# Control of Many-body Quantum Systems

Aberystwyth University

1872 PRIFYSGOL
ABERYSTWYTH
UNIVERSITY

Alexander James Gordon Pitchford

14 June 2019

# Abstract

The scaling of effort to achieve control objectives with system size is an important consideration in the development of many quantum devices, especially in those intended for quantum information processing. This thesis investigates the scaling with system size of effort required to control many-body quantum systems, and asks whether this is sufficiently favourable to enable supremacy in quantum computation. This is tested using quantum control theory and numerical control optimisation, which are well established tools for investigating the dynamics of driven quantum systems, in a variety of theoretical models related to state-of-the-art quantum devices. Gates acting on a chain of coupled oscillators are found to achievable in times that scale approximately linear with chain length, where the control is over only a single oscillator. The scalability of a scheme for implementing quantum gates in a many-body quantum simulator is shown to be favourable enough to allow for them to be configured to perform quantum computations. A state measurement protocol is proposed as part of the scheme. The scalability is validated through numerical simulation of simulator models composed of up to 9 qubits. The reachable set of operations for systems with quadratic Hamiltonians and infinite-dimensional Hilbert spaces is investigated. The passive operations in quantum optics are proven to be unreachable for single-mode systems with 'unstable' Hamiltonians. Further characterisation of the reachable set is made through numerical simulation of control optimisation. Some progress is made in extending the reachability result to $n$-modes. A non-Markovian noise model is used in simulating control optimisation of a dynamical decoupling scheme, which would be impossible to simulate with Markovian dynamics. The popular QuTiP Python library has been extended to allow simulation of control optimisation with a range of dynamical models. A description of these software modules and their method of use is given. Many of the software tools developed for the study are made available through open-source repositories. Some outlook is given for the use of *in-situ* open-loop control in optimising controls in quantum system experiments.

# Declaration

| | |
|---|---|
| Word Count of thesis:<br>DECLARATION | 76,422 |
| This work has not previously been accepted in substance for any degree and is not being concurrently submitted in candidature for any degree. | |
| Candidate name | Alexander James Gordon Pitchford |
| Signature: | |
| Date | 14 June 2019 |

**STATEMENT 1**

This thesis is the result of my own investigations, except where otherwise stated. Where ***correction services** have been used, the extent and nature of the correction is clearly marked in a footnote(s).

Other sources are acknowledged by footnotes giving explicit references. A bibliography is appended.

| | |
|---|---|
| Signature: | |
| Date | 14 June 2019 |

[*this refers to the extent to which the text has been corrected by others]

**STATEMENT 2**

I hereby give consent for my thesis, if accepted, to be available for photocopying and for inter-library loan, and for the title and summary to be made available to outside organisations.

| | |
|---|---|
| Signature: | |
| Date | 14 June 2019 |

# Acknowledgements

I have much to thank my supervisor Daniel Burgarth for in relation to this PhD. Firstly, his support in the application for the award that has funded this study. He has allowed me the freedom to investigate quantum control theory within the loose bounds of the question of scalability in many-body systems, but provided the right motivation and direction when needed, and helped me to extend the bounds of my understanding. I am especially grateful for his encouragement as the study came towards its end. The many interesting projects I have had opportunity to collaborate in are mainly thanks to Daniel's connections, and their successful outcomes are largely due to his direction. Working with my close collaborators Uther Shackerley-Bennet, Ben Dive and Neill Lambert has been a very rewarding and fruitful experience. Our complementary skills and collaborative efforts have produced some very satisfying results

I am grateful to Prof. Tomasso Calarco for inviting me to Ulm, and my hosts there Prof. Simone Montangero, Jonathan Zoller and Ressa Said. This was good opportunity to understand the practical considerations required when performing actual control experiments. I am also very grateful to Dr Franco Nori for twice inviting me to RIKEN, Japan, and to my hosts there Neill and Mauro Cirio for our discussions, and generally entertaining me. My Japanese experiences were most memorable, and have led to what I hope will be long lasting collaborations. Thank you to Robert Johansson and Paul Nation for inviting me to join QuTiP team, which has since provided many opportunities.

I am grateful for the HPC-Wales project, which provided the computing power that underpins many of the results herein. I am grateful to Aberystwyth University for the award to has supported me and this project.

I would like to thank all my friends and colleagues who have helped, support, encouraged and entertained me throughout the last four years. A few people deserve a specific mention. Kim Kenobi for our interesting discussions generally, and his advice on writing and staying 'in the moment'. His example of positivity in the face of adversity has helped a lot in completing the thesis. Thank you to my colleague Jukka Kiukas for a scientific and general discussions, including help with understanding the symplectic dynamics. My office mate for the first few years, and continued friend, Christian Arenz, who helped me with understanding many aspects of quantum control theory. Thank you to all the academic staff, fellow demonstrators, and students I had opportunity to work with when helping with undergraduate teaching. This has been an important part of the experience for me, which I found very rewarding. I am grateful for all the administration team and technical support staff, in my institute and at RIKEN, who have always made every effort to help whenever I needed it.

A special mention for my fellow travellers through this PhD journey, Flavia, Gennaro, Francesca, Nikoli, Azam, Claire and Tirion. Despite our different fields of study, we have had opportunity to be involved in each other's projects, discussing ideas, and also having a lot of fun. Additional thanks to Claire who provided feedback on some parts of this thesis. Also my close friends

Danny & Janet, Dale & Debs, and Francesca, who have provided much support and encouragement throughout the period of the study, especially in the closing stages. I am always grateful for my parents, David and Lesley, and sisters, Becky and Jenny, who have supported and encouraged me all along, including this PhD experience.

# Contents

# List of Figures

# List of Tables

# Chapter 1

# Introduction

## 1.1 Overview

This thesis asks whether control of many-body, high-dimensional quantum systems can be achieved with acceptable scalability for quantum information processing. This is tested by applying the theories and optimisation algorithms that have been used successfully for achieving control objec-

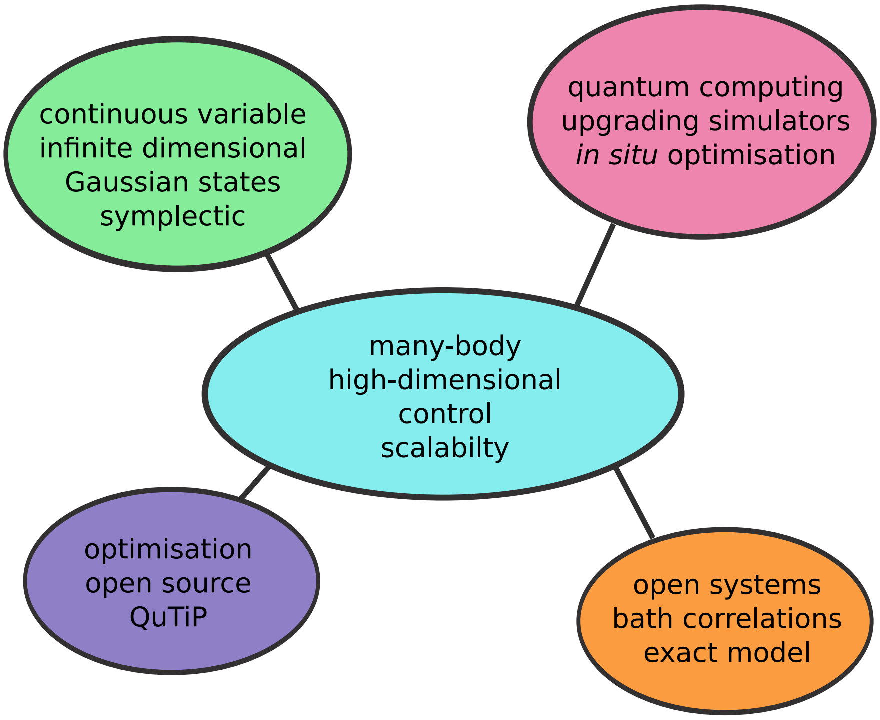


**Figure 1.1:** The central theme of this thesis is shown in context with the four main research areas that seek to answer it.

tives on low-dimensional systems to simulations of systems of increasing size, up to the limit of the available high-performance computing clusters, with various models of quantum systems and control objectives. In particular the scaling of the minimum time required to perform operations (gates) on the systems is investigated, as this is one of the limiting factors on performing a quantum computation.

Quantum control developed out of the nuclear magnetic resonance (NMR) community, where it was used to improve the radio frequency pulses that drive the states of the molecules under study [1, 2], and quantum chemistry, where the objective was to drive specific chemical reactions [3, 4]. Today it is part of the core toolset used by engineers developing state-of-the-art quantum technology [5, 6]. Experiments with actual quantum systems are typically challenging to perform, and often require expensive equipment, as they typically need cryogenic temperatures when working with constrained energy level systems, such ion-traps and superconducting circuits, or the vast arrays of optical devices in quantum optics experiments. This makes simulation a vital tool for investigating quantum systems, including the theoretical control of their dynamics.

Technological developments tend to arise from fundamental research, as technologists are always looking to new findings for opportunities. However, there is currently a strong, direct push for the quantum phenomena to be utilised in devices that will benefit the wider society. The UK has invested heavily in the Quantum Technology Hubs with this intention. Some of these quantum devices are now commercial products (quantum key distribution), and many more are in prototype stage (high precision sensors) – these are described later in this chapter. The realisation of a *useful* universal quantum computer (defined later in this chapter) however remains elusive. To be useful, a quantum computer must be composed of many elements, and hence this thesis.

The majority of research into developing quantum computers has been based on so-called spin-computers [7, 8], which are described later in this chapter. Computers based on continuous variable (CV) quantum systems, such as harmonic oscillators and electromagnetic field modes, also have potential [9]. Gates acting on the mode operators of these systems in Gaussian states can be represented by symplectic matrix transformations. Strong controllability criterion have been derived for these systems [10, 11, 12, 13]. However, the criteria may not be manifest in the system interest, raising the question of what is possible when these criteria are not met. In one type of example the Hamiltonians result in non-recurring dynamics, and are hence described as ‘unstable’. The characterisation of the set of possible operations for systems with unstable Hamiltonians is addressed in Chapter 3. The first study to use control optimisation with symplectic transformations found that the allowed gate time was crucial in achieving high fidelity [14]. Chapter 3 investigates the scalability of implementing CV gates. The minimum time required to perform an entangling transformation on chains of harmonic oscillators of increasing length is found through simulation of control optimisation.

The viability of classical simulation the dynamics of many-body quantum systems is limited by the scaling of the Hilbert space in which quantum systems are represented (except in some specific cases where symmetries can be exploited). This scaling is with $2^n$ for composites of $n$ two-state systems. Therefore, although quantum computing algorithms can be simulated on a classical computer, the computation cost grows exponentially with system size. As the quantum computer can solve some problems exponentially faster with greater $n$, then for some size of system it will out-perform its classical counterpart, so long as the effort to set up the problem scales only polynomially in $n$. Large, coupled arrays of controllable quantum systems have been developed specifically for simulating quantum dynamics. These *quantum simulators* are described later in this chapter. A scheme is proposed in Chapter 4 for upgrading these quantum simulators to act as quantum computers. The scheme uses control optimisation to determine the parameters that implement specific gates. The chapter focuses on demonstrating that the scheme scales no worse than polynomially with $n$.

The quantum interference phenomena are what underpins most of quantum technologies being developed. The constructive and destructive interference relies on the coherence of component quantum systems. Interactions with the surrounding environment destroy this coherence, and it is impossible to fully isolate a system from its environment. Consequently, understanding exactly how these interactions affect system, and how to negate or even take advantage of them, has been the focus of much research over the last decades. This research may also lead to a fuller understanding of quantum theory, especially the ‘measurement problem’, and the non-observance of macroscopic superpositions [15], as essentially the macroscopic world is a very big and noisy quantum system. A noise model is therefore needed for calculating quantum dynamics. Noise models for simulating quantum dynamics, and in particular simulation of control optimisation, are the subject of Chapter 5. The method for calculating quantum dynamics, using hierarchical equations of motion, that does not make the Born-Markov approximation is compared with the Lindblad master equation, which does make this approximation. A control objective is attempted that is impossible with the Born-Markov approximation, as it requires the return of quantum information from the environment, which implies memory.

Many of the results presented in this thesis are found through numerical simulation. Some of the methods are fairly detailed and technical, and so these are described separately in Chapter 2. The main aim of the chapter is assist anyone wishing to reproduce or extend the results, and may be skipped over by readers interested only in the physics. The codes for performing quantum control optimisation are described, with references to the Python functions used where this is appropriate. Some description of the computer software and systems used is also given. The software tools developed by the author to support these methods have largely be made available in open-source libraries. In particular, the author has made many contributions to the QuTiP project [16, 17] during the period of this study. The codes used to produce unpublished results are not publicly available, but they are available in private repositories for sharing with anyone wishing to collaborate.

The thesis is organised as follows. The rest of this chapter introduces the concepts and mathematical formulae that are used throughout the thesis. A brief outline of quantum mechanics is given in Section 1.2. Quantum control theory is introduced in Section 1.3, with controllability of quantum systems specifically in Section 1.4. Control function optimisation for quantum system is described in detail in Section 1.6. The last section of this chapter, Section 1.5, gives a brief description of the quantum technologies, for which quantum control is either a key component, or vital in their development. The chapters outlined above follow, each of them starting with an introduction outlining the key developments, concepts and methods which are specific to them, and conclude with some discussion of the findings. Finally, Chapter 6 makes some overall conclusion, and gives some outlook for control of many-body quantum systems.

## 1.2 Quantum Mechanics

In order to predict the behaviour of microscopic entities they need to be described in terms of their wavelike properties. The properties of the wave are related to the mass and velocity in a way that makes them insignificant for the macroscopic particles we encounter around us – the wavelength being a tiny fraction of the length scales of interest. There are many physical interpretations of these waves and this continues to be a subject of much discussion and controversy [15, 18]. The equations proposed based on the premise of their existence that predict the behaviour of microscopic particles are confirmed by experiment to the limit of precision. This is one the primary objective of physics – to understand the fundamentals of nature sufficiently to predict how systems will evolve. This theory of *quantum mechanics* is considered one the most successful or all time, having withstood scrutiny for over 100 years.

A solid introduction to quantum mechanics and its mathematical description in the context of control theory is given by Domenico D'Alessandro in his book Ref. [19]. Similarly, the popular reference for quantum computing and information Ref. [20], also provides a detailed background on the fundamental concepts and mathematics of quantum mechanics. These are the main sources for the brief introduction given below. In this thesis units are chosen such that the reduced Planck constant $\hbar = 1$, and Boltzmann's constant $k_{\mathrm{B}} = 1$.

A system that is considered isolated from its surroundings is called *closed*. The pure state of a closed quantum system is represented by a vector $|\psi\rangle$ in a complex Hilbert space $\mathcal{H}$. The direction in this Hilbert space is what identifies the state, and states differing only by a difference global phase factor are considered the same state. As the phase and magnitude can be ignored for state identification, then $\langle\psi|\psi\rangle = \||\psi\rangle\|^2 = 1$, $|\psi\rangle$ is therefore called a *normalised state vector*. The relative phase is important it determining how quantum systems will interact, due to constructive and destructive interference, which is why the Hilbert space is complex. As any linear combination of these states is also a solution to Schrödinger's equation (defined shortly), then this is also a valid quantum state, known as a *superposition state*.

The state of a composite system is represented by a vector on a tensor product of their Hilbert spaces. That is, if two systems are separately defined on spaces $\mathcal{H}_A$ and $\mathcal{H}_B$, then the state of the composite system can be represented by a vector on the space $\mathcal{H}_A \otimes \mathcal{H}_B$. If the state of the composite system can be given by a tensor product, that is $|\psi_{AB}\rangle = |\psi_A\rangle \otimes |\psi_B\rangle$, then it is a *product state* and is described as *separable*. Otherwise it is *inseparable*, and described as *entangled*.

Systems that are not isolated from their surroundings are called *open*. In this case the state of the system can become *mixed*. This is expanded upon below in Subsection 1.2.3. A state vector cannot fully describe a mixed state, but it can be described by a *density operator* or *density matrix*. This is first introduced here in terms of an ensemble of identical systems, for which the state, in terms of measurement statistics, can also be described by a *density operator*

$$\rho := \sum_{j}^{N_w} w_j |\psi_j\rangle\langle\psi_j| \,. \tag{1.1}$$

Here the ensemble comprises systems in one of $N_w$ states $|\psi_j\rangle$ in proportions $w_j$ ($0 < w_j \leq 1$, $\sum_j w_j = 1$). Hence $w_j$ is the probability that any one system is in state $|\psi_j\rangle$. In the special case that all systems are in the same state, that is one $w_j = 1$, then this is referred to as a *pure ensemble* or the ensemble is said to be in a *pure state*, and $\rho = |\psi_j\rangle\langle\psi_j|$. From this definition it follows that $\rho$ has the following properties:

$$\operatorname{Tr} \rho = 1 \tag{1.2a}$$

$$\rho^\dagger = \rho \tag{1.2b}$$

$$\rho \geq 0 \,. \tag{1.2c}$$

That is the density operator has trace zero, is self-adjoint and positive. Any operator that has these properties is a valid quantum state.

