and G. Brassard, *Quantum cryptography: Public key distribution and coin tossing*, Theoretical Computer Science **560**, 7–11 (2014).

[15] C. H. Bennett, G. Brassard, C. Crépeau, R. Jozsa, A. Peres, and W. K. Wootters, *Teleporting an unknown quantum state via dual classical and Einstein-Podolsky-Rosen channels*, Physical Review Letters **70**, 1895–1899 (1993).

[16] D. P. DiVincenzo, *The Physical Implementation of Quantum Computation*, Fortschritte der Physik **48**, 771–783 (2000).

[17] G. M. Huang, T. J. Tarn, and J. W. Clark, *On the controllability of quantum-mechanical systems*, Journal of Mathematical Physics **24**, 2608–2618 (1983).

[18] R. Vijay, C. Macklin, D. H. Slichter, S. J. Weber, K. W. Murch, *et al.*, *Stabilizing Rabi oscillations in a superconducting qubit using quantum feedback*, Nature **490**, 77–80 (2012).

[19] D. Ristè and L. DiCarlo, Digital feedback in superconducting quantum circuits (2015), arXiv:1508.01385 [cond-mat, physics:quant-ph] .

[20] J. Zhang, Y.-x. Liu, R.-B. Wu, K. Jacobs, and F. Nori, *Quantum feedback: Theory, experiments, and applications*, Physics Reports Quantum Feedback: Theory, Experiments, and Applications, **679**, 1–60 (2017).

[21] F. K. Wilhelm, S. Kirchhoff, S. Machnes, N. Wittler, and D. Sugny, An introduction into optimal control for quantum technologies (2020), arXiv:2003.10132 [quant-ph] .

[22] S. Machnes, U. Sander, S. J. Glaser, P. de Fouquières, A. Gruslys, *et al.*, *Comparing, optimizing, and benchmarking quantum-control algorithms in a unifying programming framework*, Physical Review A **84**, 022305 (2011).

[23] P. Rembold, N. Oshnik, M. M. Müller, S. Montangero, T. Calarco, and E. Neu, *Introduction to quantum optimal control for quantum sensing with nitrogen-vacancy centers in diamond*, AVS Quantum Science **2**, 024701 (2020).

[24] N. Khaneja, T. Reiss, C. Kehlet, T. Schulte-Herbrüggen, and S. J. Glaser, *Optimal control of coupled spin dynamics: Design of NMR pulse sequences by gradient ascent algorithms*, Journal of Magnetic Resonance **172**, 296–305 (2005).

[25] B. C. Hall, *Lie Groups, Lie Algebras, and Representations*, Graduate Texts in Mathematics, Vol. 222 (Springer New York, New York, NY, 2003).

[26] M. Rossignolo, T. Reisser, A. Marshall, P. Rembold, A. Pagano, *et al.*, *QuOCS: The Quantum Optimal Control Suite*, Computer Physics Communications , 108782 (2023).

[27] J. Johansson, P. Nation, and F. Nori, *QuTiP 2: A Python framework for the dynamics of open quantum systems*, Computer Physics Communications **184**, 1234–1240 (2013).

[28] M. H. Goerz, QuantumControl.jl, `https://github.com/JuliaQuantumControl/QuantumControl.jl` (2023).

[29] M. Goerz, D. Basilewitsch, F. Gago-Encinas, M. G. Krauss, K. P. Horn, *et al.*, *Krotov: A Python implementation of Krotov's method for quantum optimal control*, SciPost Physics **7**, 080 (2019).

[30] P. Doria, T. Calarco, and S. Montangero, *Optimal Control Technique for Many-Body Quantum Dynamics*, Physical Review Letters **106**, 190501 (2011).

[31] T. Caneva, T. Calarco, and S. Montangero, *Chopped random-basis quantum optimization*, Physical Review A **84**, 022326 (2011).

[32] M. M. Müller, R. S. Said, F. Jelezko, T. Calarco, and S. Montangero, *One decade of quantum optimal control in the chopped random basis*, Reports on Progress in Physics **85**, 076001 (2022).

[33] J. A. Nelder and R. Mead, *A Simplex Method for Function Minimization*, The Computer Journal **7**, 308-313 (1965).

[34] N. Rach, M. M. Müller, T. Calarco, and S. Montangero, *Dressing the chopped-random-basis optimization: A bandwidth-limited access to the trap-free landscape*, Physical Review A **92**, 062343 (2015).

[35] S. Machnes, E. Assémat, D. Tannor, and F. K. Wilhelm, *Tunable, Flexible, and Efficient Optimization of Control Pulses for Practical Qubits*, Physical Review Letters **120**, 150401 (2018).

[36] R. T. Q. Chen, Y. Rubanova, J. Bettencourt, and D. K. Duvenaud, in *Advances in Neural Information Processing Systems*, Vol. 31 (Curran Associates, Inc., 2018).

[37] M. Goerz, *Optimizing Robust Quantum Gates in Open Quantum Systems*, Ph.D. thesis, Universität Kassel (2015).

[38] J. M. Kreikebaum, K. P. O'Brien, A. Morvan, and I. Siddiqi, *Improving wafer-scale Josephson junction resistance variation in superconducting quantum coherent circuits*, Superconductor Science and Technology **33**, 06LT02 (2020), arXiv:1909.09165 [cond-mat, physics:physics, physics:quant-ph] .

[39] E. J. Zhang, S. Srinivasan, N. Sundaresan, D. F. Bogorin, Y. Martin, *et al.*, *High-performance superconducting quantum processors via laser annealing of transmon qubits*, Science Advances **8**, eabi6690 (2022).

[40] J. Ghosh, A. G. Fowler, J. M. Martinis, and M. R. Geller, *Understanding the effects of leakage in superconducting quantum-error-detection circuits*, Physical Review A **88**, 062329 (2013).

[41] J. Koch, T. M. Yu, J. Gambetta, A. A. Houck, D. I. Schuster, *et al.*, *Charge-insensitive qubit design derived from the Cooper pair box*, Physical Review A **76**, 042319 (2007).

[42] H.-P. Breuer and F. Petruccione, *The Theory of Open Quantum Systems* (Oxford University Press, 2007).

[43] H.-P. Breuer and F. Petruccione, *The Theory of Open Quantum Systems* (Oxford University Press, Oxford ; New York, 2002).

[44] C. H. Bennett, G. Brassard, S. Popescu, B. Schumacher, J. A. Smolin, and W. K. Wootters, *Purification of Noisy Entanglement and Faithful Teleportation via Noisy Channels*, Physical Review Letters **76**, 722–725 (1996).

[45] D. A. Lidar and T. A. Brun, eds., *Quantum Error Correction* (Cambridge University Press, Cambridge, 2013).

[46] I. M. Georgescu, S. Ashhab, and F. Nori, *Quantum simulation*, Reviews of Modern Physics **86**, 153–185 (2014).

[47] T. Albash and D. A. Lidar, *Adiabatic quantum computation*, Reviews of Modern Physics **90**, 015002 (2018).

[48] X. Chen, I. Lizuain, A. Ruschhaupt, D. Guéry-Odelin, and J. G. Muga, *Shortcut to Adiabatic Passage in Two- and Three-Level Atoms*, Physical Review Letters **105**, 123003 (2010).

[49] D. Guéry-Odelin, A. Ruschhaupt, A. Kiely, E. Torrontegui, S. Martínez-Garaot, and J. G. Muga, *Shortcuts to adiabaticity: Concepts, methods, and applications*, Reviews of Modern Physics **91**, 045001 (2019).

[50] M. Demirplak and S. A. Rice, *Adiabatic Population Transfer with Control Fields*, The Journal of Physical Chemistry A **107**, 9937–9945 (2003).

[51] M. V. Berry, *Transitionless quantum driving*, Journal of Physics A: Mathematical and Theoretical **42**, 365303 (2009).

[52] S. Ibáñez, X. Chen, E. Torrontegui, J. G. Muga, and A. Ruschhaupt, *Multiple Schrödinger Pictures and Dynamics in Shortcuts to Adiabaticity*, Physical Review Letters **109**, 100403 (2012).

[53] H. R. Lewis, Jr. and W. B. Riesenfeld, *An Exact Quantum Theory of the Time-Dependent Harmonic Oscillator and of a Charged Particle in a Time-Dependent Electromagnetic Field*, Journal of Mathematical Physics **10**, 1458–1473 (1969).

[54] S. Oh, Y.-P. Shim, J. Fei, M. Friesen, and X. Hu, *Resonant adiabatic passage with three qubits*, Physical Review A **87**, 022332 (2013).

[55] B. B. Zhou, A. Baksic, H. Ribeiro, C. G. Yale, F. J. Heremans, *et al.*, *Accelerated quantum control using superadiabatic dynamics in a solid-state lambda system*, Nature Physics **13**, 330–334 (2017).

[56] S. Martínez-Garaot, E. Torrontegui, X. Chen, and J. G. Muga, *Shortcuts to adiabaticity in three-level systems using Lie transforms*, Physical Review A **89**, 053408 (2014).

[57] Y.-H. Chen, W. Qin, X. Wang, A. Miranowicz, and F. Nori, *Shortcuts to Adiabaticity for the Quantum Rabi Model: Efficient Generation of Giant Entangled Cat States via Parametric Amplification*, Physical Review Letters **126**, 023602 (2021).

[58] P. W. Claeys, M. Pandey, D. Sels, and A. Polkovnikov, *Floquet-Engineering Counterdiabatic Protocols in Quantum Many-Body Systems*, Physical Review Letters **123**, 090602 (2019).

[59] D. Sels and A. Polkovnikov, *Minimizing irreversible losses in quantum systems by local counterdiabatic driving*, Proceedings of the National Academy of Sciences **114**, E3909–E3916 (2017).

[60] F. Cárdenas-López and X. Chen, *Shortcuts to Adiabaticity for Fast Qubit Readout in Circuit Quantum Electrodynamics*, Physical Review Applied **18**, 034010 (2022).

[61] F. A. Cárdenas-López, J. C. Retamal, and X. Chen, *Shortcuts to adiabaticity in superconducting circuits for fast multi-partite state generation*, Communications Physics **6**, 1–10 (2023).

[62] A. Del Campo, M. M. Rams, and W. H. Zurek, *Assisted Finite-Rate Adiabatic Passage Across a Quantum Critical Point: Exact Solution for the Quantum Ising Model*, Physical Review Letters **109**, 115703 (2012).

[63] M. Demirplak and S. A. Rice, *On the consistency, extremal, and global properties of counterdiabatic fields*, The Journal of Chemical Physics **129**, 154111 (2008).

[64] F. Motzoi, J. M. Gambetta, P. Rebentrost, and F. K. Wilhelm, *Simple Pulses for Elimination of Leakage in Weakly Nonlinear Qubits*, Physical Review Letters **103**, 110501 (2009).

[65] F. Motzoi, *Controlling Quantum Information Devices*, Ph.D. thesis, University of Waterloo (2012).

[66] J. M. Gambetta, F. Motzoi, S. T. Merkel, and F. K. Wilhelm, *Analytic control methods for high-fidelity unitary operations in a weakly nonlinear oscillator*, Physical Review A **83**, 012308 (2011).

[67] F. Motzoi and F. K. Wilhelm, *Improving frequency selection of driven pulses using derivative-based transition suppression*, Physical Review A **88**, 062318 (2013).

[68] L. S. Theis, F. Motzoi, and F. K. Wilhelm, *Simultaneous gates in frequency-crowded multilevel systems using fast, robust, analytic control shapes*, Physical Review A **93**, 012324 (2016).

[69] L. S. Theis and F. K. Wilhelm, *Nonadiabatic corrections to fast dispersive multiqubit gates involving Z control*, Physical Review A **95**, 022314 (2017).

[70] E. Hyyppä, A. Vepsäläinen, M. Papič, C. F. Chan, S. Inel, *et al.*, Reducing leakage of single-qubit gates for superconducting quantum processors using analytical control pulse envelopes (2024), arXiv:2402.17757 [quant-ph] .

[71] M. Werninghaus, D. J. Egger, F. Roy, S. Machnes, F. K. Wilhelm, and S. Filipp, *Leakage reduction in fast superconducting qubit gates via optimal control*, npj Quantum Information **7**, 14 (2021), arXiv:2003.05952 [quant-ph] .

[72] J. Berges, N. Tetradis, and C. Wetterich, *Non-perturbative renormalization flow in quantum field theory and statistical physics*, Physics Reports Renormalization Group Theory in the New Millennium. IV, **363**, 223–386 (2002).

[73] B. Li, T. Calarco, and F. Motzoi, *Nonperturbative Analytical Diagonalization of Hamiltonians with Application to Circuit QED*, PRX Quantum **3**, 030313 (2022).

[74] J. R. Schrieffer and P. A. Wolff, *Relation between the Anderson and Kondo Hamiltonians*, Physical Review **149**, 491–492 (1966).

[75] S. Bravyi, D. DiVincenzo, and D. Loss, *Schrieffer-Wolff transformation for quantum many-body systems*, Annals of Physics **326**, 2793–2826 (2011).

[76] E. Brion, L. H. Pedersen, and K. Mølmer, *Adiabatic elimination in a lambda system*, Journal of Physics A: Mathematical and Theoretical **40**, 1033–1043 (2007).

[77] E. H. Lieb and B. Simon, *The Thomas-Fermi theory of atoms, molecules and solids*, Advances in Mathematics **23**, 22–116 (1977).

[78] M. Born and R. Oppenheimer, *Zur Quantentheorie der Molekeln*, Annalen der Physik **389**, 457-484 (1927).

[79] K. Suzuki and R. Okamoto, *Perturbation Theory for Quasidegenerate System in Quantum Mechanics*, Progress of Theoretical Physics **72**, 534–548 (1984).

[80] E. Magesan and J. M. Gambetta, *Effective Hamiltonian models of the cross-resonance gate*, Physical Review A **101**, 052308 (2020).

[81] M. Malekakhlagh, E. Magesan, and D. C. McKay, *First-principles analysis of cross-resonance gate operation*, Physical Review A **102**, 042605 (2020).

[82] J. Romhányi, G. Burkard, and A. Pályi, *Subharmonic transitions and Bloch-Siegert shift in electrically driven spin resonance*, Physical Review B **92**, 054422 (2015).

[83] M. H. Goerz, F. Motzoi, K. B. Whaley, and C. P. Koch, *Charting the circuit QED design landscape using optimal control theory*, npj Quantum Information **3**, 37 (2017).

[84] T. Menke, F. Häse, S. Gustavsson, A. J. Kerman, W. D. Oliver, and A. Aspuru-Guzik, *Automated design of superconducting circuits and its application to 4-local couplers*, npj Quantum Information **7**, 49 (2021).

[85] C. G. J. Jacobi, *Über ein leichtes Verfahren, die in der Theorie der Sacularstorungen vorkommenden Gleichungen numerisch aufzulosen*, Journal fur die reine und angewandte Mathematik 10.1017/cbo9781139568012.016 (1846).

[86] G. E. Forsythe and P. Henrici, *The Cyclic Jacobi Method for Computing the Principal Values of a Complex Matrix*, Transactions of the American Mathematical Society **94**, 1 (1960).

[87] P. Henrici, *On the speed of convergence of cyclic and quasicyclic jacobi methods for computing eigenvalues of hermitian matrices*, Journal of the Society for Industrial and Applied Mathematics **6**, 144-162 (1958).

[88] C. Edmiston and K. Ruedenberg, *Localized atomic and molecular orbitals*, Reviews of Modern Physics **35**, 457–464 (1963).

[89] M. Malekakhlagh and E. Magesan, *Mitigating off-resonant error in the cross-resonance gate*, Physical Review A **105**, 012602 (2022).

[90] D. Sank, Z. Chen, M. Khezri, J. Kelly, R. Barends, *et al.*, *Measurement-Induced State Transitions in a Superconducting Qubit: Beyond the Rotating Wave Approximation*, Physical Review Letters **117**, 190503 (2016).

[91] B. Baker, A. C. Y. Li, N. Irons, N. Earnest, and J. Koch, *Adaptive rotating-wave approximation for driven open quantum systems*, Physical Review A **98**, 052111 (2018).

[92] J. Krause, C. Dickel, E. Vaal, M. Vielmetter, J. Feng, *et al.*, *Magnetic field resilience of three-dimensional transmons with thin-film al/alo$_x$/Al josephson junctions approaching 1 t*, Physical Review Applied **17**, 034032 (2022).

[93] P. Forn-Díaz, J. Lisenfeld, D. Marcos, J. J. Garcia-Ripoll, E. Solano, *et al.*, *Observation of the Bloch-Siegert shift in a qubit-oscillator system in the ultrastrong coupling regime*, Physical Review Letters **105**, 237001 (2010).

[94] L. DiCarlo, J. M. Chow, J. M. Gambetta, L. S. Bishop, B. R. Johnson, *et al.*, *Demonstration of two-qubit algorithms with a superconducting quantum processor*, Nature **460**, 240–244 (2009).

[95] Y. Chen, C. Neill, P. Roushan, N. Leung, M. Fang, *et al.*, *Qubit Architecture with High Coherence and Fast Tunable Coupling*, Physical Review Letters **113**, 220502 (2014).

[96] R. Barends, J. Kelly, A. Megrant, A. Veitia, D. Sank, *et al.*, *Superconducting quantum circuits at the surface code threshold for fault tolerance*, Nature **508**, 500–503 (2014).

[97] M. A. Rol, F. Battistel, F. K. Malinowski, C. C. Bultink, B. M. Tarasinski, *et al.*, *Fast, High-Fidelity Conditional-Phase Gate Exploiting Leakage Interference in Weakly Anharmonic Superconducting Qubits*, Physical Review Letters **123**, 120502 (2019).

[98] P. Zhao, P. Xu, D. Lan, J. Chu, X. Tan, *et al.*, *High-Contrast ZZ Interaction Using Superconducting Qubits with Opposite-Sign Anharmonicity*, Physical Review Letters **125**, 200503 (2020).