Another important property of the density operator is that $\rho = \rho^2$ if and only if the system or ensemble is in a pure state. Any ensemble state that is not pure is called a *mixed state*. A mixed state can also occur as the reduced state of a subsystem of any composite system, including in the open systems model described in Subsection 1.2.3. For states described by density matrices, they are called *separable* if they can be written as sum of product states on respective spaces $\mathcal{H}_A$ and $\mathcal{H}_B$, that is

$$\rho^{(AB)} = \sum_{j}^{N_w} w_j \rho_j^{(A)} \otimes \rho_j^{(B)} \,. \tag{1.3}$$

If the density matrix is not separable, then the state is entangled.

### 1.2.1 Quantum system evolution

The dynamics of the state are described *Schrödinger's equation*

$$\frac{\mathrm{d}}{\mathrm{d}t}|\psi\rangle = -\mathrm{i}H|\psi\rangle \tag{1.4}$$

where $H$ is the Hamiltonian operator, which could be time dependent, and is always self-adjoint, *i.e.* $H^\dagger = H$. Here is implied that $|\psi\rangle$ evolves in time $t$. As this is a linear first order differential equation the solution is given by

$$|\psi(t)\rangle = U(t)|\psi(0)\rangle, \quad U(0) = \mathbb{I}\,, \tag{1.5}$$

where $|\psi(0)\rangle$ is the initial state of the system, that is when $t = 0$. The *propagator* $U(t)$ is a unitary operator due to $H$ being Hermitian. It is also called the *evolution operator*. Substituting Eq. (1.5) into Eq. (1.4) shows that $U(t)$ must evolve according to what is called the *Schrödinger operator equation*

$$\dot{U}(t) = -\mathrm{i}H(t)U(t)\,. \tag{1.6}$$

The equation of motion for a quantum system in terms of its density operator $\rho$ is given by the *Liouville-von Neumann* equation,

$$\dot{\rho} = -\mathrm{i}[H, \rho]\,, \tag{1.7}$$

the solution of which describes the evolution of $\rho$ according to the Schrödinger equation,

$$\rho(t) = U\rho(0)U^\dagger = \mathcal{U}_t[\rho(0)]\,. \tag{1.8}$$

where the superoperator $\mathcal{U}_t$ is the map that describes the adjoint action of $U$ on $\rho$ from time $0$ to $t$.

### 1.2.2 Qubits

An important mathematical construct in *quantum information* (which will be discussed later in Section 1.5) is the *quantum bit* or *qubit*. This is based on the concept of a quantum system with only two states, typically denoted $|0\rangle$ and $|1\rangle$. Because it is a quantum system, then any linear combination (with the usual normalisation), is also a valid quantum state. That is

$$|\psi\rangle := \alpha|0\rangle + \beta|1\rangle\,, \tag{1.9}$$

with $\alpha, \beta \in \mathbb{C}$ and $|\alpha|^2 + |\beta|^2$, due to $\||\psi\rangle\|^2 = 1$. This can also be given a geometrical representation

$$|\psi\rangle := \mathrm{e}^{\mathrm{i}\gamma}\left(\cos(\tfrac{\theta}{2})|0\rangle + \sin(\tfrac{\theta}{2})\mathrm{e}^{\mathrm{i}\phi}|1\rangle\right)\,, \tag{1.10}$$

where $\gamma, \theta, \phi \in \mathbb{R}$. Any two states differing only in global phase $\gamma$ are physically indistinguishable, hence this term is typically ignored. The $\alpha$ of Eq. (1.9) can be made real by shifting the phase of $\beta$, making Eq. (1.10)

$$|\psi\rangle := \cos(\tfrac{\theta}{2})|0\rangle + \sin(\tfrac{\theta}{2})\mathrm{e}^{i\phi}|1\rangle\,. \tag{1.11}$$

This geometrical representation defines a point on what is known as the *Bloch sphere*, shown in Fig. 1.2. Any point on the surface represents a pure state, with the north and south pole being the canonical *computational basis* states, other surface points being superpositions in this basis, and points inside the sphere being mixed states. Note that the definition in terms of $\theta/2$ means that a full rotation about any axis in the $x - y$ plane is a given by $\theta = 4\pi$, whereas a full rotation about $z$-axis is given by $\phi = 2\pi$.

There is much interest in physical systems that can be used to represent a qubit, because of the usefulness of the qubit in quantum computing. Examples of these two states are spin angular momentum of a spin-½ particle, the energetic states of a two-level atom, and the polarisation of photons.

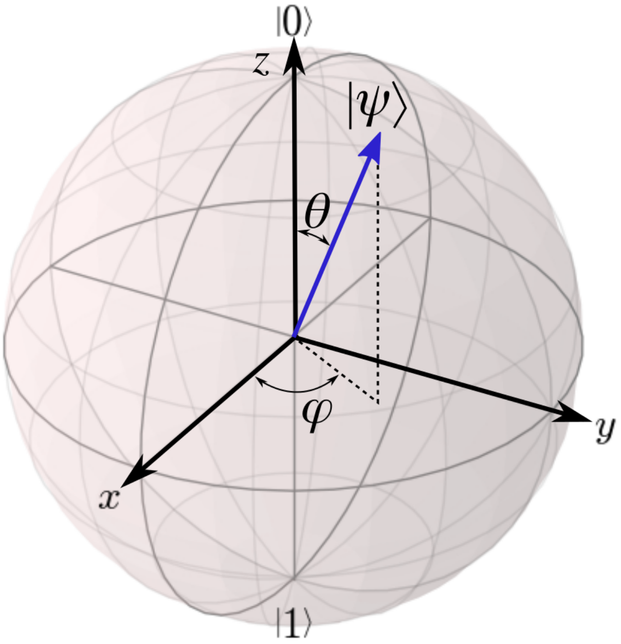


**Figure 1.2:** The Bloch sphere is a geometrical representation of the qubit state. The angles $\theta$ and $\phi$ are spherical coordinates on the unit sphere. The basis states are shown $|0\rangle$ uppermost and $|1\rangle$ lowermost.

The Pauli spin matrices have a variety of applications with qubits. They are defined as

$$\sigma_1 \equiv \sigma_x = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} , \quad \sigma_2 \equiv \sigma_y = \begin{pmatrix} 0 & -\mathrm{i} \\ \mathrm{i} & 0 \end{pmatrix} , \quad \sigma_3 \equiv \sigma_z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} , \tag{1.12}$$

They act as measurement operators or *observables*. Defining the computation basis vectors as

$$|0\rangle := \begin{pmatrix} 1 \\ 0 \end{pmatrix} , \quad |1\rangle := \begin{pmatrix} 0 \\ 1 \end{pmatrix} , \tag{1.13}$$

the expectation value given by $\langle\psi|\sigma_j|\psi\rangle$, where $\sigma_j$ is the $j^{\text{th}}$ component of the vector $\vec{\sigma} = (\sigma_x, \sigma_y, \sigma_z)$, is the coordinate in the corresponding axis of the Bloch sphere. For example $\langle 0|\sigma_z|0\rangle = 1$, which would typically correspond to the excited state of a two-level system or positive spin (spin-up) for a spin-½ particle. With the $-\mathrm{i}$ pre-factor they are skew-Hermitian and form a basis for the special unitary Lie algebra $\mathfrak{su}(2)$, and hence exponentiate to generate the special unitary Lie group

SU(2). When acting as a Hamiltonian on a qubit system, that is $H = \sigma_j$ in Eq. (1.4), hence $|\psi(t)\rangle = \exp(-\mathrm{i}\sigma_j t)|\psi(0)\rangle$, then they drive rotations around the corresponding Bloch sphere axis with period $2\pi$.

### 1.2.3 Open quantum systems

All the description of quantum mechanics above assumes that the system is completely detached from its surroundings, that is, a *closed system*. In reality this is practically impossible to achieve, and so it is necessary to have to way to model the effects of interactions with the environment on the system of interest. In this it is typical to consider a system of interest and the environment interacting through fields. This is referred to as the study of *open quantum systems*, for which textbooks Refs. [21, 22] are used as reference. It is partially analogous to the study of thermodynamics where the interactions are the flow of heat and entropy. The most interesting features of quantum systems relate to interference phenomena, which rely on the coherence of the ensemble, and so as well as the dissipation of heat between the system and its environment, the decoherence effects of the system-environment interactions are also of great importance.

Typically the study of open systems works from the idea that one can model the environment as another quantum system, such that the system and its environment together can be considered one closed system. To do this a model for the immediate environment is chosen that is unaffected by its own surroundings. Again borrowing from thermodynamics, the idea of a heat bath in thermal equilibrium is given a quantum mechanical description such that any interactions with the outside can be absorbed, and those with the system can be modelled as a stochastic process, meaning fully randomised. The combined system then must evolve according to Schrödinger's equation and hence the combined system dynamics are unitary. So the system state $\rho_S$ is defined on a Hilbert space $\mathcal{H}_S$, the state of the bath $\rho_B$ on $\mathcal{H}_B$, and the combined system state $\rho_{SB}$ on $\mathcal{H}_{SB} = \mathcal{H}_S \otimes \mathcal{H}_B$. The evolution then is described by

$$\rho_{SB}(t) = U\rho_{SB}(0)U^\dagger = \mathcal{U}[\rho_{SB}(0)]\,, \tag{1.14}$$

where $U$ is a unitary operator (and $\mathcal{U}$ the superoperator) acting on the composite space $\mathcal{H}_{SB}$.

Typically only the evolution of the system is of interest. Assuming the $\rho_S$ starts as some pure state, what can be said about its state at some future time $t$ ? Of course without the environmental interactions it would remain in a pure state. The effects of the system-environment interactions are contained in the unitary $U$, but this is likely to be a huge operator and of little use. The state of the system (its density matrix) can be recovered through use of the partial trace.

For a general description of the partial trace, let a composite Hilbert space $\mathcal{H}_{AB} = \mathcal{H}_A \otimes \mathcal{H}_B$, then for a general operator $O$ acting on $\mathcal{H}_{AB}$, the partial trace can be defined as [21]

$$\mathrm{Tr}_B[O] = \sum_{j=1}^{d_B} \left(\mathbb{I}_A \otimes \langle b_j|\right) O \left(\mathbb{I}_A \otimes |b_j\rangle\right) \ , \tag{1.15}$$

where the $|b_j\rangle$ are an orthogonal basis set for $\mathcal{H}_B$. It follows that

$$\mathrm{Tr}_B[A \otimes B] = \mathrm{Tr}[B]\ A\,, \tag{1.16}$$

where $A, B$ are operators acting on the spaces $\mathcal{H}_A$ and $\mathcal{H}_B$ respectively, and in fact $\mathrm{Tr}_B$ is the unique function that provides this mapping.

Due to the $\mathrm{Tr}\,\rho = 1$ for all density operators $\rho$, the state of the system can be recovered as

$$\rho_S = \mathrm{Tr}_B[\rho_{SB}]\,. \tag{1.17}$$

If the evolution of the combined system thorough Eq. (1.14) results in entanglement between the system and bath through correlations, then the $\rho_S$ retrieved through Eq. (1.15) will be a mixed state, as quantum information is lost to the bath, and hence it is known as the *reduced state*. This is so even though the composite system remains in a pure state, due to the unitary evolution of Eq. (1.14), as effectively the state of the system is mixed with that of the bath. From a physical perspective, this means that the measurement outcome will be equivalent to that of an ensemble in the mixed state [20].

A method of describing the evolution of the system state alone would be preferred, that is

$$\rho_S(t) = M[\rho_S(0)]\,, \tag{1.18}$$

where $M$ is the *dynamical map* (or *quantum channel*) that describes the evolution of $\rho_S$. It turns out that $\rho_S$ will remain pure if $U$ can be written $U = U_S \otimes U_B$, that is the evolution of system and bath separately are unitary, and therefore $M$ would be a unitary map. However $\rho_S$ could become a mixed state, due to correlations between the system and bath, indicating a loss of energy and or coherence to the environment. So the map $M$ need not be unitary, but it must have properties to ensure that system and environment both remain in valid quantum states after its application. These stem from the need to preserve the properties of the density operator given in Eq. (1.2):

$$\operatorname{Tr} M(\rho) = 1 \tag{1.19a}$$

$$M(\rho)^\dagger = M(\rho) \tag{1.19b}$$

$$M \otimes \mathcal{I}_d(\rho \otimes \mathbb{I}_d) \geq 0\,. \tag{1.19c}$$

where $\rho$ is any density operator of the system. The first states that the map must preserve the trace of $\rho$ as $1$, the second that it must ensure $\rho$ remains self-adjoint. The last one states that positivity must be preserved, not only for the map itself, but also any ancilla system, such as the bath. The extension of the state to include the state of some ancilla of dimension $d$, which the map must act trivially on, expressed by the identity map $\mathcal{I}_d$, states that the map $M$ must be *completely positive*. This is sufficient for complete positivity so long as $d$ is at least equal to the dimension of the system.

Because of these properties dynamical maps in this context are called completely positive, trace preserving maps, often shortened to CPT maps (or elsewhere CPTP maps). There are many ways of expressing CPT maps, some of which are introduced in later chapters as required. One method is the Kraus representation, which is that every CPT map can be written in a form that ensures complete positivity

$$M(\rho) = \sum_{j=1}^{R} K_j \rho K_j^\dagger\,, \tag{1.20}$$

where the $K$ are called the Kraus operators, and the following condition ensures the trace is preserved

$$\sum_{j=1}^{R} K_j^\dagger K_j = \mathbb{I}\,. \tag{1.21}$$

The Kraus rank $R$ is the minimum number of them required to represent $M$, it is upper bounded by $d^2$ ($d$ being the dimension of the system). The set of Kraus operators is not unique, in fact there are infinitely many sets that are equivalent through unitary transformation. It also holds that any set of Kraus operators where $\sum K_j^\dagger K_j = \mathbb{I}$ produce a CPT map.

This description of open systems is used in modelling the effects of environmental noise through electromagnetic or otherwise radiation. Many treatments of open systems assume that the bath has no memory of its interactions with the system, and hence the effects of the bath on the system are not related to anything from a previous time. This is referred to as the assumption of Markovianity. It is a reasonable assumption when interaction is with some random entity of an effectively infinite bath where the coupling is very weak, and it is widely used, including in control theory. However, the system-bath model described above need not make this assumption. The interactions may remain random, but could be through stronger coupling that could imply some back-action, or rebound, out of the bath, implying the bath remembers past interactions. This regime is therefore called non-Markovian, and is the subject of Chapter 5. The subject of Markovianity has been, and continues to be, one of great debate, especially in a control setting. When the assumption can and cannot be made, how to define it, and how to test for its presence or otherwise remains a key area of research.

## 1.3 Quantum Control Theory

To be able to drive a quantum system in some desired way it must be possible have some control over the Hamiltonian. This could be some electric field generated by a laser or a radio frequency transmitter. Assuming that the system is subject to some constant Hamiltonian that is always on due to internal interactions and constant external influences, these can be considered as one *drift* Hamiltonian $H_0$. The other controllable influences will be considered here as fixed *control* Hamiltonians $H_j$ adding to the *drift* Hamiltonian independently and scaling linearly with some controllable amplitude. This will give rise to some combined Hamiltonian of the form

$$H(t) = H_0 + \sum_{j=1}^{N_c} u_j(t) H_j \, , \tag{1.22}$$

where $N_c$ is the number of controllable Hamiltonians, which are denoted $H_j$. The $u(j)$ are the amplitude functions, which can be considered as real valued without any loss of generality, so long as $H^\dagger = H$. Substituting Eq. (1.22) into Eq. (1.6) gives the *controlled* Schrödinger operator equation

$$\dot{U}(t) = -\mathrm{i}\left(H_0 + \sum_{j=1}^{N_c} u_j(t) H_j\right) U(t), \quad U(0) = \mathbb{I}\,. \tag{1.23}$$

The Schrödinger's equation, with its dynamics driven by Hamiltonians, resulting in unitary evolution is a specific case of a more general linear differential equation

$$\frac{\mathrm{d}\vec{x}}{\mathrm{d}t} = A\vec{x}\,, \tag{1.24}$$

with solution

$$\vec{x}(t) = X(t)\vec{x}(0), \quad X(0) = \mathbb{I}\,. \tag{1.25}$$

Here the nature of $X(t)$ depends on the constraints put on $A$. As in the Schrödinger's equation case, there is an operator equivalent,

$$\dot{X}(t) = A(t)X(t)\,. \tag{1.26}$$

If $A(t)$ is of the form

$$A(t) = A_0 + \sum_{j=1}^{N_c} u_j(t) A_j\,, \tag{1.27}$$

then Eq. (1.26) is a that of a general bilinear control system for operator control

$$\dot{X}(t) = \left(A_0 + \sum_{j=1}^{N_c} u_j(t) A_j\right) X(t), \quad X(0) = \mathbb{I}\,. \tag{1.28}$$

This is introduced here because much of this study focuses on systems where the dynamics is not driven by skew Hermitian generators, and hence the evolution is not described by unitary operators.

For a general time-dependent dynamics generator in Eq. (1.26) the natural logarithm of evolution operator $X(t = T)$ can be computed by integrating the generator, hence

$$X(T) = \hat{\mathcal{T}} \exp \left( \int_0^T dt A(t) \right) . \tag{1.29}$$

where $\hat{\mathcal{T}}$ denotes Dyson's time ordering operator. In the Schrödinger case this is

$$U(T) = \hat{\mathcal{T}} \exp \left( -\mathrm{i} \int_0^T dt H(t) \right) \tag{1.30}$$

This can be approximated by taking the product of operators evolving over a short period $\Delta t$ during which the generator is taken to be constant:

$$X(T) \approx X(T, T - \Delta t) \cdots X(t + \Delta t, t) \cdot X(t, t - \Delta t) \cdots X(\Delta t, 0) \, , \tag{1.31}$$

with

$$X(t + \Delta t, t) \approx \exp \left[ A(t) \Delta t \right] . \tag{1.32}$$

Note the order of the operations in Eq. (1.31) is important as the generators for different times will not necessarily commute. If the control functions $u_j$ are piecewise constant, that is their value is constant throughout the any duration of $t$ to $t + \Delta t$, then Eq. (1.31) becomes exact. Using $k$ as an index for the piecewise constant timeslot where there are $N_t$ timeslots, the time at the end of the timeslots is denoted $\{t_1, \ldots, t_k, \ldots, t_{N_t}\}$ and the duration of each timeslot as $\Delta t_k$ (for the purpose of this theoretical outline these can be considered equal, but they need not be). Using this notation Eq. (1.31) for piecewise constant control functions can be written

$$X(T) = X_{N_t} \cdot X_{N_t - 1} \cdots X_k \cdots X_2 \cdot X_1 \, , \tag{1.33}$$

with

$$X_k = \exp \left( A_{t_k} \Delta t_k \right), \quad k = 1, 2, \ldots, N_t \, , \tag{1.34}$$

being the evolution of the system during the $k^{th}$ timeslot. Here $A_{t_k} = A(t_k)$, with $A$ remaining constant up to and including $t = t_{k+1}$. As so far $X_0 = X(0) = \mathbb{I}$ it can therefore be ignored in this evolution.

## 1.4 Controllability

Clearly when designing an experiment, or some technological development, that requires control of quantum entities, it is important to know what operations are possible. Most specifically whether the desired operations are achievable. These could be specific transfers from one state to another or a range of unitary gate operations to be applied to some arbitrary state. It is clear that if all unitary operations are achievable, then all state-to-state transfers are also achievable. This situation is known as *operator controllable* or *fully controllable*. This study focuses on whether systems are fully-controllable or not, hence unless otherwise stated, in this thesis controllable means fully controllable.

The mathematical tools of differential geometry have long been used by physicists in understanding dynamical systems. The Lie algebra and group theory can be used in determining the controllability of systems evolving under the bilinear control of Eq. (1.28). For a mathematical description of how this can be applied to quantum systems, see Ref. [19]. A Lie algebra can be seen as a set of directions that can be taken in a space, and the corresponding Lie group as the set of operations that move from one point in the space to another. If all directions are available, then given sufficient time, all operations are possible, and hence all points are reachable. For operators to belong to a Lie algebra they must obey the properties of a *Lie bracket*, which for those studied herein are fulfilled by the usual matrix commutator $[A, B] = AB - BA$. A Lie algebra $\mathfrak{g}$ is said to *generate* elements of the Lie group $G$ through the relation

$$G = \mathrm{e}^{\mathfrak{g}} \,. \tag{1.35}$$

Due this relation, linear combinations of elements from $\mathfrak{g}$ result in consecutive operations in $G$. Hence the $A_0, A_j$ in Eq. (1.28) can be seen to be elements of a Lie algebra and the $X$ in Eq. (1.31) belonging to the corresponding Lie group. This allows Lie algebra and group theory to be applied to controllability, which is expanded upon below in Subsection 1.4.1. The relationship Eq. (1.35) is not in general surjective, however it is for the *unitary* $\mathrm{U}(n)$ and *special unitary* $\mathrm{SU}(n)$ Lie groups.

The set of operations that can be reached for a given control equation is called its *reachable set*. The reachable set of Eq. (1.28) is denoted $\mathcal{R}$. It is assumed that any piecewise constant control function, in terms of amplitude and the granularity of the time slicing, is possible. Not all operations are necessarily reachable for a given time $T$, those that are reachable are in the set $\mathcal{R}(T)$. So the definition of the reachable set can be clarified as

$$\mathcal{R} = \cup_{T\geq 0}\mathcal{R}(T)\,. \tag{1.36}$$

So if the closure of $\mathcal{R}$ is equal to Lie group corresponding to the Lie algebra of the generators, then, for compact groups, the system is fully controllable. For the skew Hermitian generators in Eq. (1.23) the group would be $\mathrm{U}(n)$, although $\mathrm{SU}(n)$ is often sufficient, as discussed below. The study of the characterisation of the reachable set for a non-compact group, the symplectic Lie group $\mathrm{Sp}(2n, \mathbb{R})$, is described in Chapter 3.

### 1.4.1 Lie algebra rank criterion

Working with the assumption that the total time $T$ is long enough, and that the piecewise control functions are sufficiently sliced, the *Lie algebra rank criterion* can determine whether the system is fully controllable. This has been applied to general bilinear control systems long before the advent of quantum control theory [10], as shown for compact groups in Ref. [23]. It is applicable to quantum control of closed systems because all matrices $-\mathrm{i}H$ are skew-Hermitian, hence belong to the skew-Hermitian Lie algebra $\mathfrak{u}(n)$, which generate the (compact) *special unitary* group $\mathrm{SU}(n)$. A full description of the Lie algebra rank criterion and its application to controllability, including a proof, is given in [19]. An outline of the principles behind it and a description of how it is applied is given below.

Substituting in the exponentials in to Eq. (1.33) gives

$$X(T) = \prod_{k=N_t}^{k=1} \exp\left(\left[A_0 + \sum_{j=1}^{N_c} u(t_k)A_j\right]\Delta t_k\right)\,, \tag{1.37}$$

where the product here is time-ordered, as in Eq. (1.33). If the $u(t_k)A_j$ commute for all times $t_k$

then this would be equivalent

$$\prod_{k=N_t}^{k=1} \exp\left(A_0 \Delta t_k\right) \prod_{j=1}^{N_c} \exp\left(u(t_k) A_j \Delta t_k\right).$$

However, for non-commuting matrices $A, B$, ***i.e.*** $[A, B] \neq 0$,

$$\mathrm{e}^{A+B} \neq \mathrm{e}^{A}\mathrm{e}^{B}\,, \tag{1.38}$$

but for some short time $\Delta t$, the *Baker-Campbell-Hausdorff* formula (truncated here as the further nested commutator terms will have factors of increasingly higher powered $\Delta t$)

$$\mathrm{e}^{A\Delta t}\mathrm{e}^{B\Delta t} \approx \mathrm{e}^{A\Delta t + B\Delta t + \frac{\Delta t^2}{2}[A,B]}\,. \tag{1.39}$$

gives some insight into how the commutators of the drift and control Hamiltonians can influence the dynamics of the system, by effectively adding directions that are not explicit.

The Lie algebra rank condition is fulfilled if the number of linearly independent generators that can be made through nested commutation of the drift and control dynamics is equal to the dimension of the Lie algebra to which they belong. When this is fulfilled the generators are said to *span* the space. Each can be thought of as generating a movement in a unique direction in the operator space of the system. In the case of the unitary dynamics of closed quantum systems the dynamics generators are the skew Hermitian matrices $\mathrm{i}H_0, \mathrm{i}H_1, \ldots, \mathrm{i}H_j, \ldots, \mathrm{i}H_{N_c}$ from the drift and control Hamiltonians belonging to the Lie algebra $\mathfrak{u}(n)$, corresponding to the unitary Lie group $\mathrm{U}(n)$ where $n$ is the dimension of the quantum system, with $n^2$ as the dimension of the Lie algebra. For spin systems or their equivalents the generators are traceless, and hence their Lie algebra $\mathfrak{su}(n)$ and corresponding special unitary Lie group $SU(n)$, have dimension $n^2 - 1$.
The implicit generators come from nested commutators of the form

$$B = [A_1, [A_2, [\ldots [A_{l-1}, A_l]]]]. \tag{1.40}$$

Here the $A$ matrices are the original dynamics generators and others formed from these nested commutators. The matrix $B$ is a generator that may or may not be linearly independent. The description of an algorithm that can be used to determine the Lie rank is given in Section 2.5. As a simple example a control system of a two-level or spin system can be considered

$$H = \sigma_z + u(t)\sigma_x \tag{1.41}$$

The Pauli matrices $\sigma_x, \sigma_y, \sigma_z$ are known to provide a basis for generating all operations in a single spin system, as they span the $\mathfrak{u}(2)$ Lie algebra. In this example there are two explicit generators $-iH_0 = -\mathrm{i}\sigma_z$ and $-\mathrm{i}H_1 = -\mathrm{i}\sigma_x$. The commutator of the two yields $[-\mathrm{i}\sigma_z, -\mathrm{i}\sigma_x] = -2\mathrm{i}\sigma_y$, hence the full basis is achieved. The number of linearly independent generators is 3, $2n^2 - 1 = 3$, hence the Lie algebra rank condition is met.

## 1.5 Quantum technology

Initially the study around the strange behaviour of particles on the microscopic scale was motivated by the desire to understand some unexplained phenomena such as ultraviolet catastrophe of black body radiation and the photoelectric effect. Starting in the early twentieth century this developed into what now know as *quantum mechanics*. This understanding led to the development of what is now commonplace technology such as transistors, light emitting diodes and lasers. Richard Feynman is credited [24] with first suggesting that the quantum mechanical effects, such as superposition, could be used in some kind of computer to solve physics problems that are intractable on so-called classical computers due to the number of calculations needed [25]. Since then ideas for new technology exploiting the features of quantum mechanics have burgeoned. Many of these have reach the point of, or are near to, commercial exploitation. This is illustrated and assisted by the UK Government's £270M investment in Quantum Technology Hubs in 2014.

It is essential to better understand how quantum systems can be controlled in order to exploit these quantum phenomena for commercial and wider scientific use. In most cases this will involve a complex system of many entities. Therefore it is important to know what tools are applicable to aid their development, hence this thesis to addresses the question of scalability in quantum control optimisation. Some specific areas of technological development are introduced below. Although quantum control is relevant to all, quantum simulators and quantum computing are the areas where scalability is the greatest challenge.

### 1.5.1 Quantum communication

Codification and encryption of messages has been of huge importance in all of human history [26]. Originating with the necessity for communicating military messages without interception, this developed into the field of *cryptography*, and continues to be paramount to national security. The greatest minds of the time were employed in the making and breaking of codes during the Second World War. Today the encryption of digital information underpins national and international commerce. With the potential of a quantum computer to crack traditional codes, and evermore sophisticated and determined cyber-snooping and cyber-attacking, national security agencies and businesses are intensely interested in any technology that will allow secure exchange of information, and those that can intercept it!

*Quantum cryptography* broadly includes all applications of quantum effects to enable the secure transmission of information. The most well known and well developed is *quantum key distribution*, in which exchange of quantum entities is used to share an encryption key, which can then be used to encrypt data and send via a classical communication channel. There are well known protocols BB84 protocol [27] and E91 [28]. Devices that use BB84 or E91 protocols over fibre optics cables are commercially available. Their effectiveness is limited by the distance over which the photons can be sent and received with sufficient reliability – all fibre-optic cables have some absorption probability. Quantum key distribution has also been demonstrated over over free space [29] and even intercontinentally via satellite [30]. The development of a global network for quantum key distribution is clearly of great value. The quantum networks that would support this offer many other potential applications, such as distributed quantum computing and metrology [31], and so quantum networks remain a key area of research interest and investment [32]. Quantum control optimisation has been used to implement teleportation protocols in quantum networks [33].

### 1.5.2 Quantum metrology

Use of quantum entanglement, superposition and squeezed states to achieve precision beyond that of purely classical devices is known as *quantum metrology*. The most discussed application of this is in the measurement of gravitational waves, where it is anticipated that quantum effects will be used to enhance the already highly sensitive instrument at the Laser Interferometer Gravitational-wave Observatory (LIGO) [34]. There are many other applications for high-sensitivity gravity sensors, such exploring what is below ground for infrastructure planning, predicting subsidence and water resource management.

The same principles can be applied to sensors of many types, including magnetic field and rotation sensors. Rotation sensors have a wide variety of industrial applications, and particularly in navigation and stabilisation system. High precision quantum clocks also come under the umbrella of quantum metrology. High sensitivity imaging is also a key objective for this field. It is less mature than quantum communication, however practical prototypes are being developed, such as this brain scanning technology [35], and other biological applications [36].

Control has been used to optimise states used to increase phase sensitivity [37]. Squeezed states are typically a feature of these systems. Controlling squeezing in unstable systems is investigated in Chapter 3.

### 1.5.3 Quantum simulators

The idea behind quantum computing originated from considering how computers can be used to simulate physical systems [25]. If all the interactions of the nuclei and electrons are to be considered at the quantum level, then even modelling a small molecule requires a huge number of calculations [38], and the problem scales exponentially with number of particles considered. By taking some controllable quantum system and engineering the Hamiltonian it is possible to simulate other quantum systems that may be of particular interest [39].

These controllable systems are called *quantum simulators*, and their proposed applications are numerous and varied. At this time they are mostly to aid research in physics and other disciplines, including condensed matter physics, high-energy physics and quantum chemistry [40, 39, 41].

The much mentioned specific examples being understanding the mechanism of high temperature superconductivity and quantum magnetism. It is speculated that if these physics questions are addressed successfully using quantum simulators, and they develop into ever more sophisticated devices, then the range of applications will extend to biological reactions such photosynthesis, nitrogen fixation and drug target interfaces.

The quantum simulators discussed here of an analogue rather than the digital type, which are quantum computers able to simulate some other quantum system by solving a general time-dependent Schrödinger equation through gate operations [42, 43]. The digital quantum simulator could therefore simulate any quantum dynamics, whereas the analogue version is limited to a class of problems which the inherent Hamiltonian can be manipulated to emulate the system of study. Large scale analogue simulators are a reality, with ever larger devices being developed. This system simulates Ising interactions with a lattice of beryllium ions proposed for investigating quantum magnetism [44, 45]. In this experiment two dimensional arrays of Rydberg atoms are established, also for investigating quantum magnetism [46]. It it is a popular view that quantum supremacy will be demonstrated on these devices before a quantum algorithm out performs a classical counterpart [40], as discussed in the next section.

### 1.5.4 Quantum information and computing

Electronic digital computing is based voltage thresholds in wires, thus a particular wire or connection holds a Boolean value or either “0” or “1”. Transistors are used to effect Boolean logic operations such as NOT, OR, AND. From these building blocks computing processors are constructed that effect arithmetic on numbers entered or stored in the binary base. Quantum computing is analogous to its digital counterpart in that a series of gates work on parallel inputs and outputs where the values are Boolean. The two separate due to the potential, in fact requirement for any quantum speedup, for states in process to be superpositions of “0” and “1” and maybe entangled. The gates also must effect reversible processes, that is it must be possible to map the outputs back to unique set of inputs. This implies that the number of inputs and outputs must be equal, and hence the commonplace two-input-one-output digital gates cannot be replicated in a quantum computer.

To achieve these quantum features the values have to exist as the quantum state of a qubit, such as the energetic states of a two-level system or the directions of a quantum spin-½ system. This provides the potential for superposition states. The gates are effected through some unitary transformation on one or more qubits, hence why the they must be reversible processes. The qubits are organised into *quantum registers*, hence they can represent numbers in binary format. The series of gates that act on the register is called a *quantum circuit*. A device that that can effect a quantum circuit is known as a *quantum processor*. These are the components of a *digital quantum computer*.

A digital quantum computer could execute a *quantum algorithm* that comprises quantum circuits. A quantum algorithm implies one that can theoretically solve some problem faster than any classical counterpart. These kind of problems are characterised by having many possible (correct or incorrect) solutions that could be considered, but only one solution is required. By creating superpositions between the states in the register the potential solution set can be considered as a whole, with quantum interference effects constructively and destructively filtering for a solution. David Deutsch describes this in the 'many universe' interpretation as each universe considering one of the potential solutions, the incorrect (or sub-optimal) ones are ruled out, leaving all universes with a correct (or optimal) solution [24]. If there is only one solution, then all universes agree, otherwise the measurement outcome is probabilistic. Problems that can be solved exponentially faster on a quantum computer form the BQP complexity class. This is a small subset of problems that can be solved by a computer, and so quantum computers will never replace classical computers. there is just this small subset of problems on which a quantum computer could out-perform a classical computer.

The most famous quantum algorithm is Peter Shor's algorithm for prime factoring of large numbers [47, 48]. It is famous because it is practical and dangerous. Practical for the purpose of demonstrating quantum computing, because the qubit register simply needs to be initialised with the product of two prime numbers. It is dangerous because the public-private key encryption that underpins much of current internet security relies on the assumption that factorising the product of two large prime numbers is computationally very challenging. Public keys that could not be cracked within any reasonable time frame using any classical algorithm could in theory be cracked almost instantaneously by a quantum computer running Shor's algorithm. However, to crack currently used keys thousands of coherent qubits would be needed, which is very far from the current technology. Also, there is a whole research devoted to preparing for this potential danger – *post quantum cryptography*.

It can be proven that only a small set of quantum gates involving a maximum of 2 qubits each is required to perform any quantum algorithm [49]. One such set of *universal quantum gates* is the single qubit gates Hadamard $H$, phase gate $T = R_z(\pi/4)$ and the two qubit controlled-NOT. In matrix form these are

$$H = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}, \quad T = \begin{pmatrix} 1 & 0 \\ 0 & \mathrm{e}^{\mathrm{i}\pi/4} \end{pmatrix}, \quad \mathrm{CNOT} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}. \tag{1.42}$$

So any computer able to effect these gates can be considered a *universal quantum computer*.