[99] J. Ku, X. Xu, M. Brink, D. C. McKay, J. B. Hertzberg, *et al.*, *Suppression of Unwanted Z Z Interactions in a Hybrid Two-Qubit System*, Physical Review Letters **125**, 200504 (2020).

[100] Y. Xu, J. Chu, J. Yuan, J. Qiu, Y. Zhou, *et al.*, *High-fidelity, high-scalability two-qubit gate scheme for superconducting qubits*, Physical Review Letters **125**, 240503 (2020).

[101] E. A. Sete, A. Q. Chen, R. Manenti, S. Kulshreshtha, and S. Poletto, *Floating tunable coupler for scalable quantum computing architectures*, Physical Review Applied **15**, 064063 (2021).

[102] X. Xu and M. H. Ansari, *ZZ freedom in two qubit gates*, Physical Review Applied **15**, 064074 (2021).

[103] P. Zhao, D. Lan, P. Xu, G. Xue, M. Blank, *et al.*, *Suppression of static ZZ interaction in an all-transmon quantum processor*, Physical Review Applied **16**, 024037 (2021), arXiv:2011.03976 [quant-ph] .

[104] J. Stehlik, D. M. Zajac, D. L. Underwood, T. Phung, J. Blair, *et al.*, *Tunable Coupling Architecture for Fixed-Frequency Transmon Superconducting Qubits*, Physical Review Letters **127**, 080505 (2021).

[105] Y. Sung, L. Ding, J. Braumüller, A. Vepsäläinen, B. Kannan, *et al.*, *Realization of high-fidelity CZ and ZZ-free iSWAP gates with a tunable coupler*, Physical Review X **11**, 021058 (2021).

[106] A. Kandala, K. X. Wei, S. Srinivasan, E. Magesan, S. Carnevale, *et al.*, *Demonstration of a High-Fidelity CNOT Gate for Fixed-Frequency Transmons with Engineered ZZ Suppression*, Physical Review Letters **127**, 130501 (2021).

[107] P. Mundada, G. Zhang, T. Hazard, and A. Houck, *Suppression of Qubit Crosstalk in a Tunable Coupling Superconducting Circuit*, Physical Review Applied **12**, 054023 (2019).

[108] M. C. Collodo, J. Herrmann, N. Lacroix, C. K. Andersen, A. Remm, *et al.*, *Implementation of Conditional-Phase Gates based on tunable ZZ-Interactions*, Physical Review Letters **125**, 240502 (2020).

[109] J. Chu and F. Yan, *Coupler-Assisted Controlled-Phase Gate with Enhanced Adiabaticity*, Physical Review Applied **16**, 054020 (2021).

[110] A. D. K. Finck, S. Carnevale, D. Klaus, C. Scerbo, J. Blair, *et al.*, *Suppressed crosstalk between two-junction superconducting qubits with mode-selective exchange coupling*, Physical Review Applied **16**, 054041 (2021), arXiv:2105.11495 [quant-ph] .

[111] K. X. Wei, E. Magesan, I. Lauer, S. Srinivasan, D. F. Bogorin, *et al.*, Quantum crosstalk cancellation for fast entangling gates and improved multi-qubit performance (2021), arXiv:2106.00675 [quant-ph] .

[112] B. K. Mitchell, R. K. Naik, A. Morvan, A. Hashim, J. M. Kreikebaum, *et al.*, *Hardware-Efficient Microwave-Activated Tunable Coupling between Superconducting Qubits*, Physical Review Letters **127**, 200502 (2021).

[113] H. Xiong, Q. Ficheux, A. Somoroff, L. B. Nguyen, E. Dogan, *et al.*, Arbitrary controlled-phase gate on fluxonium qubits using differential ac-Stark shifts (2021), arXiv:2103.04491 [cond-mat, physics:quant-ph] .

[114] C. F. Van Loan and G. Golub, *Matrix computations* (The Johns Hopkins University Press, 1996).

[115] L. S. Cederbaum, J. Schirmer, and H. D. Meyer, *Block diagonalisation of Hermitian matrices*, Journal of Physics A: Mathematical and General **22**, 2427–2439 (1989).

[116] D. Zeuch, F. Hassler, J. J. Slim, and D. P. DiVincenzo, *Exact Rotating Wave Approximation*, Annals of Physics **423**, 168327 (2020), arXiv:1807.02858 [quant-ph] .

[117] B. Khani, J. M. Gambetta, F. Motzoi, and F. K. Wilhelm, *Optimal generation of Fock states in a weakly nonlinear oscillator*, Physica Scripta **2009**, 014021 (2009).

[118] E. T. Holland, B. Vlastakis, R. W. Heeres, M. J. Reagor, U. Vool, *et al.*, *Single-Photon-Resolved Cross-Kerr Interaction for Autonomous Stabilization of Photon-Number States*, Physical Review Letters **115**, 180501 (2015).

[119] X. Li, T. Cai, H. Yan, Z. Wang, X. Pan, *et al.*, *Tunable Coupler for Realizing a Controlled-Phase Gate with Dynamically Decoupled Regime in a Superconducting Circuit*, Physical Review Applied **14**, 024070 (2020).

[120] G. Zhu, D. G. Ferguson, V. E. Manucharyan, and J. Koch, *Circuit QED with fluxonium qubits: Theory of the dispersive regime*, Physical Review B **87**, 024510 (2013).

[121] J. B. Hertzberg, E. J. Zhang, S. Rosenblatt, E. Magesan, J. A. Smolin, *et al.*, *Laser-annealing Josephson junctions for yielding scaled-up superconducting quantum processors*, npj Quantum Information **7**, 129 (2021).

[122] G. S. Paraoanu, *Microwave-induced coupling of superconducting qubits*, Physical Review B **74**, 140504(R) (2006).

[123] C. Rigetti and M. Devoret, *Fully microwave-tunable universal gates in superconducting qubits with linear couplings and fixed transition frequencies*, Physical Review B **81**, 134507 (2010).

[124] J. M. Chow, A. D. Córcoles, J. M. Gambetta, C. Rigetti, B. R. Johnson, *et al.*, *Simple All-Microwave Entangling Gate for Fixed-Frequency Superconducting Qubits*, Physical Review Letters **107**, 080502 (2011).

[125] S. Sheldon, L. S. Bishop, E. Magesan, S. Filipp, J. M. Chow, and J. M. Gambetta, *Characterizing errors on qubit operations via iterative randomized benchmarking*, Physical Review A **93**, 012301 (2016).

[126] S. Kirchhoff, T. Keßler, P. J. Liebermann, E. Assémat, S. Machnes, *et al.*, *Optimized cross-resonance gate for coupled transmon systems*, Physical Review A **97**, 042348 (2018).

[127] G. Gualdi and C. P. Koch, *Renormalization approach to non-Markovian open-quantum-system dynamics*, Physical Review A **88**, 022122 (2013).

[128] I. de Vega and D. Alonso, *Dynamics of non-markovian open quantum systems*, Reviews of Modern Physics **89**, 015001 (2017).

[129] S. G. Schirmer and X. Wang, *Stabilizing open quantum systems by markovian reservoir engineering*, Physical Review A **81**, 062306 (2010).

[130] X. Antoine, W. Bao, and C. Besse, *Computational methods for the dynamics of the nonlinear Schrödinger/Gross–Pitaevskii equations*, Computer Physics Communications **184**, 2621–2633 (2013).

[131] M. Dalgaard, C. A. Weidner, and F. Motzoi, *Dynamical uncertainty propagation with noisy quantum parameters*, Physical Review Letters **128**, 150503 (2022).

[132] L. S. Theis, F. Motzoi, S. Machnes, and F. K. Wilhelm, *Counteracting systems of diabaticities using DRAG controls: The status after 10 years*, EPL (Europhysics Letters) **123**, 60001 (2018).

[133] R. Unanyan, L. Yatsenko, K. Bergmann, and B. Shore, *Laser-induced adiabatic atomic reorientation with control of diabatic losses*, Optics Communications **139**, 48–54 (1997).

[134] Z. Chen, J. Kelly, C. Quintana, R. Barends, A. N. Korotkov, *et al.*, *Measuring and Suppressing Quantum State Leakage in a Superconducting Qubit*, Physical Review Letters **116**, 020501 (2016).

[135] J. M. Chow, L. DiCarlo, J. M. Gambetta, F. Motzoi, L. Frunzio, *et al.*, *Optimized driving of superconducting artificial atoms for improved single-qubit gates*, Physical Review A **82**, 040305 (2010).

[136] B. Li, T. Calarco, and F. Motzoi, *Experimental error suppression in Cross-Resonance gates via multi-derivative pulse shaping*, npj Quantum Information **10**, 1–10 (2024).

[137] Y. Zhao, Y. Ye, H.-L. Huang, Y. Zhang, D. Wu, *et al.*, *Realization of an Error-Correcting Surface Code with Superconducting Qubits*, Physical Review Letters **129**, 030501 (2022).

[138] R. Acharya, I. Aleiner, R. Allen, T. I. Andersen, M. Ansmann, *et al.*, *Suppressing quantum errors by scaling a surface code logical qubit*, Nature **614**, 676–681 (2023).

[139] K. J. Satzinger, Y.-J. Liu, A. Smith, C. Knapp, M. Newman, *et al.*, *Realizing topologically ordered states on a quantum processor*, Science **374**, 1237–1241 (2021).

[140] T. I. Andersen, Y. D. Lensky, K. Kechedzhi, I. K. Drozdov, A. Bengtsson, *et al.*, *Non-Abelian braiding of graph vertices in a superconducting processor*, Nature **618**, 264–269 (2023).

[141] IBM Quantum (2024).

[142] Y. Kim, A. Eddins, S. Anand, K. X. Wei, E. van den Berg, *et al.*, *Evidence for the utility of quantum computing before fault tolerance*, Nature **618**, 500–505 (2023).

[143] K. X. Wei, E. Magesan, I. Lauer, S. Srinivasan, D. F. Bogorin, *et al.*, *Hamiltonian Engineering with Multicolor Drives for Fast Entangling Gates and Quantum Crosstalk Cancellation*, Physical Review Letters **129**, 060501 (2022).

[144] D. Willsch, M. Nocon, F. Jin, H. De Raedt, and K. Michielsen, *Gate-error analysis in simulations of quantum computers with transmon qubits*, Physical Review A **96**, 062302 (2017).

[145] M. McEwen, D. Kafri, Z. Chen, J. Atalaya, K. J. Satzinger, *et al.*, *Removing leakage-induced correlated errors in superconducting quantum error correction*, Nature Communications **12**, 1761 (2021).

[146] K. C. Miao, M. McEwen, J. Atalaya, D. Kafri, L. P. Pryadko, *et al.*, *Overcoming leakage in quantum error correction*, Nature Physics **19**, 1780–1786 (2023).

[147] B. M. Varbanov, F. Battistel, B. M. Tarasinski, V. P. Ostroukh, T. E. O'Brien, *et al.*, *Leakage detection for a transmon-based surface code*, npj Quantum Information **6**, 102 (2020).

[148] C. C. Bultink, T. E. O'Brien, R. Vollmer, N. Muthusubramanian, M. W. Beekman, *et al.*, *Protecting quantum entanglement from leakage and qubit errors via repetitive parity measurements*, Science Advances **6**, eaay3050 (2020).

[149] P. C. de Groot, J. Lisenfeld, R. N. Schouten, S. Ashhab, A. Lupaşcu, *et al.*, *Selective darkening of degenerate transitions demonstrated with two superconducting quantum bits*, Nature Physics **6**, 763–766 (2010).

[150] P. Groszkowski, A. G. Fowler, F. Motzoi, and F. K. Wilhelm, *Tunable coupling between three qubits as a building block for a superconducting quantum computer*, Physical Review B **84**, 144516 (2011).

[151] S. Sheldon, E. Magesan, J. M. Chow, and J. M. Gambetta, *Procedure for systematically tuning up cross-talk in the cross-resonance gate*, Physical Review A **93**, 060302(R) (2016).

[152] M. Takita, A. D. Córcoles, E. Magesan, B. Abdo, M. Brink, *et al.*, *Demonstration of Weight-Four Parity Measurements in the Surface Code Architecture*, Physical Review Letters **117**, 210505 (2016).

[153] M. Takita, A. W. Cross, A. D. Córcoles, J. M. Chow, and J. M. Gambetta, *Experimental Demonstration of Fault-Tolerant State Preparation with Superconducting Qubits*, Physical Review Letters **119**, 180501 (2017).

[154] V. Tripathi, M. Khezri, and A. N. Korotkov, *Operation and intrinsic error budget of a two-qubit cross-resonance gate*, Physical Review A **100**, 012301 (2019).

[155] K. X. Wei, E. Pritchett, D. M. Zajac, D. C. McKay, and S. Merkel, *Characterizing non-Markovian off-resonant errors in quantum gates*, Physical Review Applied **21**, 024018 (2024).

[156] P. Zhao, K. Linghu, Z. Li, P. Xu, R. Wang, *et al.*, *Quantum Crosstalk Analysis for Simultaneous Gate Operations on Superconducting Qubits*, PRX Quantum **3**, 020301 (2022).

[157] S. K. Tolpygo, V. Bolkhovsky, T. J. Weir, L. M. Johnson, M. A. Gouker, and W. D. Oliver, *Fabrication process and properties of fully-planarized deep-submicron nb/al- AlO_x/Nb josephson junctions for vlsi circuits*, IEEE Transactions on Applied Superconductivity **25**, 1-12 (2015).

[158] X. Xu and M. Ansari, *Parasitic-Free Gate: An Error-Protected Cross-Resonance Switch in Weakly Tunable Architectures*, Physical Review Applied **19**, 024057 (2023).

[159] M. Dalgaard, F. Motzoi, J. J. Sørensen, and J. Sherson, *Global optimization of quantum dynamics with AlphaZero deep exploration*, npj Quantum Information **6**, 6 (2020).

[160] Y. Baum, M. Amico, S. Howell, M. Hush, M. Liuzzi, *et al.*, *Experimental Deep Reinforcement Learning for Error-Robust Gate-Set Design on a Superconducting Quantum Computer*, PRX Quantum **2**, 040324 (2021).

[161] T. Alexander, N. Kanazawa, D. J. Egger, L. Capelluto, C. J. Wood, *et al.*, *Qiskit Pulse: Programming Quantum Computers Through the Cloud with Pulses*, Quantum Science and Technology **5**, 044006 (2020).

[162] A. Morvan, V. V. Ramasesh, M. S. Blok, J. M. Kreikebaum, K. O'Brien, *et al.*, *Qutrit Randomized Benchmarking*, Physical Review Letters **126**, 210504 (2021).

[163] N. Sundaresan, I. Lauer, E. Pritchett, E. Magesan, P. Jurcevic, and J. M. Gambetta, *Reducing Unitary and Spectator Errors in Cross Resonance with Optimized Rotary Echoes*, PRX Quantum **1**, 020318 (2020).

[164] A. Somoroff, Q. Ficheux, R. A. Mencia, H. Xiong, R. Kuzmin, and V. E. Manucharyan, *Millisecond Coherence in a Superconducting Qubit*, Physical Review Letters **130**, 267001 (2023).

[165] C. Wang, X. Li, H. Xu, Z. Li, J. Wang, *et al.*, *Towards practical quantum computers: Transmon qubit with a lifetime approaching 0.5 milliseconds*, npj Quantum Information **8**, 3 (2022).

[166] K. N. Nesterov, C. Wang, V. E. Manucharyan, and M. G. Vavilov, *CNOT Gates for Fluxonium Qubits via Selective Darkening of Transitions*, Physical Review Applied **18**, 034063 (2022).

[167] E. Dogan, D. Rosenstock, L. Le Guevel, H. Xiong, R. A. Mencia, *et al.*, *Two-Fluxonium Cross-Resonance Gate*, Physical Review Applied **20**, 024011 (2023).

[168] Q. Ficheux, L. B. Nguyen, A. Somoroff, H. Xiong, K. N. Nesterov, *et al.*, *Fast Logic with Slow Qubits: Microwave-Activated Controlled-Z Gate on Low-Frequency Fluxoniums*, Physical Review X **11**, 021026 (2021).

[169] Y. Kim, A. Morvan, L. B. Nguyen, R. K. Naik, C. Jünger, *et al.*, *High-fidelity three-qubit iToffoli gate for fixed-frequency superconducting qubits*, Nature Physics **18**, 783–788 (2022).

[170] N. Goss, A. Morvan, B. Marinelli, B. K. Mitchell, L. B. Nguyen, *et al.*, *High-fidelity qutrit entangling gates for superconducting circuits*, Nature Communications **13**, 7481 (2022).

[171] T. Q. Cai, X. Y. Han, Y. K. Wu, Y.-L. Ma, J. H. Eberly, *et al.*, *Impact of Spectators on a Two-Qubit Gate in a Tunable Coupling Superconducting Circuit*, Physical Review Letters **127**, 060505 (2021).

[172] J. F. Marques, H. Ali, B. M. Varbanov, M. Finkel, H. M. Veen, *et al.*, *All-Microwave Leakage Reduction Units for Quantum Error Correction with Superconducting Transmon Qubits*, Physical Review Letters **130**, 250602 (2023).

[173] A. Osman, J. Fernández-Pendás, C. Warren, S. Kosen, M. Scigliuzzo, *et al.*, *Mitigation of frequency collisions in superconducting quantum processors*, Physical Review Research **5**, 043001 (2023).

[174] L. H. Pedersen, N. M. Møller, and K. Mølmer, *Fidelity of quantum operations*, Physics Letters A **367**, 47–51 (2007).

[175] D. C. McKay, C. J. Wood, S. Sheldon, J. M. Chow, and J. M. Gambetta, *Efficient Z gates for quantum computing*, Physical Review A **96**, 022330 (2017).

[176] D. Danin and F. Tennie, Procedure for improving cross-resonance noise resistance using pulse-level control (2023), arXiv:2303.12771 [quant-ph] .

[177] B. Li, F. A. Cárdenas-López, A. Lupascu, and F. Motzoi, Universal pulses for superconducting qudit ladder gates (2024), arXiv:2412.18339 [quant-ph] .