This description of quantum computing based on qubit registers is the most common. The principles are applicable to other types of quantum system, some of which are discussed in the next chapter. Entirely different models, not based on gates, have also been proposed. As quantum computers are only ever expected to perform a specific small subset of computing problems with any advantage over digital computers, then it clearly makes sense to aim for binary inputs and outputs to allow simple logical interfaces between them.

In practice coupling to the environment leads to loss of coherence in the qubit states. Preparation of the initial state, gate application, and read out of the final state all also prone to error with some probability. For all of these sources of error there are protection protocols. However, they all require some additional qubits, known as ancillas, with a greater number required for higher error probabilities. Decoherence is a function of time, and the time required for the algorithm is a function of the gate time. Hence maximising the gate fidelity and minimising the gate time are imperative objectives for furthering this technology. These are key objectives of quantum control, and a central focus of this study.

Computers that run Shor's algorithm have been built. The first claim for this was from IBM in 2001, using the 7 spins of a molecule in an NMR controlled experiment to factorise 15 into 3 and

5 [50]. Since May 2016 IBM have provided public access to a 5 qubit quantum processor based on superconducting transmon qubits. This they call the 'IBM Q Experience', and as of May 2018 it has two 5 qubit and one 16 qubit processors available to the public. As well as an online circuit configuration interface, quantum algorithms can be constructed using a Python library called Qiskit and submitted to the job queue via the portal. Google partnering with the University of California Santa Barbara have also chosen to lead their quantum computer development using a similar technology. A 16 qubit processor will not demonstratively out-perform current classical computing hardware in solving any problem. Scaling these systems to a number of coherent qubits that will demonstrate this remains the current challenge, and is called demonstrating *quantum supremacy* [51]. Google claim that a 50 qubit processor will be sufficient to show quantum supremacy with their technology [52], although it is anticipated that the first demonstration of supremacy is likely be through boson sampling to calculate the permanent of a large matrix [53, 54].

Such is the intensity of drive to succeed in this elusive field, and the time elapsed since its conception, that a great many solutions have been proposed and developed. Ionised atoms trapped by electromagnetic fields have been held as a forerunner [55], with qubits stored in electronic states and interactions for multi-qubit gates through coupled motional states. Qubits represented by the polarised states of photons are also believed to be strong candidate for success. Gates are effected through standard linear optical equipment of beam splitters and phases shifters. They would also work at room temperature, unlike transmon and ion trap computers. They are particularly suited to distributed computing, quantum communication and networks, as the qubits are inherently mobile. Qubits held in single electron transistors or nuclear spins in doped silicon offer the best interconnectivity with current classical digital computers [56, 57].

Superconducting circuit qubits provide a platform where the single and two-qubit gates can be effected with high fidelity and the preparation and readout steps can be performed reliably [58]. In 2014 the Google and Santa Barbara partnership reported fidelities with single-qubit gates of up to 99.92% and two-qubit gate fidelity of up to 99.4% [59]. Their current challenge is scaling this to sufficiently large qubit registers. Scalability problems are also faced by those groups trying to build a scalable quantum computer using quantum optical devices [60]. High fidelity single qubit and multi-qubit gates can be effected [61]. Trapping many ions is relatively easy, but the individual addressing to make them a registers of qubits is not. In all of these systems *scalability* is now the main challenge.

It is impossible to predict which of these technologies, or others not mentioned, will succeed and maybe come to dominate, or whether some combination provides they key. Some doubt that quantum supremacy will ever be demonstrated [62], and that maybe the idea violates some principle yet to be discovered.

### 1.5.5 Quantum machine learning

With recent advances in machine learning technology, in particular the graphical processing units that enable fast training of large data sets, applications have been finding mainstream attention. One of the most striking was when Google's AlphaGo beat the world champion at Go [63]. The state-of-the-art for solving some higher dimensional space optimisation problems are these kind of deep learning tools. Machine learning approaches have been shown to out-perform established methods in many areas of physics research, including devising new experiments generating quantum entanglement [64], quantum error correction [65], and in quantum control [66]. Reinforcement learning is based on trailing potential solutions and learning through rewards. In deep reinforced learning neural networks provide the learning platform to support the learning process. This deep reinforced learning method is used by AlphaGo, and has now been used in quantum optimal control [67, 68]. Machine learning has also be shown to work *in situ* to optimise Bose-Einstein condensate production [69], where they refer to it as 'online optimization'.

Consequently scientists and technologists look to improve upon the technology. The main time limiting factor in systems based on neural networks is the computational resource required to optimise the weights based on the training data. Therefore many scientists are looking at this process to assess the potential for quantum speedups.

The training of these neural networks actually uses the same linear algebra and gradient-based, multi-variable optimisation methods that are discussed in Subsection 1.6.2. As discussed in the review article [70], there is much on-going work to make fully quantum machine learning tools. Underlying components such as linear algebra and gradient based optimisation have proposed quantum algorithms [71, 72], although all have limitations that make them currently impractical due to the number of qubits required. All of these algorithmic tools require a quantum computer to run on, and so until quantum supremacy has been demonstrated, presumably with simpler algorithms, quantum speedups for machine learning cannot be verified. However, the hints of recursion here at least allow the imagining of a fully quantum computing device that optimises itself for gate time and fidelity. Some steps towards this are the subject of Chapter 4.

## 1.6 Quantum control function optimisation

Determining the shape of the functions $u_j(t)$ in Eq. (1.28) to achieve some objective is the role of *control function optimisation*. When equation Eq. (1.28) describes the dynamics of a quantum system, such as in Eq. (1.23) then this is called *quantum control function optimisation*, or simply *quantum control optimisation*. This objective could be transfer from one state to another, known as *state-to-state* control, or some specific operator, known as *gate synthesis*. For the unitary dynamics described above, in the state-to-state case the initial condition is set as state $|\psi_0\rangle$ and the target would be some state $|\psi_{\text{target}}\rangle$. In the gate synthesis case the initial condition is $\mathbb{I}$ and the target $U_{\text{target}}$. Although there are analytical methods optimising the control functions, for instance those used in Refs. [73, 74], these are applicable only in specific cases, and this study investigates control of many-body systems without specific symmetries, such as those used in quantum information processing. Typically the process starts from some guess for the control functions, and then a numerical optimisation method is used to iteratively improve upon the guess until the objective is achieved. This is often referred to as *control pulse optimisation* or just *pulse optimisation*.

### 1.6.1 Cost functions

Typically an optimisation method will look to manipulate a set of variables with the aim of minimising some *cost function*. The primary objective here is to achieve the desired evolution target, therefore some method of measuring the equivalence of the operator or state evolved under the controls and the target is required. This is often referred to as a *figure of merit* or *fidelity* when there is a theoretical ceiling, and hence it is a valid aim to maximise the fidelity. As it is more typical in numerical algorithms to minimise a function, rather than maximise, the aim is to minimise the error in fidelity, known as the *fidelity error* or *infidelity*. Therefore the first component of the cost function is the infidelity. The fidelity will be denoted $\mathcal{F}$ and the fidelity error $\mathcal{E}$, both are real numbers. For the purpose of optimising controls to implement gates or state transfers all that is really needed is some measure that increases or decreases monotonically as the target is approached, and that is equal to a known value when the target is reached. Ideally, for comparability, these would be bounded as $0 < \mathcal{F} < 1, 1 > \mathcal{E} > 0$ [75]. Whereas for all measures presented below it possible to find a lower bound for $\mathcal{E} > 0$, it is not always possible to find an upper bound, as will be seen for the non-compact group investigated in Chapter 3.

### Unitary fidelity

When the dynamics are that of a normalised state vector $|\psi\rangle$ in Eq. (1.4), or equivalently a unitary operator $U$ in Eq. (1.6), then, assuming that the target has been achieved exactly, the normalised overlap of the final evolution and target will yield unity. That is

$$g = \langle \psi_{\text{target}} | \psi(T) \rangle \,, \quad g = 1 \implies |\psi(T)\rangle = |\psi_{\text{target}}\rangle \,, \tag{1.43}$$

or

$$g = \lambda \mathrm{Tr}\left\{U^\dagger_{\text{target}} U(T)\right\} , \quad g = 1 \implies U(T) = U_{\text{target}} \,, \tag{1.44}$$

where $g \in \mathbb{C}, 0 < |g| < 1$. For a system of dimension $d$, the scaling factor $\lambda = 1/d$. If $|g| = 1$ then the evolution and target differ only by a global phase. Therefore the most typical fidelity measure for unitary systems, where global phase differences are typically ignored, is

$$\mathcal{F}_{\text{PSU}} := |g| \,. \tag{1.45}$$

The PSU subscript here indicates the projective special unitary group, in which matrices differing only by a phase are equivalent. If global phase differences are to be respected then the measure is

$$\mathcal{F}_{\text{SU}} := \mathrm{Re}\, g \,. \tag{1.46}$$

In either case the infidelity is given by

$$\mathcal{E} = 1 - \mathcal{F} \,. \tag{1.47}$$

### General matrix infidelity

For a general matrix evolution it is possible to define a measure of the infidelity, which will be referred to as the *trace difference* infidelity, as

$$\mathcal{E}_{\mathrm{TD}} := \lambda \left\langle X(T) - X_{\mathrm{target}} \right\rangle_{\mathsf{F}} = \lambda \operatorname{Tr}[(X(T) - X_{\mathrm{target}})^{\dagger}(X(T) - X_{\mathrm{target}})] \,, \tag{1.48}$$

where $\lambda$ is some scale factor, and $\langle\cdot\rangle_{\mathsf{F}}$ denotes the Frobenius inner product, which is the square of the Frobenius norm $\|\cdot\|_{\mathsf{F}}$. This can therefore be used when the evolution operators belong to any subgroup of the general linear Lie group $\mathrm{GL}(n)$. It is chosen because it is relatively efficient to calculate gradients, which is discussed later in this section. The application of this infidelity in various regimes of quantum dynamics is discussed in Ref. [76]. The trace difference infidelity can related to the unitary gate fidelity by first choosing the appropriate scale factor, which is $\frac{1}{2n}$ for an $n \times n$ matrix, and then

$$\mathcal{F}_{\mathrm{SU}} = \sqrt{1 - \mathcal{E}_{\mathrm{FD}}} \,, \tag{1.49}$$

for $X_{\mathrm{target}}, X(T) \in \mathrm{U}(d)$. There is no equivalence for ignoring global phase differences with this measure, and hence it cannot be related to $\mathcal{F}_{\mathrm{PSU}}$ in any similar way.

### Density matrix infidelity

The general matrix infidelity can also be used in optimising state-to-state transfer for ensembles or reduced states, that is when the states are represented by density matrices. The equivalent fidelity measure to the $\mathcal{F}_{\mathrm{SU}}$ of pure states is

$$\mathcal{F}_{\mathrm{DM}} := \operatorname{Tr}[\sqrt{\rho_{\mathrm{target}}} \cdot \rho(T) \sqrt{\rho_{\mathrm{target}}}] \,. \tag{1.50}$$

Note that in some literature, including [20] the fidelity is defined as $\sqrt{\mathcal{F}_{\mathrm{DM}}}$. However, there are strong arguments for using the definition of Eq. (1.50), as in the context of quantum processes this can be interpreted as the probability of success for a quantum computation [75]. A typical measure for the infidelity of two density matrices is the *trace distance*

$$D(\rho_1, \rho_2) := \tfrac{1}{2} \operatorname{Tr} \left| \rho_1 - \rho_2 \right| , \tag{1.51}$$

where $|X| = \sqrt{X^\dagger X}$. With the definition of $\mathcal{F}_{\mathrm{DM}}$ in Eq. (1.50) $D(\rho_1, \rho_2)^2 = 1 - \mathcal{F}_{\mathrm{DM}}(\rho_1, \rho_2)$ [20], and so $\mathcal{E}_{\mathrm{TD}}(\rho_{\mathrm{target}}, \rho(T)) = \left(D(\rho_{\mathrm{target}}, \rho(T))\right)^2$. Hence a reasonable infidelity for density matrices is

$$\mathcal{E}_\rho := \tfrac{1}{2} \left\langle \rho(T) - \rho_{\mathrm{target}} \right\rangle_{\mathsf{F}} \ , \tag{1.52}$$

which is the same as the trace difference Eq. (1.48), with $X = \rho, \lambda = 0.5$.

Both the trace distance and the fidelity are widely used in quantum information science according to Ref. [75], in which a wide variety of distance measures are discussed. As what is of primary interest is $\mathcal{F} \lessapprox 1$ or $\mathcal{E} \gtrapprox 0$, then either is valid for optimisation purposes. The square root makes analytic gradients difficult, so the trace difference is preferred over the trace distance when using optimisation methods that use gradients.

### Penalty functions

It may be desirable to combine the optimisation with respect to the infidelity with some other objectives that are important for some physical implementation of the control function. These would often include favouring smoothness in the functions, or minimising the total energy required, and very often minimising the total evolution time. These can objectives can be included in the optimisation procedure by adding some *penalty functions* to the cost function. These would be some function of the control function parameters, which in the given examples would scale with mean absolute gradient, integral of the absolute amplitude, total evolution time.

Although the penalty functions can be fairly straight forward to include in the cost function, they are not always so trivial to include in the gradient calculations. This is a problem in gradient based algorithms, which will be discussed later in this section. Recently the use of *automatic* *differentiation* techniques have been able to overcome this problem [77].

### 1.6.2 Multi-variable optimisation methods

Having defined some cost function to minimise, the next step is find some scheme of parametrisation for the control functions. That is, to have some discrete set of variables for an optimisation method. The most obvious parametrisation is the piecewise constant timeslots, as this also allows for exact calculation of the evolution using Eq. (1.33). In some experimental settings smooth control functions are advantageous, in which case other parametrisation schemes are needed. These are discussed below in Subsection 1.6.6. In this study the piecewise constant parametrisation is used exclusively, as analytic gradients can be computed, allowing the most efficient use of gradient based optimisation methods, and hence it is the focus here.

The variables for the optimisation method are then the amplitudes for a specific control in a given timeslot. They will be referred as *control timeslot amplitudes*. Following the notation of Section 1.3, $u_{j,k}$ is the amplitude of the $j^{th}$ control in the $k^{th}$ timeslot, hence

$$u_j(t) = u_{j,k}, \quad t_{k-1} < t \leq t_k, \quad k = 1, 2, \dots, N_t \,. \tag{1.53}$$

Hence this leads to $N_p = N_c N_t$ variables for the optimisation method ($N_c$ being the number of controls). For numerical methods the efficiency, and often efficacy, is improved by using the lowest possible number of variables. However, a sufficient number are required, otherwise the problem is over-constrained. Theoretically the minimum should be $d^2$ for gate synthesis and $2d - 1$ for state-to-state control, as this is the number of parameters in $U$ and $|\psi\rangle$ respectively for a $d$ dimensional system. However, in practice, this is difficult to predict in many-body systems, and trial-and-error approaches are valid for finding lower numbers of parameters that work. To a multi-variable optimisation method they would be indistinguishable, and hence they could be represented by a vector on real-valued vector space $\vec{\alpha} = (\alpha_1, \dots, \alpha_{N_p})$.

### Gradient based methods

Many methods use the gradient of the cost function with respect to the optimisation variables to direct the search for optimal parameters. In a first order gradient method each iteration of the method is to take a single step in the direction of maximum gradient. This can be visualised as attempting to reach the top of a hill by taking each step in the steepest direction, which may be the only option travelling by foot on a foggy day with no map. Consequently, gradient based methods are often referred to as ‘hill-climbing’ methods. Optimisation problems can generally be turned on their head with an appropriate minus sign, and it is more typical to frame a problem as a minimisation. In which case the analogy would be ‘hill-descending’. However, in a theoretical multi-dimensional parameter space there is no limit to the size of step that may be taken. Therefore it is possible to jump directly to the solution if it can be determined in some way. In practice first-order gradient methods are slow to converge, and it is difficult to chose an appropriate step size [78].

### Newton-Raphson method

The Newton-Raphson method for finding the roots of a function can also be applied to multi-variable optimisation. In one dimension the root of function $f(x)$ can be approximated by successive iterations of

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}, \tag{1.54}$$

where $f'$ is the derivative of the function. Geometrically this can be seen as $x_{n+1}$ being the $X$-intercept of the tangent to the function. The method is known to converge on the root quadratically so long as the derivative is well behaved near the root. It is clear that the method can be adapted to find stationary points by looking for the roots of the derivative, that is

$$x_{n+1} = x_n - \frac{f'(x_n)}{f''(x_n)}, \tag{1.55}$$

In multiple dimensions the derivative of the function on the parameter space $\nabla f(\vec{\alpha})$ is known as the gradient. In the context of the control pulse optimisation this is the derivative with respect to each of the pulse parameters, and hence will be a vector of length $N_p$. The matrix of second derivatives is called the Hessian, and will be denoted $B$. Note that Hessian is symmetric as $\frac{\partial^2 f}{\partial \alpha_l \alpha_m} = \frac{\partial^2 f}{\partial \alpha_m \alpha_l}$ for a well behaved function $f$. The Hessian for the pulse optimisation will be a $N_p \times N_p$ matrix. The iterative step for a multi-variable function optimisation is then

$$\vec{\alpha}_{n+1} = \vec{\alpha}_n - B^{-1} \nabla f(\vec{\alpha}) \,, \tag{1.56}$$

where $B^{-1}$ is the inverse of the Hessian. In the large dimensional system the inverse of the Hessian could be numerically expensive to calculate, and so the step could be solved as if for a system of linear equations

$$B \Delta \vec{\alpha}_= \nabla f(\vec{\alpha}) \,, \tag{1.57}$$

where $\Delta \vec{\alpha} = \vec{\alpha}_{n+1} - \vec{\alpha}_n$.

Due to the perceived difficulty of calculating second order derivatives of fidelity functions, or the anticipation of excessive computational expense, the use full Newton-Raphson optimisation method has been avoided in quantum optimal control, although some success with this method is reported in Ref. [79]. Instead the so-called *quasi-Newton* methods are preferred.

### Quasi-Newton methods

It may not be trivial to gain the analytic expressions the second order derivatives with respect to all the pulse parameters, or they may just be too numerically expensive to compute. For these situations a family of methods have been developed that make an approximation of the Hessian, or its inverse, over successive iterations. This is known as the quasi-Newton family of methods. One of the most widely used is the *Broyden, Fletcher, Goldfarb, and Shanno* (BFGS) method [80]. In variants where the full Hessian matrix (or its inverse) is not stored, rather a set of gradients from successive iterations are stored and used to compute the iteration step, the method is called *limited memory* or L-BFGS. In general BFGS is an unconstrained optimisation method, that is no bounds are placed on the optimisation variables. Variants where bounds can be placed on the variables are called BFGS-B, and the limited memory, bounded variants called L-BFGS-B [81].

### 1.6.3 Gradient calculations

Not all optimisation methods require gradients, however those that do are generally found to converge faster in quantum optimal control [78, 82]. It is possible to calculate approximate gradients using finite difference method, that is

$$\frac{\partial \mathcal{E}}{\partial \alpha_k} \approx \left[\mathcal{E}((\alpha_1, \ldots, \alpha_k + \epsilon, \ldots, \alpha_{N_p})) - \mathcal{E}(\vec{\alpha})\right] / \epsilon \,, \tag{1.58}$$

where $\epsilon$ is a 'trial' step-size in each direction to estimate the gradient. However, each call to an infidelity function can be costly to compute, as each would require computation of the system evolution as Eq. (1.33). This is particularly so for a large Hilbert space, as numerical computation of a matrix exponential is an expensive operation for large matrices. This would typically be compounded by there also being a large number of optimisation parameters, which in a piecewise constant parametrisation would correspond to many propagators that require exponentiation to calculate. Consequently, finding a method for calculating gradients that does not require repeated calculation of the full evolution would be highly advantageous. It is also difficult to chose an appropriate step-size $\epsilon$, too small will fail to find a gradient, and too large will miss features of the parameter landscape.

The method for calculating exact gradients used in this study was first introduced as the GRAPE algorithm. The GRadient Ascent Pulse Engineering algorithm was first proposed as a method for optimising the control pulse in NMR coupled spin dynamics [83]. GRAPE was compared with the Krotov type algorithms in Ref. [78], which introduced a library developed in MATLAB called 'DYNAMO'. This paper is the main source of reference for this section, as the version of the algorithm used in this study was based upon it. For more detail on the algorithm used in this study see section Section 2.3. The GRAPE paper proposed this method of calculating the gradients using what they called forward and back propagation. In the original GRAPE paper these gradients were used in a first order hill climbing method to maximise the fidelity. The DYNAMO paper and corresponding library improved the performance of the algorithm by introducing new methods for calculating the gradient, using the BFGS optimisation method, and other efficiencies gained through reducing the number of matrix multiplications. The GRAPE algorithm was further enhanced by proposals in

Ref. [76] by clarifying how to efficiently calculate the propagator gradient for the general matrix evolution and apply this to the trace difference fidelity, hence making GRAPE generally accessible for open system and symplectic dynamics, as will be seen later.

### Propagator gradient

To calculate the gradients of the fidelity with respect to the optimisation parameters, first the *propagator gradient* corresponding to each of the parameters must be calculated. In the piecewise constant setting this is a propagator gradient for each control timeslot. The method for unitary dynamics, based on the spectral theorem, given in [78] is

$$\langle \lambda_l | \frac{\partial X}{\partial u_{j,k}} | \lambda_m \rangle = \begin{cases} -i\Delta t \, \langle \lambda_l | H_j | \lambda_m \rangle \, e^{-i\Delta t \lambda_l} & \text{if } \lambda_l = \lambda_m \\ -i\Delta t \, \langle \lambda_l | H_j | \lambda_m \rangle \, \frac{e^{-i\Delta t \lambda_l} - e^{-i\Delta t \lambda_m}}{-i\Delta t \, (\lambda_l - \lambda_m)} & \text{if } \lambda_l \neq \lambda_m \; , \end{cases} \tag{1.59}$$

here the $|\lambda_i\rangle$ are the orthonormal eigenbasis of $H_{t_k}$, the combined Hamiltonian in the $k^{th}$ timeslot, and $\langle \lambda_l | \frac{\partial X}{\partial u_{j,k}} | \lambda_m \rangle$ is the propagator gradient is this basis. One of the other advantages of this method is that it puts the timeslot Hamiltonian in its eigenbasis so that propagator $X_k$ can be calculated by exponentiating the diagonal elements, which is significantly less expensive than a full matrix exponentiation. The derivation of this method is given in Appendix A of [78].

The propagator gradient for a general linear matrix can be calculated using the *augmented matrix* method, as described in [76], that is

$$\begin{pmatrix} X_k & \frac{\partial X}{\partial u_{j,k}} \\ 0 & X_k \end{pmatrix} = \exp \begin{pmatrix} A_{t_k} \Delta t & A_j \Delta t \\ 0 & A_{t_k} \Delta t \end{pmatrix} \tag{1.60}$$

where $A_{t_k}$ is the combined dynamics generator at time $t_k$. Here it can be seen that the propagator is again computed as a bi-product. This is also known as the *Fréchet derivative* [84].

### Infidelity gradients

To see how these propagator gradients can be used to calculate the exact gradient of the infidelity it is convenient to introduce a notation borrowed from Ref. [78] for the time evolution of the system up to $t_k$

$$X(t_k) = X_{k:0} := X_k \cdot X_{k-1} \cdots X_1 \cdot X_0 \tag{1.61}$$

where $X_0$ is the initial state or operator, that is $X_0 = |\psi_0\rangle$ for state-to-state ($X_0 = \vec{x}$ in general vector evolution) or $X_0 = \mathbb{I}$ for operator evolution. This is referred to as the *forward propagation*. Also it is necessary to define what is called the *co-state* in Ref. [78] which comes from what is described as *backward propagation* in Refs. [83, 78], but here is called *onto propagation* to the target $X_{N_t+1} \equiv X_{\text{target}}$,

$$\begin{aligned} \Lambda^\dagger(t_k) = \Lambda^\dagger_{:k+1} :=& X^\dagger_{\text{target}} X_{N_t} X_{N_t-1} \cdots X_{k+1} \\ =& X^\dagger_{N_t+1} X_{N_t} X_{N_t-1} \cdots X_{k+1} \,. \end{aligned} \tag{1.62}$$

This is the evolution *backwards* (in the Dyson time-ordering sense) from the target operator, or the evolution *onto* the target, from the end of timeslot $k$.

In terms of these the overlap from Eq. (1.43) is

$$g = \lambda \mathrm{Tr}\left\{\Lambda^\dagger(t_k) X(t_k)\right\} = \lambda \mathrm{Tr}\left\{\Lambda^\dagger_{:k+1} X_{k:0}\right\} \, \forall k \,, \tag{1.63}$$

with $\lambda = \frac{1}{\|X_{\text{target}}\|_2}$, and where $\Lambda^\dagger_{:k+1}$ is the *onto propagation* defined in Eq. (1.62).

Using these definitions the fidelity gradients, as derived in Ref. [78], are

$$\frac{\partial \mathcal{F}_{\text{PSU}}}{\partial u_{j,k}} = \lambda \, \mathrm{Re} \, \mathrm{Tr}\left\{ \mathrm{e}^{-\mathrm{i}\phi_\mathrm{g}} \cdot \Lambda^\dagger_{:\mathrm{k}+1} \left(\frac{\partial \mathrm{X_k}}{\partial \mathrm{u_{j,k}}}\right) \mathrm{X_{k-1:0}} \right\} \,, \tag{1.64}$$

where $\phi_g$ is the phase of $g$.

$$\frac{\partial \mathcal{F}_{\text{SU}}}{\partial u_{j,k}} = \lambda \, \mathrm{Re} \, \mathrm{Tr}\left\{ \Lambda^\dagger_{:k+1} \left(\frac{\partial X_k}{\partial u_{j,k}}\right) X_{k-1:0} \right\} \,, \tag{1.65}$$

In both cases the infidelity gradient is

$$\frac{\partial \mathcal{E}}{\partial u_{j,k}} = -\frac{\partial \mathcal{F}}{\partial u_{j,k}} \,. \tag{1.66}$$

Following the method in Ref. [76] the gradients for the trace difference infidelity, using $\Delta(T) = X(T) - X_{\text{target}}$, can be calculated as

$$\frac{\partial \mathcal{E}_{\text{FD}}}{\partial u_{j,k}} := -2\lambda \text{Tr} \left\{ \Delta(T)^{\dagger} X_{N_t:k+1} \left( \frac{\partial X_k}{\partial u_{j,k}} \right) X_{k-1:0} \right\} . \tag{1.67}$$

These are the final components required for the GRAPE algorithm. Note the propagation operator from the timeslot $k$ onward until final time $X_{N_t:k+1}$ used here is referred to as the *onward propagation*.

### 1.6.4 GRAPE overview

All the components for the GRAPE algorithm have been defined in this section. The algorithm, as applied in this study, is shown in the flow diagram Fig. 1.3. Further detail on some of the steps is given here.

1. Initialisation
   (a) define dynamics – specify drift and control dynamics generators
   (b) specify objective – specify initial and target state or operation
   (c) set cost function – choose cost (fidelity error) and corresponding gradient calculation functions
   (d) define timeslots – specify piecewise constant evolution parameters: total time $T$ and number of timeslots $N_t$
   (e) guess pulse – generate an initial pulse with randomised amplitudes
   (f) set term. conditions – specify termination conditions: target fidelity error $\mathcal{E}_{\text{target}}$, maximum iterations etc.
   (g) initiate MVO – initiate multi-variable optimisation method: BFGS variant
2. BFGS steps
   (a) calculate system evolution – solve equation of motion, e.g. Schrödinger

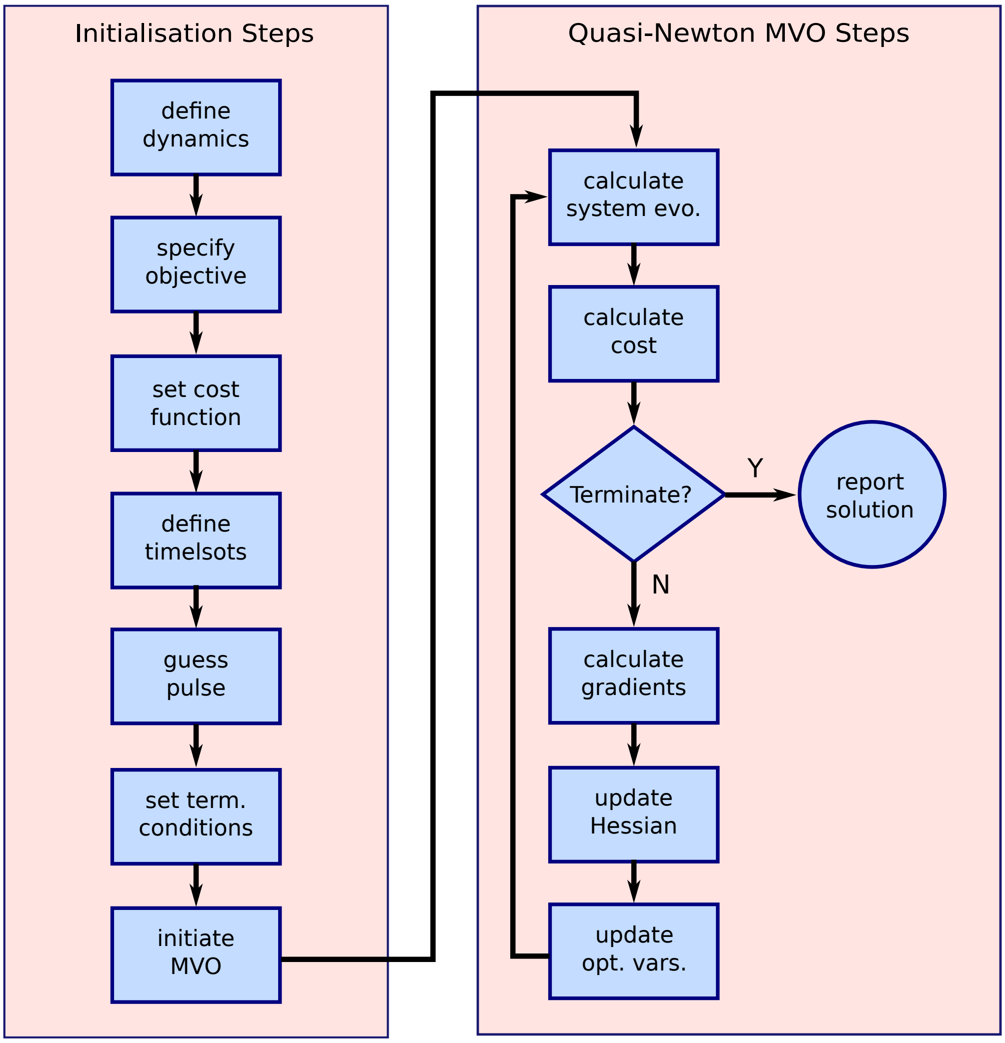


**Figure 1.3: Flow diagram of the GRAPE algorithm**. The Initialisation Steps are performed once. The Quasi-Newton Multi-Variable Optimisation (MVO) Steps are repeated until the method terminates, either having found a solution or otherwise. Further detail on the steps is given in the main text.

(b) calculate cost – compute infidelity, and potentially penalty functions

(c) check for termination – a solution has been found or early termination (discussed further below)

(d) calculate gradients – exact or approximate gradients with respect to the control variables

(e) update Hessian approximation – estimate second order gradients (MVO method dependent)

(f) update optimisation variables – set control parameters with values proposed by the MVO method

The initialisation steps are completed once, the BFGS steps are repeated until some termination has been met. A successful outcome would be when the cost function returns a value that is below a specified target $\mathcal{E}_{\mathrm{target}}$. In practice, if a target is reachable, then achieving infidelities orders of magnitude lower is simply a case of more iterations, and so in this study, which is mainly about whether targets in a wide range of scenarios can be reached in principle, target infidelities are set reasonably high, *e.g.* $1 \times 10^{-3}$. The BFGS method will terminate itself if it determines the function has converged on a local minima. It does this either by finding that the norm of the gradient is below some threshold, or that the reduction in the cost function in successive iterations is below some threshold. It is also possible to set iteration and processing time limits for practical reasons.

### 1.6.5 GRAPE implementation

The GRAPE algorithm was selected for this theoretical study of various control objectives in many-body systems for many reasons. It is well established technique that has been used to produce the results in many publications. It is comparatively efficient to compute, especially when analytical gradients can be used. This allowed for many different scenarios to be tested, and sufficient range of scale in the number of interacting entities for hypotheses to be trialled. The results presented herein are for model systems, and so the realistic modelling of physical system constraints is not necessary, which is the motivation behind some of the other algorithms, as discussed later in this chapter.

The GRAPE algorithm was implemented in QuTiP by the thesis author, and is used throughout this study. QuTiP is an open source Python toolkit for simulating quantum dynamics [16, 17]. QuTiP, particularly the control modules, and the other developments within it that were made by the thesis author specifically for this study, are described in some detail in Chapter 2. The aforementioned DYNAMO library [78] is also open source. It runs within the Matlab environment, which suits many researchers because they are experienced Matlab users. However, it can lead to restrictions in usage, as Matlab is not free to use, and for instance it is not available on some high-performance computing clusters. A library called Spinach for simulating the dynamics of spin systems in NMR experiments is developed and maintained by the Spin Dynamics group at Southampton University [85]. This is another open-source Matlab library that contains an implementation of GRAPE, and is specifically aimed at NMR experiment optimisation.

Matlab has been the industry standard tool for numerical methods for many years. However, as the Python libraries that offer equivalent functionally, in particular Numpy [86] and Scipy [87], have matured, increasingly many research groups have switched to using these in preference. Also, many students are now taught how to develop numerical methods using Python in their undergraduate courses. The cost benefits are obvious, both to academic institutions and the students and researchers, but there are also the other benefits associated with open access to source code. In the particular case of quantum dynamics simulation, QuTiP has become something of an industry standard. It has many thousands of users and hundreds of publications citing its use.