[178] D. Gottesman, in *Quantum Computing and Quantum Communications*, Vol. 1509, edited by G. Goos, J. Hartmanis, J. Van Leeuwen, and C. P. Williams (Springer Berlin Heidelberg, Berlin, Heidelberg, 1999) pp. 302–313.

[179] Y.-M. Di and H.-R. Wei, *Optimal synthesis of multivalued quantum circuits*, Physical Review A **92**, 062317 (2015).

[180] F. Motzoi, M. P. Kaicher, and F. K. Wilhelm, *Linear and Logarithmic Time Compositions of Quantum Many-Body Operators*, Physical Review Letters **119**, 160503 (2017).

[181] S. Cao, M. Bakr, G. Campanaro, S. D. Fasciati, J. Wills, *et al.*, *Emulating two qubits with a four-level transmon qudit for variational quantum algorithms*, Quantum Science and Technology **9**, 035003 (2024), arXiv:2303.04796 [quant-ph] .

[182] A. Galda, M. Cubeddu, N. Kanazawa, P. Narang, and N. Earnest-Noble, Implementing a Ternary Decomposition of the Toffoli Gate on Fixed-FrequencyTransmon Qutrits (2021), arXiv:2109.00558 [quant-ph] .

[183] B. P. Lanyon, M. Barbieri, M. P. Almeida, T. Jennewein, T. C. Ralph, *et al.*, *Simplifying quantum logic using higher-dimensional Hilbert spaces*, Nature Physics **5**, 134–140 (2009).

[184] P. J. Ollitrault, G. Mazzola, and I. Tavernelli, *Nonadiabatic molecular quantum dynamics with quantum computers*, Phys. Rev. Lett. **125**, 260511 (2020).

[185] A. Miessen, P. J. Ollitrault, and I. Tavernelli, *Quantum algorithms for quantum dynamics: A performance study on the spin-boson model*, Phys. Rev. Res. **3**, 043212 (2021).

[186] E. Rico, M. Dalmonte, P. Zoller, D. Banerjee, M. Bögli, *et al.*, *So(3) "nuclear physics" with ultracold gases*, Annals of Physics **393**, 466-483 (2018).

[187] G. Mazzola, S. V. Mathis, G. Mazzola, and I. Tavernelli, *Gauge-invariant quantum circuits for u(1) and yang-mills lattice gauge theories*, Phys. Rev. Res. **3**, 043209 (2021).

[188] M. Meth, J. F. Haase, J. Zhang, C. Edmunds, L. Postler, *et al.*, Simulating 2D lattice gauge theories on a qudit quantum computer (2024), arXiv:2310.12110 .

[189] D. Bruß and C. Macchiavello, *Optimal eavesdropping in cryptography with three-dimensional quantum states*, Phys. Rev. Lett. **88**, 127901 (2002).

[190] H. Bechmann-Pasquinucci and A. Peres, *Quantum cryptography with 3-state systems*, Phys. Rev. Lett. **85**, 3313–3316 (2000).

[191] M. Grace, C. Brif, H. Rabitz, I. Walmsley, R. Kosut, and D. Lidar, *Encoding a qubit into multilevel subspaces*, New Journal of Physics **8**, 35 (2006).

[192] A. Chiesa, E. Macaluso, F. Petiziol, S. Wimberger, P. Santini, and S. Carretta, *Molecular Nanomagnets as Qubits with Embedded Quantum-Error Correction*, The Journal of Physical Chemistry Letters **11**, 8610–8615 (2020).

[193] E. T. Campbell, *Enhanced Fault-Tolerant Quantum Computing in d-Level Systems*, Physical Review Letters **113**, 230501 (2014).

[194] P. J. Low, B. M. White, A. A. Cox, M. L. Day, and C. Senko, *Practical trapped-ion protocols for universal qudit-based quantum computing*, Physical Review Research **2**, 033128 (2020).

[195] M. Ringbauer, M. Meth, L. Postler, R. Stricker, R. Blatt, *et al.*, *A universal qudit quantum processor with trapped ions*, Nature Physics **18**, 1053–1057 (2022).

[196] P. Hrmo, B. Wilhelm, L. Gerster, M. W. van Mourik, M. Huber, *et al.*, *Native qudit entanglement in a trapped ion quantum processor*, Nature Communications **14**, 2242 (2023).

[197] P. J. Low, B. White, and C. Senko, Control and Readout of a 13-level Trapped Ion Qudit (2023), arXiv:2306.03340 .

[198] D. González-Cuadra, T. V. Zache, J. Carrasco, B. Kraus, and P. Zoller, *Hardware Efficient Quantum Simulation of Non-Abelian Gauge Theories with Qudits on Rydberg Platforms*, Physical Review Letters **129**, 160501 (2022).

[199] R. Hussain, G. Allodi, A. Chiesa, E. Garlatti, D. Mitcov, *et al.*, *Coherent manipulation of a molecular ln-based nuclear qudit coupled to an electron qubit*, Journal of the American Chemical Society **140**, 9814-9818 (2018).

[200] M. Chizzini, L. Crippa, L. Zaccardi, E. Macaluso, S. Carretta, *et al.*, *Quantum error correction with molecular spin qudits*, Phys. Chem. Chem. Phys. **24**, 20030-20039 (2022).

[201] H. Biard, E. Moreno-Pineda, M. Ruben, E. Bonet, W. Wernsdorfer, and F. Balestro, *Increasing the Hilbert space dimension using a single coupled molecular spin*, Nature Communications **12**, 4443 (2021).

[202] M. Kues, C. Reimer, P. Roztocki, L. R. Cortés, S. Sciara, *et al.*, *On-chip generation of high-dimensional entangled quantum states and their coherent control*, Nature **546**, 622–626 (2017).

[203] M. Erhard, M. Malik, M. Krenn, and A. Zeilinger, *Experimental Greenberger–Horne–Zeilinger entanglement beyond qubits*, Nature Photonics **12**, 759–764 (2018).

[204] Y.-H. Luo, H.-S. Zhong, M. Erhard, X.-L. Wang, L.-C. Peng, *et al.*, *Quantum Teleportation in High Dimensions*, Physical Review Letters **123**, 070505 (2019).

[205] E. J. Davis, G. Bentsen, L. Homeier, T. Li, and M. H. Schleier-Smith, *Photon-Mediated Spin-Exchange Dynamics of Spin-1 Atoms*, Physical Review Letters **122**, 010405 (2019).

[206] Y. Chi, J. Huang, Z. Zhang, J. Mao, Z. Zhou, *et al.*, *A programmable qudit-based quantum processor*, Nature Communications **13**, 1166 (2022).

[207] M. S. Blok, V. V. Ramasesh, T. Schuster, K. O'Brien, J. M. Kreikebaum, *et al.*, *Quantum Information Scrambling on a Superconducting Qutrit Processor*, Physical Review X **11**, 021010 (2021).

[208] P. Liu, R. Wang, J.-N. Zhang, Y. Zhang, X. Cai, *et al.*, *Performing SU (d) Operations and Rudimentary Algorithms in a Superconducting Transmon Qudit for d = 3 and d = 4*, Physical Review X **13**, 021028 (2023).

[209] E. Champion, Z. Wang, R. Parker, and M. Blok, Multi-frequency control and measurement of a spin-7/2 system encoded in a transmon qudit (2024), arXiv:2405.15857 [quant-ph] .

[210] M. A. Yurtalan, J. Shi, M. Kononenko, A. Lupascu, and S. Ashhab, *Implementation of a Walsh-Hadamard Gate in a Superconducting Qutrit*, Physical Review Letters **125**, 180504 (2020).

[211] M. Kononenko, M. A. Yurtalan, S. Ren, J. Shi, S. Ashhab, and A. Lupascu, *Characterization of control in a superconducting qutrit using randomized benchmarking*, Physical Review Research **3**, L042007 (2021).

[212] M. Yurtalan, J. Shi, G. Flatt, and A. Lupascu, *Characterization of Multilevel Dynamics and Decoherence in a High-Anharmonicity Capacitively Shunted Flux Circuit*, Physical Review Applied **16**, 054051 (2021).

[213] K. Luo, W. Huang, Z. Tao, L. Zhang, Y. Zhou, *et al.*, *Experimental Realization of Two Qutrits Gate with Tunable Coupling in Superconducting Circuits*, Physical Review Letters **130**, 030603 (2023).

[214] E. Lucero, J. Kelly, R. C. Bialczak, M. Lenander, M. Mariantoni, *et al.*, *Reduced phase error through optimized control of a superconducting qubit*, Physical Review A **82**, 042339 (2010).

[215] Z. Wang, R. W. Parker, E. Champion, and M. S. Blok, Systematic study of high e_j/e_c transmon qudits up to $d = 12$ (2024), arXiv:2407.17407 [quant-ph] .

[216] F. Preti, T. Calarco, and F. Motzoi, Continuous quantum gate sets and pulse class meta-optimization (2022), arXiv:2203.13594 [quant-ph] .

[217] V. Tripathi, N. Goss, A. Vezvaee, L. B. Nguyen, I. Siddiqi, and D. A. Lidar, Qudit Dynamical Decoupling on a Superconducting Quantum Processor (2024), arXiv:2407.04893 [quant-ph] .