Although it has proven very successful in solving a wide range of quantum optimal control problems, GRAPE does have limitations. These are mainly related to its main strength, which is very computationally efficient convergence on a solution due to the use of analytic gradients. The analytic gradient calculations require that total evolution time $T$ is fixed for a particular pulse optimisation. Due to the inherent problem of decoherence due to interaction of quantum systems with the environment, *time optimal control*, in which the minimum time to achieve the target is sought, is a common objective. This is possible with GRAPE, but only through successive optimisation attempts with decreasing evolution times. Another issue is that there is nothing inherent in GRAPE to favour control pulses that are easier to implement experimentally. In many experimental setups pulses are produced by analogue devices that generate smooth waveforms. In these cases a parametrisation other than piecewise constant would more accurately simulate the pulse. Some of these limitations have been addressed by penalty function adaptations to GRAPE, as mentioned above in Subsection 1.6.1. Others have been addressed by alternative quantum optimal control algorithms.

### 1.6.6 Other QOC algorithms

#### Krotov-type

There are a class of methods referred to as Krotov-type, based on principles introduced in [88], which were first formulated to facilitate the soft landing of spacecraft. They are based on the variational principle for minimising some functional that includes the infidelity and any penalty functions. They are also based on a fixed set of control parameters, which are typically piecewise constant control functions. They differ from GRAPE primarily due to their update scheme, which is that control amplitudes in a timeslot are updated sequentially, rather than concurrently as they are in GRAPE. They are favoured over GRAPE by some research groups, including the Quantum Dynamics and Control Group, University of Kassel. A Krotov-type algorithm for quantum control pulse optimisation as an open-source library for Python, that uses QuTiP for computing the system evolution, has recently been released [89].

#### Chopped RAndom Basis algorithm

The Chopped RAndom Basis algorithm known as CRAB [90, 91] looks to find experimentally achievable pulses by allowing for parametrisation of pulses based on those available in the lab. For instance, it may be possible to construct a control pulse based on the sum waveforms, such as the modified the Fourier basis, which is typical in CRAB, given by

$$u(t) = \sum_{n=1}^{N_{\mathrm{F}}} \{a_n \sin(\omega_n t) + b_n \cos(\omega_n t)\} \tag{1.68}$$

where $\omega_n = \frac{2n\pi}{T} + r_n$ with $r_n$ uniformly distributed in $[-0.5, 0.5]$. The frequencies, based on the number of modes $N_{\mathrm{F}}$ and the randomly generated frequency offsets $r_n$, are fixed for a particular optimisation. The optimisation variables are the coefficients, these forming the vector in the parameter space

$$\vec{\alpha} = (a_1, b_1, \ldots, a_n, b_n, \ldots, a_{N_{\mathrm{F}}}, b_{N_{\mathrm{F}}}) \,. \tag{1.69}$$

Using such a basis it is possible to specify a smoothly varying control pulse with significantly fewer parameters than with a close approximation using a piecewise constant parametrisation.

The algorithm does not use analytic gradients, but rather uses a simplex optimisation method *Nelder-Mead* [92], which is feasible due to the small number of parameters. Nelder-Mead does not perform well with large dimensional parameter spaces. There are many variants of the method, but all are based on the concept of a simplex, which is a polygon in the $N_\mathrm{p}$ dimensional parameter space with $N_\mathrm{p} + 1$ vertices, *i.e.* in the two-dimensional hill descending example, this would be a triangle on the map. In the optimisation method these points are the $N_\mathrm{p} + 1$ lowest cost function evaluations so far. The primary step type is called a 'reflection', where the worst point is reflected about the centroid of the simplex outside of its bounds. If the cost is lower, then 'expansion' steps extend in the same direction while the cost is continued to be found lower, and the best point replaces the worst point. 'Contraction' steps follow the same direction, but inside of the simplex, with the assumption that a local minima is within it.

As it does not use gradients, then the penalties to the cost for maximum amplitudes, total energy, and total time can easily be added. The parameter $N_\mathrm{F}$ allows exclusion of higher frequencies, which may be unrealisable in an experiment. CRAB also addresses experimental constraints such as *ramping*, where a control field may have to start at zero amplitude and have a maximum gradient, or specific profile, that it can achieve when heading to maximum amplitude. This is done by specifying envelopes for the pulse.

The evolution of the system is calculated using a piecewise constant approximation to the smooth pulse. The precision of the approximation clearly depends on the granularity of the discretisation, with it tending to exact as $\Delta t \rightarrow 0$. This means many more matrix exponentiations than a GRAPE evolution with the equivalent number of optimisation variables. Simplex methods generally require many more steps for convergence than gradient based methods. These two factors make CRAB significantly more computationally expensive than GRAPE for an equivalent problem, so it is only appropriate when it is necessary to address experimental realisation.

The thesis author implemented the CRAB algorithm in QuTiP during the period of this study. However, it is not used to produce any of the results. As actual physical realisation here is unimportant, hence the shape of the pulse not relevant, then the faster convergence of GRAPE make it more suitable.

### Gradient Optimization of Analytic conTrols algorithm

Clearly, experimental realisation is a vital part of the scientific research process, and therefore the combining the features of the CRAB algorithm that support this in quantum control experiments, with the faster convergence of gradient based algorithms is an important aim of scientific tool development. The Gradient Optimization of Analytic conTrols algorithm or GOAT [82] appears to provide the advantages of physical motivated parametrisation of control functions with the fast convergence of gradient based optimisation methods.

An outline of the algorithm, as described in [82], is given below. First reframing the bilinear control system of Eq. (1.27) as

$$A(t, \vec{\alpha}) = A_0 + \sum_{j=1}^{N_c} u_j(t, \vec{\alpha}) A_j \,. \tag{1.70}$$

Here the components of $\vec{\alpha}$ determine the value of the control functions at time $t$, and are assumed accessible to the optimiser. They are the components of a parametrisation scheme, which could be any that is differentiable by the components, as $\frac{\partial X(t)}{\partial \alpha}$ needs to be derived for each component. The algorithm is centred around the co-evolution of propagator gradients, described by

$$\frac{\partial}{\partial t} \begin{pmatrix} X(t) \\ \frac{\partial X(t)}{\partial \vec{\alpha}} \end{pmatrix} = \begin{pmatrix} A & 0 \\ \frac{\partial A}{\partial \vec{\alpha}} & A \end{pmatrix} \begin{pmatrix} X(t) \\ \frac{\partial X(t)}{\partial \vec{\alpha}} \end{pmatrix} \tag{1.71}$$

representing multiple equations of motion, one for each of the components of $\vec{\alpha}$. Here $A = A(t, \vec{\alpha})$. The equations of motion can be solved by an ordinary differential equation (ODE) solver. The use of numerical ODE solvers in time-dependent quantum dynamics is well established [16, 17]. Solving Eq. (1.71) to $t = T$ gives $X(T)$ and the $\frac{\partial X(T)}{\partial \vec{\alpha}}$.

As an example of this applied to cost gradient evaluation, if the $A$ are skew Hermitian, that is $A = -iH$, then the evolution is unitary, that is $X(t) = U(t)$. In this case then the analytic gradients for projective special unitary infidelity, from Eq. (1.45) $\mathcal{E}_{\text{PSU}} := 1 - |g|$ with $g = \lambda \text{Tr}\left\{U_{\text{target}}^{\dagger} U(T)\right\}$ and $\lambda = 1/\|U_{\text{target}}\|_2$, can be calculated as

$$\frac{\partial \mathcal{E}_{\text{PSU}}}{\partial \vec{\alpha}} = -\lambda \,\mathrm{Re}\,\mathrm{Tr}\left(\mathrm{e}^{-\mathrm{i}\phi_{\mathrm{g}}} \mathrm{U}_{\mathrm{target}}^{\dagger} \cdot \frac{\partial \mathrm{U}(\tilde{\alpha}, \mathrm{T})}{\partial \tilde{\alpha}}\right) \,. \tag{1.72}$$

Note that no ‘onto’ propagation (Eq. (1.62)) is required with this method.

This then provides an alternative method for calculating gradients of cost functions with respect to parameters that drive the control system for any practical parametrisation scheme and infidelity measure. This means that it provides all the necessary ingredients for a gradient based multi-variable optimisation method, such as BFGS. Hence a procedure otherwise almost identical to that for GRAPE described in Section 1.6.4 can be applied to complete the GOAT algorithm.

Ref. [82] describes how the parameter space can be mapped onto other subspaces in order to apply constraints such as amplitude bounds and pulse ramping. It is not immediately apparent how penalties such as total time and total energy could be applied to the cost function, but assuming that this is possible, then it would seem that GOAT provides all the experimental setting advantages of CRAB, but with a more computationally efficient method.

### 1.6.7 Local traps

In general a multi-variable optimisation method will converge on the first local minima it finds from the initial random guess (or analytically estimated) parameter space vector. In the hill descending analogy this could be some elevated bowl. If the global minima is the objective, which may be degenerate, for instance in the quantum control case there may be many solutions that achieve the target fidelity, then finding a sub-optimal minima is a failed search. For obvious reasons these sub-optimal minima are referred to as *traps*. There are sophisticated methods for escaping these traps, such as an adapted technique from simulated annealing: applying a small ‘kick’ to try and get over the lip of the bowl and start descending downhill again. The ‘Dresssed CRAB’ or DCRAB algorithm [93] looks to address this by adaptively changing the basis for its parametrisation when a local minima is encountered that is outside of the target infidelity.

A commonly used, less sophisticated, method is to simply try different initial guess solutions until the optimisation achieves its goal. With the infidelity minimisation goal the lowest possible global minimum is zero, and so it is easy to set some target threshold. Without some understanding of the topology of the parameter landscape it is difficult to conclude much when the goal is not reached, even after many attempts, other than there are at least some non-global minima.

The Rabitz group at Princeton have studied the control parameter landscape extensively [94, 95, 96, 97]. They prove that no local sub-optimal minima exist for controllable systems when the controls are unconstrained [94]. This was refuted in [98] by counter-example, showing that traps exist in three-level systems with a forbidden transition between the lowest two levels, know as $\Lambda$ systems. The Rabitz group position was reduced to claim “almost always trap free” in [97]. Meaning that control optimisation is trap free in almost all types of quantum system, and in particular, those systems most of interest in quantum computing – coupled qubit systems. The proof is based on a geometric analysis of the control parameter landscape, which is shown to be convex for systems of interest in quantum information processing.

Many studies using numerical quantum optimal control encounter traps, especially those involving many-body quantum systems [99]. Any parametrisation of a control pulse introduces some constraint, even if the there are no bounds on amplitude, as it limits the set of potential functions. And so the debate continues, with those who are working practically to find successful control pulses typically on side of traps being a reality. Whether the traps are encountered due to parametrisation, or they a true traps due to the physics of the system, their potential existence needs to be considered. Certainly it would not be valid to claim that a target is unreachable because a quantum optimal control algorithm failed to find a solution from a single initial guess.

# Chapter 2

# Methods

The original implementation of the HEOM solver in QuTiP was developed by Dr. Neill Lambert, RIKEN, Japan. The thesis author improved this code in terms of clarity, documentation and efficiency (quantified in this chapter). The description in Subsection 2.8.2 is of the code resulting from their combined effort. The other methods described in this chapter were developed by the thesis author, although often building on existing libraries, which are clearly attributed.

## 2.1 Introduction

This chapter gives details of the numerical and mathematical methods used to produce the results in the later chapters. The purpose of the chapter is to give some additional description of how some of the numerical results were produced, and provide sufficient detail for full reproducibility of the results. A reader that is only interested in the scientific results may want to skip this chapter. Many of this chapter's sections, in particular Sections 2.6, 2.7 and 2.8, may make more sense if the corresponding later chapters have been read first.

All the numerical methods described in this chapter were implemented in Python [100]. The numerical library NumPy [86] and mathematical functions in SciPy [87] were used extensively. The Pandas library [101] was used to read data from spreadsheet files when necessary. The computer algebra system SymPy [102] was also used for some calculations. All plots were produced using Matplotlib [103], except the 3D surface plots in Fig. 3.1 and Fig. 2.4, for which Mathematica [104] was used, as the surface plotting was found to be superior to that in Matplotlib. Jupyter notebooks [105] are an excellent medium for sharing numerical codes, as they combine codes with narrative and illustrations. The codes for this study were mainly developed in Python modules and executed through scripts, as this allows for automation, such as on a remote system, like a high performance computing cluster. Linux Mint desktop [106] was the operating system for the development and local execution of code.

In general Python 3.x was used, moving on from initially using Python 2.7, as recommended by QuTiP colleagues. Many numerical libraries will freeze support for Python 2.7 in the near future, so all scientific Python users should look to move to version 3.x if they have not already done so. The Anaconda distribution [107] of Python is preferred by the thesis author, as it includes the libraries typically used in scientific research. It also allows for multiple environments, where different versions of libraries can be installed, so application specific environments can be set up that do not risk other applications, including use of Python by system components. The NumPy and SciPy libraries in Anaconda are compiled against the Intel Math Kernel Library (MKL) [108], which claims to provide the best performance on Intel CPU-based systems. Instructions for setting up an Anaconda environment for using QuTiP are available in the documentation [114].

## 2.2 QuTiP

QuTiP [16, 17], Quantum Toolkit in Python, is the most widely used and comprehensive software for solving quantum dynamics. QuTiP development was started in 2011 by the Quantum Condensed Matter Group (QCMG) in RIKEN, Japan. The library was initially primarily developed by Robert Johansson and Paul Nation during their postdoctoral research appointments at RIKEN. Their motivation was to provide an open-source and open-platform library to replace the Quantum Optics Toolbox [109] that, although still widely used at that time, had not been maintained for ten years, and requires the Matlab software [110] to run. Matlab is widely used in scientific research, however the cost is prohibitive to adoption by all institutions, whereas in contrast, Python is free to use and the runtime components are available for almost any computing platform. By choosing Python the developers of QuTiP have made numerical solving of quantum dynamics available to everyone who may want to, including researchers in all labs, large or small, wealthy or less so, undergraduate students, and hobbyists.

The primary focus of QuTiP was originally in simulating open quantum systems through master equation and Monte-Carlo solvers. The main physical systems they were looking to simulate environment interaction effects within were quantum optics, trapped ions, superconducting circuit devices, and nanomechanical systems. As has been discussed elsewhere in this thesis, these continue to be central in quantum physics research. The popularity in, and value to, the research community is evident through: the number of papers citing use of QuTiP is in excess of 370; the number of unique visitors to the website, which was 25,473 in 2016; and the number of downloads of the `conda-forge` package – over fifty-thousand downloads since March 2016.

The QuTiP project has always been, and continues to be, passionately supported by its main sponsor Dr. Franco Nori, Chief Scientist, and head of the Theoretical Quantum Physics Laboratory, at RIKEN, who has attracted funding and researchers to continue development and support the library since its conception. There has always been a strong commitment to coding standards, documentation and support, which has lead to its longevity. This is in contrast to many other scientific software libraries for quantum dynamics, which fall into disuse, due to the well known problems of maintaining open-source software, which typically relies on volunteers for continued maintenance. The QuTiP library is currently predominately maintained by the author of this thesis, Nathan Shammah (RIKEN), Eric Giguere (Université de Sherbrooke) and Shahnawaz Ahmed (Chalmers University, Gothenburg).

The Github development platform [111] makes collaboration on open-source projects manageable. This has allowed QuTiP to benefit from the many contributions of researchers, extending its range of capabilities. The contributions are acknowledged on the QuTiP website [112], and statistics relating to the contributions are accessible via the 'Insight' tab of the core `qutip` and related repositories combined under the project [113]. The thesis author's main contribution to QuTiP is discussed below in Section 2.3. He made two invited visits to RIKEN during the period of this study, six weeks in 2016 and five weeks in 2017. During these visits many technical aspects of QuTiP were improved, in particular the installation on Microsoft Windows, which is now possible through the Conda-forge channel described. The more scientific developments completed during the RIKEN visits are described in Section 2.8.

The simplest way to install QuTiP is via the Conda-forge channel, although there may be performance benefits to ‘installing from source’, which is described in the QuTiP documentation [114]. The installation from source means that C++ components of QuTiP are compiled specifically on the local machine. It also allows for the selection of options such as using OpenMP [115], which enables some of the QuTiP solvers to parallelise execution through multi-threading. Installation from source is typically trivial on Unix type operating systems (which includes Linux and MacOS). However, on Microsoft Windows, additional development packages are needed, as the Microsoft OS does not have a C++ compiler built in as standard. These are freely available, but can be awkward to set up. There are some runtime components of QuTiP that require a C++ compiler, these are the highest performing solvers for time-dependent Hamiltonian and Lindblad operators. The Unix-based operating systems are therefore recommended for using QuTiP, though all features are accessible under Microsoft Windows, and the vast majority do not need a C++ compiler.

## 2.3 QuTiP control

A control optimisation library was developed in Python by the author of this thesis as part of his Masters project at Aberystwyth University. The library was provisionally named Qtrl. The motivations were similar to those behind the development of QuTiP. The Quantum Control Theory Group under Dr. Daniel Burgarth were using the DYNAMO [78] package for numerical optimisation of control in their research. The Group had limited access to Matlab licenses, which was required for using DYNAMO, and therefore a license-free alternative provided the Group with much greater flexibility when control optimisation simulation was needed. A particular motivation for the license-free alternative was the opportunity to use the high-performance computing (HPC) clusters that were available for use by the Group, thanks to the HPC-Wales (HPCW) project (now Supercomputing Wales) [116]. These clusters did not have Matlab multiprocessing licenses.

Many aspects of Qtrl were based on DYNAMO, in particular the use of the BFGS algorithm in performing the multi-variable cost optimisation and the exact gradient calculations. The Group wanted to take a long-term view on the new control optimisation library, with plans to extend its capabilities beyond that of DYNAMO, and so a customisable design was sought, and therefore an object-oriented design was chosen. The closed system, unitary dynamics control optimisation, which was developed for the Masters project were validated against DYNAMO examples. The benefit of using HPC clusters was demonstrated by repeating the optimisations performed to find minimum gate time in the 'Spin-star' simulation that other Group members had recently completed [99]. The optimisations performed using DYNAMO on a desktop workstation took many weeks. These same optimisations took a few days using Qtrl and the parallel processing available on HPCW. A greater number of repeats were possible in the higher-dimensional systems, reducing the instances of data points resulting from local traps.

To complete the research in this study, and support the research of colleagues, the author of this thesis extended the Qtrl library to support optimisation using symplectic and Lindbladian dynamics. These extensions are described in Section 2.6 and Section 2.8 respectively. The intention was always to make Qtrl available to the wider quantum control research community. Qtrl was made publicly available through merging into QuTiP in January 2015. After this merger Qtrl would be more accurately referred to as 'the control modules in QuTiP' or QuTiP Control, however the shorthand of Qtrl will continue to be used in this chapter. An invited research visit was made to the Institute for Complex Quantum Systems, University of Ulm, in April 2015. Three weeks were spent implementing the CRAB algorithm [90, 91], which is described in Section 1.6.6, in Qtrl. The Qtrl library continued (and continues) to be extended and improved throughout the period of this study. Much of this development is to improve integration with the rest of the QuTiP library to take advantage of the dynamics solvers in control optimisation. Customisations have been made for specific projects, including implementation of time-dependent control operators used in the collaboration to optimise fast three-qubit gates [117]. The design also proved flexible enough for colleague Dr. Jukka Kiukas to customise Qtrl for parameter estimation [118].

### 2.3.1 QuTiP control object model

The schematic in Fig. 2.1 shows how the main objects in Qtrl interact. Their individual purpose, and how the thesis author used them in this study, are described below. Their properties and

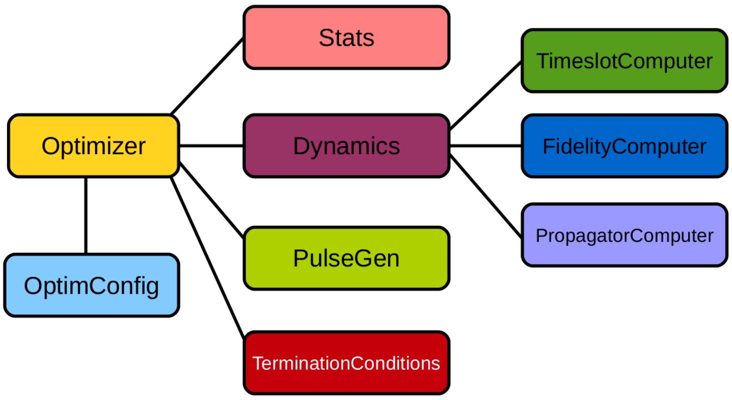


**Figure 2.1: QuTiP Control object model.** A schematic showing the base classes of the Qtrl objects and hierarchy in which they are functionally connected.

methods and described in the code documentation [114].

### OptimConfig

The `OptimConfig` object is used to hold configuration parameters that must be set prior to the instantiation of the other objects. Primarily these parameters are the subclass types for the other objects, but it is also a logical place to store user-defined properties that determine the optimisation procedure. The thesis author used this object throughout this study to store parameters such as the number of repeats and multiprocessing options.

### Optimizer

This acts as a wrapper to the `scipy.optimize` functions that perform the work of the pulse optimisation algorithms. Using the base class, the user can specify which of the optimisation methods are to be used. There are subclasses specifically for the BFGS and L-BFGS-B methods. There is another subclass for using the CRAB algorithm. The object also triggers the computation of the dynamics evolution required to compute the cost (infidelity), based on the control function parameters, which this object attempts to optimise. The bounds for the parameters are also set in this object, which for piecewise constant control functions are the timeslot amplitudes. The bounds can be set either for individual parameters or all of them. Throughout this study the thesis author used the `OptimizerLBFGSB` subclass. This uses the limited-memory and bounded variable version of the the

BFGS algorithm introduced in Section 1.6.2. Specifically the `scipy.optimize.fmin_l_bfgs_b` function wraps the Fortran implementation presented in [81, 119]. This was found to be significantly more efficient than the `scipy.optimize` BFGS method by the thesis author during his masters project.

### Dynamics

This is mainly a container for the lists that hold the control amplitudes, dynamics generators, propagators, and time evolution operators in each timeslot for the piecewise constant computation of the evolution. It also holds the dynamics generators for the controls and any pre- or post- factor or operation for the dynamics generator in computing the evolution step. The combining of dynamics generators is also completed by this object. Different subclasses support a range of types of quantum systems, including closed systems with unitary dynamics, systems with quadratic Hamiltonians that have Gaussian states and symplectic transformations. The general matrix subclass can be used for open system dynamics with Lindbladian operators, as there is no pre- or post- factor or operation when computing the timeslot propagator. The `DynamicsUnitary` subclass applies the skew Hermitian pre-factor $-\mathrm{i}$ to the combined dynamics generator and contains the spectral decomposition method used for the propagator compute. The `DynamicsSymplectic` subclass applies the transformation $-(\cdot)\Omega$ or $\Omega(\cdot)$ (see Section 3.3) as set by user configuration.

### PulseGen

There are many subclasses of pulse generators that generate different types of pulses as the initial amplitudes for the optimisation. There is a specific subclass that is used by the CRAB algorithm to generate the pulses based on the basis coefficients that are being optimised. Often the goal cannot be achieved from all starting conditions, and then typically some kind of random pulse is used. For this study the thesis author used the `PulseGenRandom` subclass, which produces uniformly distributed amplitudes in each timeslots, with the range set by upper and lower bounds. Other random pulse types were tried when control objectives were difficult to achieve, but no quantifiable advantage was found.

### TerminationConditions

This holds all the properties that will determine when the single optimisation run terminates. Primarily this is the cost (infidelity) target for a successful optimisation. Limits can be set for number of iterations, processing time (known as 'wall time'). Termination can also be triggered by the gradient norm falling below some threshold, or cost reduction between steps not meeting that threshold.

### PropagatorComputer

This object computes propagators from one timeslot to the next, and also the propagator gradient, which is used in the infidelity gradient computation. These are closely linked to the type of dynamics in use, and so each `Dynamics` subclass uses a default `PropagatorComputer` subclass. The propagator subclass for unitary dynamics is `PropCompDiag`, which uses the diagonalisation of the combined Hamiltonian in the timeslot to compute the propagator and its gradient, as described in Section 1.6.3. The `PropCompFrechet` subclass can be used for a general dynamics generator to compute the propagator and its gradient, also described in Section 1.6.3. It uses the `scipy.linalg.expm_frechet` function, which is found to be more efficient than exponentiating the augmented matrix.

### FidelityComputer

The subclass of the fidelity computer determines the type of fidelity (or infidelity) to use in computing the optimisation cost, and gradient of this with respect to the optimisation variables. These are again linked to the type of dynamics in use, and so there is a default for use with each `Dynamics` subclass. The subclass for unitary dynamics is `FidCompUnitary`, which computes the unitary fidelity Eq. (1.45) and gradients Eq. (1.64), or Eq. (1.46) and gradients Eq. (1.65). The trace difference infidelity is more generally applicable, for which `FidCompTraceDiff` can be used to compute the infidelity Eq. (1.48) and gradients Eq. (1.67). Subclasses of `FidelityComputer` are the most commonly user-developed customised Qtrl subclasses.

### TimeslotComputer

The `TimeslotComputer` object computes the system evolution, using the `Dynamics` and `PropagatorComputer` object methods. It also calculates the forward propagation and *onto* or *onward* propagation (see Section 1.6.3) that are used by the fidelity computer.

### OptimResult

The result of a pulse optimisation is returned as an object with properties describing the outcome, including the infidelity, reason for termination, performance statistics, final evolution, etc.

### Stats

Performance data are optionally collected during the optimisation. This object is shared by the other objects to provide a single location to store, calculate and report run statistics.

## 2.3.2 Executing a Qtrl optimisation

There are a number of ways to use the objects described above to perform a control optimisation. QuTiP typically provides a single function call interface for its solvers, and so Qtrl also provides this. This is the simplest way to perform a single optimisation. However, if the optimisation procedure is more involved, or the user wishes to make use of a parameter file for configuration, then instantiating the objects directly and calling their methods is recommended. If there are many different scenarios to be tried, and in particular if a remote computing resource, such as a HPC cluster, then the latter method is advised. This allows for using static code, with different parameter files for each scenario. Consequently, the parameter file configuration method was mainly preferred in this study. The two methods are described further below. Fully documented examples of using Qtrl can be found as Jupyter notebooks in the QuTiP tutorials [120], and as Python scripts in the Qtrl examples repository [121].

### Pulse optimisation functions

The simplest method for optimising a control pulse using Qtrl is to call one of the functions in the `pulseoptim` module. These are fully documented in the QuTiP API documentation [114]. The `pulseoptim` functions automate the creation and configuration of the necessary objects, generation of initial pulses, running the optimisation and returns the result. The `optimise_pulse` function can be used for all types of dynamics and algorithms (GRAPE or CRAB). It defaults to the general matrix dynamics, with corresponding Frechet propagator and trace difference infidelity. The `optimize_pulse_unitary` function is specifically for unitary dynamics, and hence uses the default propagator and fidelity computers for unitary dynamics. There are also functions specifically for the CRAB algorithm, `opt_pulse_crab` and `opt_pulse_crab_unitary`. They have the same functionality as the general functions, but default to the CRAB algorithm, and have argument names that are more familiar in that application. Examples using the functions can be found in the QuTiP tutorials [120].

The `create_optimizer_objects` function can be used to generate and configure all the objects without performing the actual optimisation. This allows for the objects to be additionally configured, such as specifying the initial pulse, before calling the optimisation. This would be more efficient when repeating runs with different starting conditions. An example of this method is given in the QuTiP tutorial 'Optimal control, QFT'. This method also enables use of customised subclasses, for which `ctrl_pulse_optim-example-QFT-custom_fidelity.py` is an example in [121].

### Direct Qtrl object instantiation

The Qtrl objects can be instantiated (created) in the usual way for a Python object. The `OptimConfig` must be instantiated first, as this is a required instantiation parameter for the other objects. As an example, this could be

```
import qutip.control.optimconfig as optimconfig
cfg = optimconfig.OptimConfig()
```

When a `Dynamics` (or subclass) object is instantiated the default `PropagatorComputer`, `FidelityComputer` and `TimeslotComputer` subclass objects will be created. These can be overridden, for instance if a custom fidelity called `FidCompChoi` was to be used, then this could be specified by

```
import qutip.control.dynamics as dynamics
dyn = dynamics.DynamicsGenMat()
dyn.fidelity_computer = FidCompChoi()
```

The `Optimizer` object is typically created last, as this requires `OptimConfig` and `Dynamics` objects at instantiation.

Each of the objects can be configured through their properties in the code, or via a configuration file. The `loadparams.load_parameters` function will read property values from a file in ‘ini’ format. The data types are assumed from the object property, so if these are not defined by the class, then a default value must be set before loading from file. The script `ctrl_pulse_optim-example-Hadamard-load_params.py` [121] provides a full working example of this. There is an example from this study in the *in situ* optimisation code repository [122] – see `qsoconfig` module.

## 2.4 High performance computing

The Supercomputing Wales (SCW) [116] project provides free access for researchers in the participating Welsh universities, Aberystwyth, Bangor, Cardiff and Swansea, to high performance computing (HPC) clusters. SCW is the successor to the HPC Wales project (HPCW). It is more focussed on scientific research than its predecessor, which had a focus on industrial collaborations. The SCW project brought funding for new hardware. These systems were brought online in 2018, making 12,000 cores available to researchers.

The computing resources of HPCW were used to execute numerical codes throughout this study. Although free, usage was limited to 160 cores (concurrent) and 2 days maximum time per job (latterly extended to 3 days). The main cluster used had approximately 3000 cores, each Intel Xeon CPU E5-2670, 2.60GHz, configured as nodes with 16 cores and 64GB of RAM. All systems operated with Linux CentOS release 6.7. Job submission, scheduling and general management is done through the Slurm workload management system [123].

In most use cases the multiple cores were exploited by running the same codes, but with different parameters. This is easily achieved in a single Slurm job using GNU Parallel [124]. Specific parameters can be selected by passing an index via the command line, which can be used in a calculation or as a list index. Each process writes its own output file, which can be collated and analysed with other scripts. This type of usage is often referred to as ‘embarrassingly parallel’, as it is very easy to implement, compared with multiple core processing where there is communication between processes. One use case required dynamic process spawning, for which the Python `multiprocessing` module was used. It is described in Subsection 2.7.2.

## 2.5 Lie tree algorithm

If the columns of an $l \times m$ matrix $L$ are vectorised elements of a Lie algebra, then the *rank* of $L$ will give the number of linearly independent elements. This mathematical tool can used in an algorithm to determine if the Lie algebra rank condition is met for a set of generators. Note that for a generator element that is an $n \times n$ matrix $l = n^2$. This is also the maximum possible rank for the matrix $L$. This corresponds with the maximum dimension of a subgroup of the general linear Lie algebra.

The algorithm described below is based on the method provided in section 3.2.1 of [19]. The process flow is illustrated in Fig. 2.2 The algorithm starts with a set of generators $\mathfrak{L}_0 = \{A_1, \ldots, A_s\}$, which will for instance be the drift and control generators of a controlled quantum system (the elements of set $\mathfrak{L}$ are the initial elements of $l$ in Fig. 2.2. The initial matrix $L$ will be constructed by vectorising each element of $\mathfrak{L}_0$ and column stacking them. The initial rank is calculated $r = \operatorname{rank}(L)$. The target $r_{\text{targ}}$ for the rank is determined based on the Lie algebra type, *e.g.* for $\mathfrak{u}(n)$: $r_{\text{targ}} = n^2$, for $\mathfrak{su}(n)$: $r_{\text{targ}} = n^2 - 1$, and for $\mathfrak{sp}(2n, \mathbb{R})$: $r_{\text{targ}} = n(2n + 1)$. Further detail on some of the steps is given below.

- select first $A$ from $l$: start iteration of all operators in $l$, referred to as outer loop. Note that $l$ may be extended during the process.

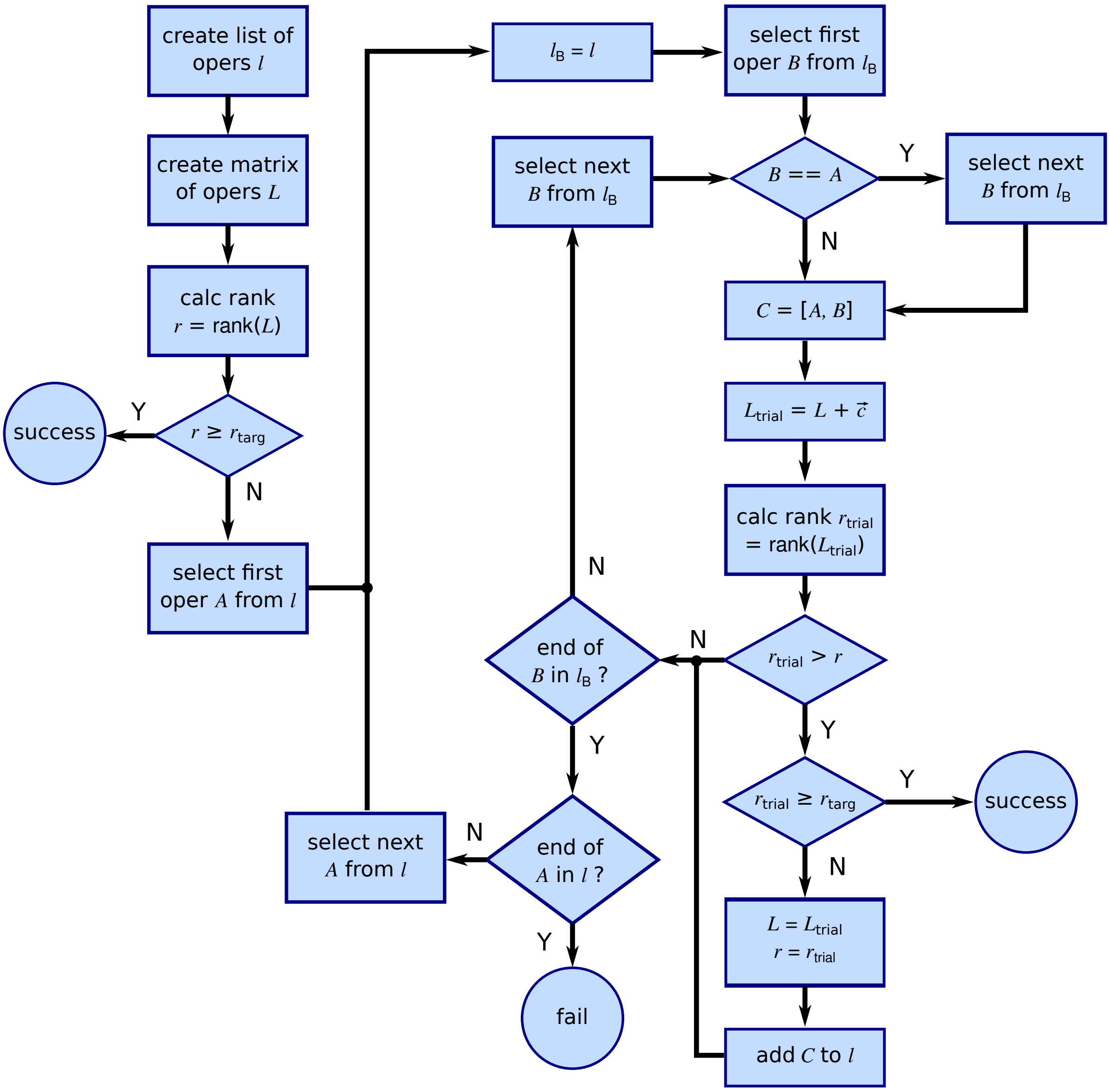

create list of opers $l$
create matrix of opers $L$
calc rank $r$ = rank($L$)
$r \geq r_{\mathrm{targ}}$
Y
success
N
select first oper $A$ from $l$
$l_{\mathrm{B}} = l$
select first oper $B$ from $l_{\mathrm{B}}$
select next $B$ from $l_{\mathrm{B}}$
$B == A$
Y
select next $B$ from $l_{\mathrm{B}}$
N
$C = [A, B]$
$L_{\mathrm{trial}} = L + \vec{c}$
calc rank $r_{\mathrm{trial}}$ = rank($L_{\mathrm{trial}}$)
$r_{\mathrm{trial}} > r$
N
end of $B$ in $l_{\mathrm{B}}$ ?
N
Y
Y
$r_{\mathrm{trial}} \geq r_{\mathrm{targ}}$
Y
success
N
$L = L_{\mathrm{trial}}$
$r = r_{\mathrm{trial}}$
add $C$ to $l$
select next $A$ from $l$
N
end of $A$ in $l$ ?
Y
fail

**Figure 2.2:** Flow diagram of the Lie tree algorithm.

- $l_{\mathrm{B}} = l$: a copy of the operator list is made that will be iterated over, referred to as the inner loop.