[218] M. Deschamps, G. Kervern, D. Massiot, G. Pintacuda, L. Emsley, and P. J. Grandinetti, *Superadiabaticity in magnetic resonance*, The Journal of Chemical Physics **129**, 204110 (2008).

[219] V. Ramakrishna, R. Ober, X. Sun, O. Steuernagel, J. Botina, and H. Rabitz, *Explicit generation of unitary transformations in a single atom or molecule*, Physical Review A **61**, 032106 (2000).

[220] G. K. Brennen, D. P. O'Leary, and S. S. Bullock, *Criteria for Exact Qudit Universality*, Physical Review A **71**, 052318 (2005), arXiv:quant-ph/0407223 .

[221] G. Ithier, E. Collin, P. Joyez, P. J. Meeson, D. Vion, *et al.*, *Decoherence in a superconducting quantum bit circuit*, Phys. Rev. B **72**, 134519 (2005).

[222] O. Astafiev, Y. A. Pashkin, Y. Nakamura, T. Yamamoto, and J. S. Tsai, *Quantum noise in the josephson charge qubit*, Phys. Rev. Lett. **93**, 267007 (2004).

[223] A. B. Zorin, F.-J. Ahlers, J. Niemeyer, T. Weimann, H. Wolf, *et al.*, *Background charge noise in metallic single-electron tunneling devices*, Phys. Rev. B **53**, 13682–13687 (1996).

[224] B. G. Christensen, C. D. Wilen, A. Opremcak, J. Nelson, F. Schlenker, *et al.*, *Anomalous charge noise in superconducting qubits*, Phys. Rev. B **100**, 140503 (2019).

[225] W. Smith, A. Kou, X. Xiao, U. Vool, and M. Devoret, *Superconducting circuit protected by two-cooper-pair tunneling*, npj Quantum Information **6**, 8 (2020).

[226] H. Zhang, S. Chakram, T. Roy, N. Earnest, Y. Lu, *et al.*, *Universal fast-flux control of a coherent, low-frequency qubit*, Phys. Rev. X **11**, 011010 (2021).

[227] V. Braginsky, V. Ilchenko, and K. Bagdassarov, *Experimental observation of fundamental microwave absorption in high-quality dielectric crystals*, Physics Letters A **120**, 300-305 (1987).

[228] C. Wang, C. Axline, Y. Y. Gao, T. Brecht, Y. Chu, *et al.*, *Surface participation and dielectric loss in superconducting qubits*, Applied Physics Letters **107**, 162601 (2015).

[229] A. P. M. Place, L. V. H. Rodgers, P. Mundada, B. M. Smitham, M. Fitzpatrick, *et al.*, *New material platform for superconducting transmon qubits with coherence times exceeding 0.3 milliseconds*, Nature Communications **12**, 1779 (2021).

[230] M. Tuokkola, Y. Sunada, H. Kivijärvi, L. Grönberg, J.-P. Kaikkonen, *et al.*, Methods to achieve near-millisecond energy relaxation and dephasing times for a superconducting transmon qubit (2024), arXiv:2407.18778 [quant-ph] .

[231] M. Bal, A. A. Murthy, S. Zhu, F. Crisa, X. You, *et al.*, *Systematic improvements in transmon qubit coherence enabled by niobium surface encapsulation*, npj Quantum Information **10**, 43 (2024).

[232] S. Kono, J. Pan, M. Chegnizadeh, X. Wang, A. Youssefi, *et al.*, *Mechanically induced correlated errors on superconducting qubits with relaxation times exceeding 0.4 ms*, Nature Communications **15**, 3950 (2024).

[233] M. Sarovar, T. Proctor, K. Rudinger, K. Young, E. Nielsen, and R. Blume-Kohout, *Detecting crosstalk errors in quantum information processors*, Quantum **4**, 321 (2020), arXiv:1908.09855 [quant-ph] .

[234] D. M. Abrams, N. Didier, S. A. Caldwell, B. R. Johnson, and C. A. Ryan, *Methods for Measuring Magnetic Flux Crosstalk between Tunable Transmons*, Physical Review Applied **12**, 064022 (2019).

[235] B. Li, S. Ahmed, S. Saraogi, N. Lambert, F. Nori, *et al.*, *Pulse-level noisy quantum circuits with QuTiP*, Quantum **6**, 630 (2022).

[236] J. Preskill, *Quantum computing in the NISQ era and beyond*, Quantum **2**, 79 (2018).

[237] I. Buluta, S. Ashhab, and F. Nori, *Natural and artificial atoms for quantum computation*, Reports on Progress in Physics **74**, 104401 (2011).

[238] K. Bharti, A. Cervera-Lierta, T. H. Kyaw, T. Haug, S. Alperin-Lea, *et al.*, *Noisy intermediate-scale quantum algorithms*, Reviews of Modern Physics **94**, 015004 (2022).

[239] R. S. Smith, M. J. Curtis, and W. J. Zeng, A Practical Quantum Instruction Set Architecture (2016), arXiv:1608.03355 .

[240] P. J. Karalekas, N. A. Tezak, E. C. Peterson, C. A. Ryan, M. P. da Silva, and R. S. Smith, *A quantum-classical cloud platform optimized for variational hybrid algorithms*, Quantum Science and Technology **5**, 024003 (2020).

[241] G. Aleksandrowicz, T. Alexander, P. Barkoutsos, L. Bello, Y. Ben-Haim, *et al.*, Qiskit: An Open-source Framework for Quantum Computing (2019).

[242] C. Developers, Cirq (2021), See full list of authors on Github: `https://github.com/quantumlib/Cirq/graphs/contributors`.

[243] D. S. Steiger, T. Häner, and M. Troyer, *ProjectQ: an open source software framework for quantum computing*, Quantum **2**, 49 (2018).

[244] V. Bergholm, J. Izaac, M. Schuld, C. Gogolin, M. S. Alam, *et al.*, PennyLane: Automatic differentiation of hybrid quantum-classical computations (2018), arXiv:1811.04968 .

[245] M. Fingerhuth, T. Babej, and P. Wittek, *Open source software in quantum computing*, PLOS ONE **13**, e0208561 (2018).

[246] B. Heim, M. Soeken, S. Marshall, C. Granade, M. Roetteler, *et al.*, *Quantum programming languages*, Nature Reviews Physics **2**, 709–722 (2020).

[247] H. Ball, M. J. Biercuk, A. R. R. Carvalho, J. Chen, M. Hush, *et al.*, *Software tools for quantum control: Improving quantum computer performance through noise and error suppression*, Quantum Science and Technology **6**, 044011 (2021).

[248] H. Silvério, S. Grijalva, C. Dalyac, L. Leclerc, P. J. Karalekas, *et al.*, *Pulser: An open-source package for the design of pulse sequences in programmable neutral-atom arrays*, Quantum **6**, 629 (2022).

[249] J. Johansson, P. Nation, and F. Nori, *QuTiP: An open-source Python framework for the dynamics of open quantum systems*, Computer Physics Communications **183**, 1760–1772 (2012).

[250] N. Shammah, S. Ahmed, N. Lambert, S. De Liberato, and F. Nori, *Open quantum systems with local and collective incoherent processes: Efficient numerical simulations using permutational invariance*, Physical Review A **98**, 063815 (2018).

[251] N. Lambert, S. Ahmed, M. Cirio, and F. Nori, *Modelling the ultra-strongly coupled spin-boson model with unphysical modes*, Nature Communications **10**, 3721 (2019).

[252] N. Lambert, T. Raheja, S. Cross, P. Menczel, S. Ahmed, *et al.*, *QuTiP-BoFiN: A bosonic and fermionic numerical hierarchical-equations-of-motion library with applications in light-harvesting, quantum control, and single-molecule electronics*, Physical Review Research **5**, 013181 (2023).

[253] J. D. Teske and H. Bluhm, in *IEEE Int. Conf. Quantum Comput. Eng. (QCE)* (2021) p. 441.

[254] L. B.-V. Horn, sequencing-dev/sequencing: v1.1.3 (2021).

[255] P. Groszkowski and J. Koch, *Scqubits: a Python package for superconducting qubits*, Quantum **5**, 583 (2021).

[256] C. R. Harris, K. J. Millman, S. J. van der Walt, R. Gommers, P. Virtanen, *et al.*, *Array programming with NumPy*, Nature **585**, 357–362 (2020).

[257] P. Virtanen, R. Gommers, T. E. Oliphant, M. Haberland, T. Reddy, *et al.*, *SciPy 1.0: Fundamental algorithms for scientific computing in Python*, Nature Methods **17**, 261–272 (2020).

[258] J. D. Hunter, *Matplotlib: A 2D Graphics Environment*, Computing in Science & Engineering **9**, 90–95 (2007).

[259] S. Behnel, R. Bradshaw, C. Citro, L. Dalcin, D. S. Seljebotn, and K. Smith, *Cython: The best of both worlds*, Computing in Science & Engineering **13**, 31 (2011).

[260] The full list of qutip-qip contributors, https://github.com/qutip/qutip-qip/graphs/contributors.

[261] A. W. Cross, L. S. Bishop, J. A. Smolin, and J. M. Gambetta, Open Quantum Assembly Language (2017), arXiv:1707.03429 .

[262] A. W. Cross, A. Javadi-Abhari, T. Alexander, N. de Beaudrap, L. S. Bishop, *et al.*, OpenQASM 3: A broader and deeper quantum assembly language (2021), arXiv:2104.14722 .

[263] T. Nguyen, A. Santana, T. Kharazi, D. Claudino, H. Finkel, and A. McCaskey, Extending C++ for Heterogeneous Quantum-Classical Computing (2020), arXiv:2010.03935 .

[264] D. A. Lidar, Lecture Notes on the Theory of Open Quantum Systems (2019), arXiv:1902.00967 .

[265] H. J. Carmichael, *Statistical methods in quantum optics 2: Non-classical fields* (Springer Science & Business Media, 2009).

[266] F. Minganti, N. Bartolo, J. Lolli, W. Casteels, and C. Ciuti, *Exact results for Schrödinger cats in driven-dissipative systems and their feedback control*, Scientific Reports **6**, 26987 (2016).

[267] Y. Tanimura and R. Kubo, *Time evolution of a quantum system in contact with a nearly Gaussian-Markoffian noise bath*, Journal of the Physical Society of Japan **58**, 101 (1989).

[268] D. Loss and D. P. DiVincenzo, *Quantum computation with quantum dots*, Physical Review A **57**, 120–126 (1998).

[269] B. E. Kane, *A silicon-based nuclear spin quantum computer*, Nature **393**, 133 (1998).

[270] M. H. Devoret and R. J. Schoelkopf, *Superconducting circuits for quantum information: An outlook*, Science **339**, 1169 (2013).

[271] A. F. Kockum and F. Nori, in *Fundamentals and Frontiers of the Josephson Effect*, edited by F. Tafuri (Springer, 2019) p. 703.

[272] D. Maslov, G. Dueck, D. Miller, and C. Negrevergne, *Quantum circuit simplification and level compaction*, IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems **27**, 436 (2008).

[273] A. Javadi-Abhari, S. Patil, D. Kudrow, J. Heckey, A. Lvov, *et al.*, *ScaffCC: Scalable compilation and analysis of quantum programs*, Parallel Computing **45**, 2 (2015).

[274] T. Häner, D. S. Steiger, K. Svore, and M. Troyer, *A software methodology for compiling quantum programs*, Quantum Science and Technology **3**, 020501 (2018).

[275] T. Fösel, M. Y. Niu, F. Marquardt, and L. Li, Quantum circuit optimization with deep reinforcement learning (2021), arXiv:2103.07585 .

[276] T. S. Metodi, D. D. Thaker, A. W. Cross, F. T. Chong, and I. L. Chuang, in *Quantum Information and Computation IV*, edited by E. J. Donkor, A. R. Pirich, and H. E. Brandt (2006) p. 62440T.

[277] S. Sargaran and N. Mohammadzadeh, *SAQIP: A Scalable Architecture for Quantum Information Processors*, ACM Transactions on Architecture and Code Optimization **16** (2019).

[278] P. Murali, J. M. Baker, A. Javadi-Abhari, F. T. Chong, and M. Martonosi, in *Proc. 24th Int. Conf. Archit. Support Program. Lang. Oper. Syst.* (ACM, 2019) p. 1015.

[279] T. S. Metodi, D. D. Thaker, A. W. Cross, F. T. Chong, and I. L. Chuang, in *Defense and Security Symposium*, edited by E. J. Donkor, A. R. Pirich, and H. E. Brandt (Orlando (Kissimmee), FL, 2006) p. 62440T.

[280] G. G. Guerreschi and J. Park, *Two-step approach to scheduling quantum circuits*, Quantum Science and Technology **3**, 045003 (2018).

[281] D. D'Alessandro, *Introduction to Quantum Control and Dynamics* (Chapman & Hall/CRC, 2007).

[282] N. Khaneja, T. Reiss, C. Kehlet, T. Schulte-Herbrüggen, and S. J. Glaser, *Optimal control of coupled spin dynamics: Design of NMR pulse sequences by gradient ascent algorithms*, Journal of Magnetic Resonance **172**, 296 (2005).

[283] S. Machnes, U. Sander, S. J. Glaser, P. de Fouquières, A. Gruslys, *et al.*, *Comparing, optimizing, and benchmarking quantum-control algorithms in a unifying programming framework*, Physical Review A **84**, 022305 (2011).

[284] T. Caneva, T. Calarco, and S. Montangero, *Chopped random-basis quantum optimization*, Physical Review A **84**, 022326 (2011).

[285] C. Piltz, T. Sriarunothai, A. Varón, and C. Wunderlich, *A trapped-ion-based quantum byte with 10-5 next-neighbour cross-talk*, Nature Communications **5**, 4679 (2014).

[286] N. Khammassi, G. G. Guerreschi, I. Ashraf, J. W. Hogaboam, C. G. Almudever, and K. Bertels, cqasm v1. 0: Towards a common quantum assembly language (2018), arXiv:1805.09607 .

[287] M. Alam, A. Ash-Saki, and S. Ghosh, in *2020 Design, Automation & Test in Europe Conference & Exhibition (DATE)* (IEEE, Grenoble, France, 2020) pp. 686–689.

[288] T. Haug and M. S. Kim, Optimal training of variational quantum algorithms without barren plateaus (2021), arXiv:2104.14543 .

[289] A. B. Magann, C. Arenz, M. D. Grace, T.-S. Ho, R. L. Kosut, *et al.*, *From pulses to circuits and back again: A quantum optimal control perspective on variational quantum algorithms*, PRX Quantum **2**, 010101 (2021).

[290] P. Murali, D. C. Mckay, M. Martonosi, and A. Javadi-Abhari, in *Proceedings of the Twenty-Fifth International Conference on Architectural Support for Programming Languages and Operating Systems*, ASPLOS '20 (Association for Computing Machinery, New York, NY, USA, 2020) pp. 1001–1016.

[291] Erik, L. Saldyt, Rob, tjproct, J. Gross, *et al.*, pyGSTio/pyGSTi: Version 0.9.10 (2021).

[292] A. Kandala, K. Temme, A. D. Córcoles, A. Mezzacapo, J. M. Chow, and J. M. Gambetta, *Error mitigation extends the computational reach of a noisy quantum processor*, Nature **567**, 491-495 (2019).

[293] T. Giurgica-Tiron, Y. Hindy, R. LaRose, A. Mari, and W. J. Zeng, in *IEEE Int. Conf. Quantum Comput. Eng. (QCE)* (2020) p. 306.

[294] R. LaRose, A. Mari, S. Kaiser, P. J. Karalekas, A. A. Alves, *et al.*, *Mitiq: A software package for error mitigation on noisy quantum computers*, arXiv preprint (2021), arXiv:2009.04417 .

[295] M. L. Dahlhauser and T. S. Humble, *Modeling noisy quantum circuits using experimental characterization*, Physical Review A **103**, 042603 (2021).

[296] K. Schultz, G. Quiroz, P. Titum, and B. D. Clader, *SchWARMA: A model-based approach for time-correlated noise in quantum circuits*, Physical Review Research **3**, 033229 (2021).

[297] S. Humpohl, L. Prediger, pcerf, P. Bethke, A. Willmes, *et al.*, qutech/qupulse: qupulse 0.6 (2021).

[298] N. Wittler, F. Roy, K. Pack, M. Werninghaus, A. S. Roy, *et al.*, *Integrated tool-set for Control, Calibration and Characterization of quantum devices applied to superconducting qubits*, Physical Review Applied **15**, 034080 (2021).

[299] B. Skinner, J. Ruhman, and A. Nahum, *Measurement-induced phase transitions in the dynamics of entanglement*, Physical Review X **9**, 031009 (2019).

[300] D. Dong, C. Chen, B. Qi, I. R. Petersen, and F. Nori, *Robust manipulation of superconducting qubits in the presence of fluctuations*, Scientific Reports **5**, 7873 (2015).

[301] D. Dong, C. Wu, C. Chen, B. Qi, I. R. Petersen, and F. Nori, *Learning robust pulses for generating universal quantum gates*, Scientific Reports **6**, 36090 (2016).

[302] S. Gustavsson, O. Zwier, J. Bylander, F. Yan, F. Yoshihara, *et al.*, *Improving Quantum Gate Fidelities by Using a Qubit to Measure Microwave Pulse Distortions*, Physical Review Letters **110**, 040502 (2013).

[303] S. Krastanov, S. Zhou, S. T. Flammia, and L. Jiang, *Stochastic estimation of dynamical variables*, Quantum Science and Technology **4**, 035003 (2019).

[304] V. E. Manucharyan, J. Koch, L. I. Glazman, and M. H. Devoret, *Fluxonium: Single Cooper-Pair Circuit Free of Charge Offsets*, Science **326**, 113–116 (2009).

[305] F. Bao, H. Deng, D. Ding, R. Gao, X. Gao, *et al.*, Fluxonium: An alternative qubit platform for high-fidelity operations (2021), arXiv:2111.13504 [quant-ph] .

[306] A. M. Forney, S. R. Jackson, and F. W. Strauch, *Multifrequency control pulses for multilevel superconducting quantum circuits*, Physical Review A **81**, 012306 (2010).

[307] J. Ikonen, J. Goetz, J. Ilves, A. Keränen, A. M. Gunyho, *et al.*, *Qubit Measurement by Multichannel Driving*, Physical Review Letters **122**, 080503 (2019).

[308] E. Barnes, F. A. Calderon-Vargas, W. Dong, B. Li, J. Zeng, and F. Zhuang, *Dynamically corrected gates from geometric space curves*, Quantum Science and Technology **7**, 023001 (2022).

[309] B.-J. Liu, Y.-S. Wang, and M.-H. Yung, *Super-robust nonadiabatic geometric quantum control*, Physical Review Research **3**, L032066 (2021).

[310] K. Yi, Y.-J. Hai, K. Luo, L. Zhang, Y. Zhou, *et al.*, Robust Quantum Gates against Correlated Noise in Integrated Quantum Chips (2024), arXiv:2401.01810 [quant-ph] .

Acknowledgement

First and foremost, I would like to express my deepest gratitude to Felix and Tommaso for your invaluable guidance and feedback over the past four years. It was through your mentorship that I was introduced to the world of quantum control. A special thanks to Felix for your unwavering support in so many aspects, for teaching me the magic behind DRAG control, and for giving me the freedom to explore my ideas while guiding me through the challenges and excitement of this research journey.

I am sincerely grateful to Prof. Rami Barends and Prof. Erwann Bocquillon for graciously agreeing to join my examination committee. Your time and expertise are greatly appreciated.

My heartfelt thanks go to Francisco, Eloisa and José for their insightful discussions and valuable input during the formulation of this thesis. A special thanks to Francisco, who patiently taught me the proper way of modelling superconducting qubits after my years of playing with three-level models.

Special thanks also go to Sandeep, José, Vidisha, and Dimitrios for the stimulating discussions on the DRAG methods and their potential applications. These conversations have sparked many interesting ideas, some of which are reflected in the discussion section of this thesis.

I am deeply appreciative of Francesco, Niklas, Thomas, Akshay, Jan, Juhi, Zsolt and the entire team PGI-8 of Forschungszentrum Jülich. These past four years of PhD life, despite beginning with the challenges of a "digital life" during the pandemic, have been deeply rewarding thanks to your camaraderie. You have all contributed to making this journey an invaluable experience.

I would also like to express my gratitude to the QuTiP admin team, Nathan, Neill, Simon, Alex, Shahnawaz, Eric, and, Franco, for the continued collaboration on the QuTiP project. It is through your collective efforts that QuTiP continues to thrive and gain widespread attention and adoption in the scientific community. QuTiP also marked my first venture into coding and collaboration, and the first lesson in quantum dynamics.

My sincere thanks also go to Ruixia and Fei for the valuable discussions on the crosstalk problem. And to Adrian for our engaging conversations on the qudit control methods, as well as for your assistance in resolving several experimental challenges during the calibration of IBM hardware. Also a special mention to my master thesis supervisors Tim and David, you were the first to teach me how to write an academic article. Though our conversations have been few in recent years, Tim your thesis remains a constant reference and source of inspiration as I was writing mine.

I acknowledge the use of IBM Quantum Platforms for the experimental part of this work. The views expressed are those of the authors and do not reflect the official policy or position of IBM or the IBM Quantum team.

Lastly, I owe my deepest thanks to my fiancée for her unwavering companionship and to my parents, who, despite being separated by continents, have supported me wholeheartedly throughout these years. Your love and encouragement have been a constant source of strength, and I could not have come this far without you.

List of Publications

6. **Boxi Li**, F. A. Càrdenas-López, Adrian Lupascu, and Felix Motzoi, *Universal Pulses for Superconducting Qudit Ladder Gates*, arXiv:2412.18339 (2024).

5. **Boxi Li**, Tommaso Calarco, and Felix Motzoi, *Experimental error suppression in Cross-Resonance gates via multi-derivative pulse shaping*, npj Quantum Information 10.1 (2024): 66.

4. **Boxi Li**, Tommaso Calarco, and Felix Motzoi, *Nonperturbative Analytical Diagonalization of Hamiltonians with Application to Circuit QED*, PRX Quantum 3.3 (2022): 030313.

3. **Boxi Li**, Shahnawaz Ahmed, Sidhant Saraogi, Neill Lambert, Franco Nori, Alexander Pitchford, Nathan Shammah, *Pulse-level noisy quantum circuits with QuTiP*, Quantum 6 (2022): 630.

2. **Boxi Li**, Tim Coopmans, and David Elkouss, *Efficient optimization of cutoffs in quantum repeater chains*, IEEE Transactions on Quantum Engineering vol. 2, pp. 1-15, 2021, Art no. 4103015: 1-15.

1. Koji Azuma, Stefan Bäuml, Tim Coopmans, David Elkouss, and **Boxi Li**, *Tools for quantum network design*, AVS Quantum Sci. 3, 014101 (2021).

Curriculum vitae

10.2020 - 10.2024	Research Center Jülich and University of Cologne
	PhD candidate
	Supervisor: Prof. Tommaso Calarco and Prof. Felix Motzoi
10.2019 - 08.2020	QuTech, Delft University of Technology
	Exchange master thesis
	Supervisor: Prof. David Elkouss and Dr. Tim Coopmans
09.2018 - 09.2020	Swiss Federal Institutes of Technology (ETHZ)
	Master of Science in Physics
09.2015 - 07.2018	Heidelberg University
	Bachelor of Science in Physics
	Supervisor: Prof. Thomas Gasenzer & Prof. Martin Gärttner

Band / Volume 276
Strain- and process engineering for polyketides production with *Pseudomonas taiwanensis* VLB120 in two-phase cultivations
T. P. Schwanemann (2023), 230 pp
ISBN: 978-3-95806-726-4

Band / Volume 277
Quantitative atomic-level investigation of solid materials through multidimensional electron diffraction measurements
H. L. Lalandec-Robert (2024), xxi, 152 pp
ISBN: 978-3-95806-735-6

Band / Volume 278
Studies on the cAMP-responsive regulatory network of *Corynebacterium glutamicum*
N. Wolf (2024), iii, 122 pp
ISBN: 978-3-95806-736-3

Band / Volume 279
Rare-earth atoms on two-dimensional materials: ab initio investigation of magnetic properties
J. P. Carbone (2024), 235 pp
ISBN: 978-3-95806-740-0

Band / Volume 280
Communities of Niche-optimized Strains (CoNoS) – a novel concept for improving biotechnological production
R. Zuchowski (2024), VIII, 168 pp
ISBN: 978-3-95806-743-1

Band / Volume 281
Enabling mixed microbial upcycling of plastic monomers
Y. S. Ackermann (2024), XVI, 203 pp
ISBN: 978-3-95806-749-3

Band / Volume 282
Folding and structural studies of *saccharomyces cerevisiae* Phosphoglycerate Kinase
N. Bustorff (2024), xxvi, 126 pp
ISBN: 978-3-95806-754-7

Band / Volume 283
The role of cellular development in multicellular antiphage defense of *Streptomyces*
T. Luthe (2024), vi, 173 pp
ISBN: 978-3-95806-768-4

Band / Volume 284
Probing the Transformation from Transition Metal Complexes to Extended Two-Dimensional Nanostructures
D. Baranowski (2024), XII, 103 pp
ISBN: 978-3-95806-772-1

Band / Volume 285
Neutron Scattering
Lectures of the JCNS Laboratory Course held at Forschungszentrum Jülich and at the Heinz-Maier-Leibnitz Zentrum Garching
edited by S. Förster, K. Friese, M. Kruteva, S. Nandi, M. Zobel, R. Zorn (2024), ca. 365 pp
ISBN: 978-3-95806-774-5

Band / Volume 286
***Ab initio* investigation of intrinsic antiferromagnetic solitons**
Amal Jawdat Nayef Aldarawsheh (2024), xv, 164 pp
ISBN: 978-3-95806-785-1

Band / Volume 287
Understanding the dynamics of Plant-Bacteria-Bacteriophage interactions as a means to improve plant performance
S. H. Erdrich (2024), ix, 176 pp
ISBN: 978-3-95806-791-2

Band / Volume 288
Prediction of Magnetic Materials for Energy and Information Combining Data-Analytics and First-Principles Theory
R. Hilgers (2024), xv, 215 pp
ISBN: 978-3-95806-795-0

Band / Volume 289
Biodegradation and microbial upcycling of plastics
J. de Witt (2025), XVI, 259 pp
ISBN: 978-3-95806-804-9

Band / Volume 290
Practical Methods for Efficient Analytical Control in Superconducting Qubits
B. Li (2025), 202 pp
ISBN: 978-3-95806-807-0

Weitere ***Schriften des Verlags im Forschungszentrum Jülich*** unter
http://wwwzb1.fz-juelich.de/verlagextern1/index.asp

Schlüsseltechnologien / Key Technologies
Band / Volume 290
ISBN 978-3-95806-807-0

Mitglied der Helmholtz-Gemeinschaft

JÜLICH
Forschungszentrum