- $B == A$: skip to next operator, as commutator will be $0$.

- $C = [A, B]$ calculate commutator.

- $L_{\mathrm{trial}} = L + \vec{c}$: extend the matrix $L$ by adding column of vectorised $C$.

- $r_{\mathrm{trial}} > r$: if the rank has not increased then skip to the next operator in the inner loop.

- $r_{\mathrm{trial}} \geq r_{\mathrm{targ}}$: if the rank target has been met, then the criterion have been satisfied.

- end of $B$ in $l_{\mathrm{B}}$: all commutations have been tried in the inner loop.
- add $C$ to $l$: extend the list of operators to test commutators for. Meaning that $C$ will be the $A$ in some future outer loop iteration.
- end of $A$ in $l$: all operators have been tried in the outer loop. If the target rank has not be met, then criterion are not satisfied.

This is not the most efficient implementation of the Lie tree algorithm, as there is potential for repeated testing of commutators. However, it is robust, and was found to be sufficiently fast for usage in this study. The algorithm is known to perform well for small systems. However, for larger systems it becomes increasingly difficult to distinguish whether two generators are linearly independent. For systems with $n \geq 14$ 64 bit numbers were found to provide insufficient precision to reliably determine the linear independence of generators.

**Python implementation**

For implementing this algorithm in Python `numpy.ndarray` can be used for the Lie algebra elements $A$ and $L$ matrix. Standard Python `list` can be used for the sets $\mathfrak{L}$. The rank can be calculated using `numpy.linalg.matrix_rank`. Vectorisation of the Lie algebra elements can be done using `v = A.reshape((-1, 1))` and appending a column to the $L$ matrix using `L_trial = numpy.hstack((L, v))`.

## 2.6 Symplectic numerics

### 2.6.1 Symplectic operator optimisation

The unbounded property of matrices belonging to the symplectic group $\mathrm{Sp}(2n, \mathbb{R})$ leads to additional challenges in numerical simulation of control optimisation when the target is some symplectic transformation. As discussed in Section 3.5.1, recurrence is only guaranteed when the combined Hamiltonian generator is positive(negative)-definite, which as seen throughout Chapter 3 is not generally the case. This gives the potential for elements of the matrices representing the symplectic transformation to take extreme values, which are either too small are too large to be represented by the data types used in most numerical libraries. For instance, the `numpy.float64` data type has a maximum value limit $O(10^{308})$. This may seem like a very large number, but due to the evolution being computed as a product of exponentials, Eq. (3.17), this can often be exceeded, especially when the allowed evolution time is large or control amplitudes are high. These extreme values relate to intensive squeezing of modes. They result in an `OverflowError` when attempting to set a NumPy array element beyond the maximum value.

The symplectic dynamics control optimisations were also found to be more prone to local minima traps than with unitary dynamics. This is potentially due to trajectories in the control parameter space being taken from which there is no way back. The parameter landscape was not studied in any detail, but rather the trapping circumvented, using a combination of methods to attempt to reach targets. As well as the brute-force method of trialling many random initial guess pulses, pre-filtering, and tactical use of pulses with known outcomes close to the target, were employed.

### Single-mode system control optimisation

In the unstable single-mode reachable set control optimisations, the results of which are seen in Fig. 3.1, the reachable points were found to be only a small proportion of the available space. Intuitively, any initial guess that is far from the 'basket' is less likely to result in an unsuccessful optimisation. However, the location of the reachable set is unknown prior to the start of the search. Due to the many points in the space to be tested, it is infeasible to try a great many optimisations from initial guesses for each target. The set of targets is defined by the singular value decomposition (SVD) parameters (see Section 3.5.3): rotation angles $\theta, \phi$ and squeezing parameter $z$. The results presented are a subset of those attempted, 25 values for $\theta$ and $\phi$, and 15 values for $z$ were used to generate 9,375 potential targets. This was repeated with a set of drift parameter $c$ values $\{0.0, \pm0.5, \pm0.9, \pm0.99, \pm1.01, \pm1.1, \pm1.5\}$, and for each these a set of total evolution time $T$ values $\{0.1, 0.5, 1, 2, 3, 4, 5, 7, 10, 20, 50, 100\}$. The process for finding the reachable targets is illustrated in Fig. 2.3. Further detail on the steps is given below.

- for each target

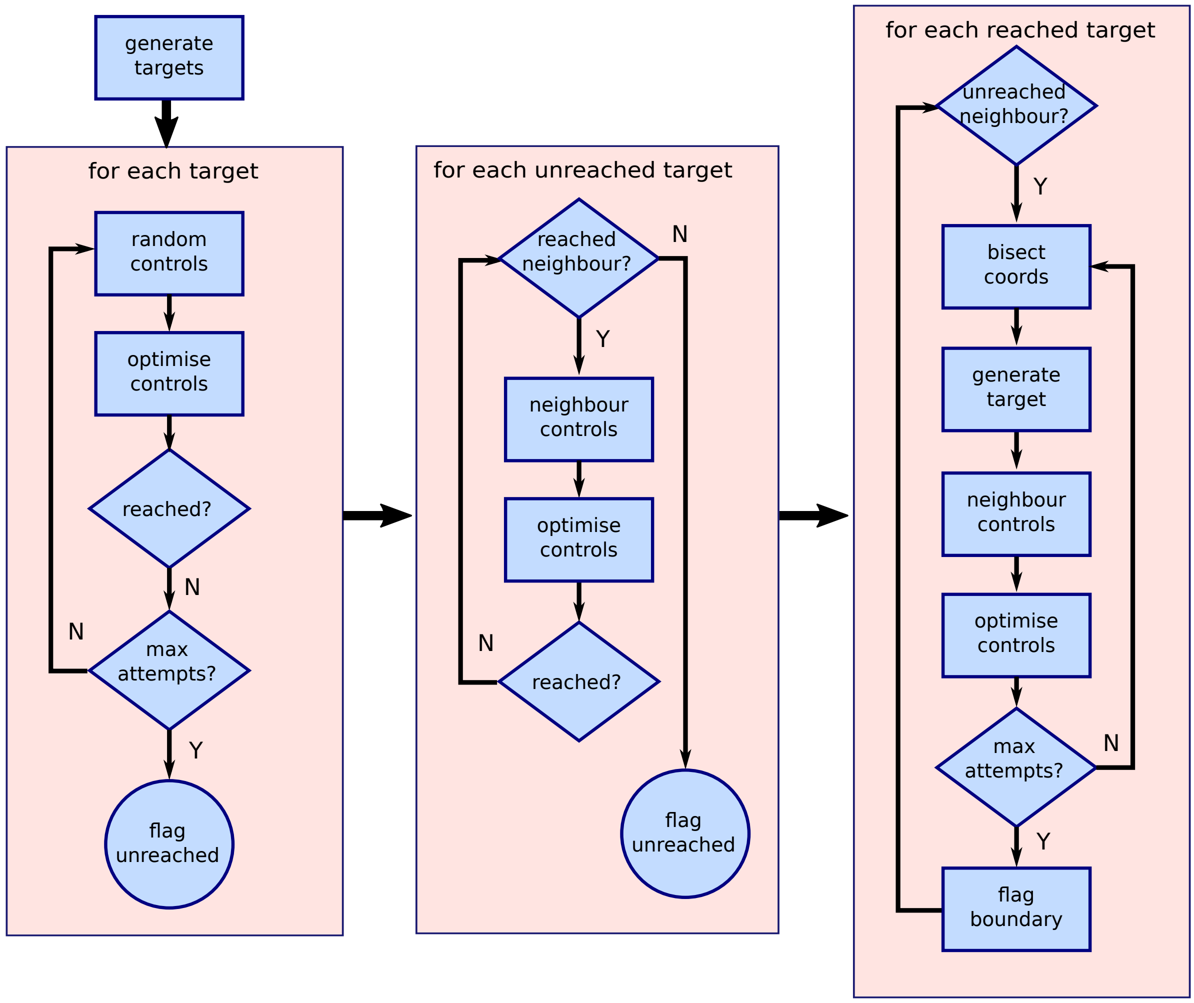


**Figure 2.3:** Flow diagram of reachable points search.

- random controls: generate set of random control timeslot amplitudes.
- optimise control: GRAPE control pulse optimisation.
- reached: if target is reached, then move to next target.
- max attempts?: a maximum of 10 sets of random controls is tried for each target.

- for each unreached target
  - reached neighbour?: any neighbour for which the target was reached, that has not be tried yet in this process stage.
  - neighbour controls: take final control pulse for reached neighbour as initial pulse.
  - optimise control: GRAPE control pulse optimisation.

  - reached: if target is reached, then move to next unreached target.

- for each reached target
  - unreached neighbour?: any neighbour for which the target was not reached, that has not be tried yet in this process stage.
  - bisect coords: bisect the line between the reached point and unreached neighbour.
  - generate target: new target from bisection point.
  - neighbour controls: take final control pulse for reached point as initial pulse for new target.
  - optimise control: GRAPE control pulse optimisation.
  - max attempts?: a maximum of 4 bisection attempts is tried.
  - flag boundary: the furthest reached point from the original reached point is flagged as a boundary point.

A *neighbour* in the above refers to any point immediately adjacent in the orthogonal directions in the SVD space, and any point on the immediate diagonals in the $\theta$ and $\phi$ directions. Despite the many initial targets attempted in the space, this still does not give a clear illustration of the shape of the reachable set, hence the bisection method used to more precisely find the boundary of the reachable set. These boundary points give the clear illustration of the bowl or basket-like shape of the reached sets seen in Fig. 3.1.

### Two-mode system control optimisation

The SVD parameter space for two-mode symplectic is 10-dimensional. This makes an evenly distributed reachability investigation, such as that undertaken for the single-mode system, infeasible. The main question investigated is whether squeezing is inevitable for non-neutral generators, hence the focus on a systematic search in the squeezing dimensions. Some choice of rotation angles is also required. As there is no *a priori* knowledge on what these would be, then a random selection seems a good option to avoid bias. The typical operations in quantum options are phase shifts of $\pm\pi/n$, where $n$ is some integer, typically $1, 2, 4$. And so angles chosen randomly from these would seem like a good option. As this approached failed to find any reachable targets, then some characterisation of the reachable set was sought through analysing randomly evolved operators, and attempting to reach targets shifted slightly from the evolved operator. For this the decomposition into SVD parameters is required, the method for this is described in Subsection 2.6.3.

### Oscillator chain control optimisation

The main challenge in optimising the controls to achieve the targets in the oscillator chain models, as presented in Section 3.7, was the extreme values of the evolved operator matrix that were discussed at the beginning of this section. The amplitude bounds of $[-10, 10]$, that were found to be necessary to reach targets, could result in `OverflowError`, even for the initial guess pulse.

To avoid solutions that have excessive mode squeezing, then the optimisation can be completed in a few stages. Firstly, the initial pulse for the optimisation can be chosen from a set of random pulses based on which pulse gives the minimum initial infidelity. Sampling ten initial pulses was found to work well. Secondly, the control optimisation can first be attempted with reduced amplitude bounds. Bounds of $[-3, 3]$ were found to be sufficiently tight in the 7 oscillator systems. If the target infidelity is not reached with these amplitudes bounds, after some iteration limit, then the optimisation can continue with the bounds extended. Also, an `OverflowError` can be handled, and just be considered a reason for termination of an optimisation attempt. Using these methods it was possible to optimise controls for all coupled oscillator systems attempted, which was up to 7 in length, at some total evolution time $T$. A maximum of 10 optimisation attempts were made before the target was deemed unreachable the given $T$.

The random symplectic target, used for comparison with the square root boson swap target, is created using a random symmetric Hamiltonian generator $H_{\mathrm{R}}$. First a matrix $R$ with random valued elements in the range $[-1, 1]$ is created using `numpy.random.random`. The lower triangle of this is $A_1$, for which `numpy.tril(R)` is used. The lower triangle, excluding the main diagonal, $A_2$ is returned by `numpy.tril(R, -1)`. Combining these as

$$H_{\mathrm{R}} = A_1 + {A_2}^{\mathsf{T}}\,, \tag{2.1}$$

gives the Hamiltonian generator matrix. This is then used to give the random symplectic target

$$S_{\text{target}} = \mathrm{e}^{\Omega H_{\mathrm{R}}} \,. \tag{2.2}$$

### 2.6.2 Singular value decomposition – single-mode

The bounds of the reachable set of an unstable single-mode system were the primary focus in Section 3.5. The reachable set was described in terms of the parameters of the decomposition given by Eq. (3.31). The decomposition for non-passives is the product of a rotation $R_\theta$, a squeezing $Z$ and another rotation $R_\theta$. For passives $Z$ acts as $\mathbb{I}$ and hence the transformation is a single rotation. Hence, any single-mode symplectic can be described by three parameters: the angle of the first rotation $\theta$, the squeezing level $z$, and the second rotation angle $\phi$.

The reachable operations Fig. 3.1 are shown in the three-dimensional space of these singular value decomposition (SVD) parameters. The boundary was most clearly illustrated by using control optimisation to determine if specific points in the space are reachable. For these points $\theta$, $z$ and $\phi$ are clearly known values, as they were used to compose the target. It is also clear that the symplectic transformation evolved through some control pulse is reachable. To gain some insight into where to look for the reachable points, a sample of points reached through random control pulses was produced. In order to plot these points in the space, the resulting transformation must be decomposed to extract the SVD parameters.

The SciPy library provides the function `scipy.linalg.svd` which gives the decomposition of $A$ as $U\vec{s}V$, where $U, V$ are unitary and $\vec{s}$ is a vector of the singular values. Hence, with $D = \operatorname{diag}\vec{s}$, $A = UDV$. The SVD is not unique, and the SciPy function acting on a symplectic will provide $U, V$ as orthogonal, but not necessarily rotations. It also returns the singular values in descending order, which is the opposite of that required to match Eq. (3.31). The SciPy function returns either

$$S = O_\alpha Z^{-1} O_\beta \tag{2.3}$$

or

$$S = R_\alpha Z^{-1} R_\beta \,, \tag{2.4}$$

where the $O$ are orthogonal, but not rotations, and the $R$ are rotations. The form of the returned parameters can be checked, as $\mathrm{Det}\, R = 1$ and $\mathrm{Det}\, O = -1$. Using $X = \sigma_x$ on the $2 \times 2$ matrices, which switches rows when applied from the left and columns when applied from the right, and consequently switches the elements for a diagonal matrix, noting also that $X^2 = \mathbb{I}$, the SVD can transformed through the identity

$$\begin{aligned} A &= U_1 D_1 V_1 \\ &= U_1 X X D_1 X X V_1 \\ &= U_2 X D_1 X V_2 \\ &= U_2 D_2 V_2 \,. \end{aligned} \tag{2.5}$$

The transformation applied to Eq. (2.3) or Eq. (2.4) yields

$$S = R_\theta Z R_\phi \tag{2.6}$$

or

$$S = O_\theta Z O_\phi \,, \tag{2.7}$$

respectively. So for either form of the return value there are rotation matrices from which the angle can be extracted. In the case where the parameters are returned as Eq. (2.4), the rotation angles $\theta$ and $\phi$ can be found using the identity

$$S = R_\alpha Z^{-1} R_\beta = R_\theta Z R_\phi \,, \tag{2.8}$$

with $\theta = \alpha + \pi/2, \phi = \beta - \pi/2$.

NumPy provides a function `numpy.arctan2` which gives the unique angle in the range $(-\pi, \pi)$ based on its sine and cosine. Hence when a matrix is known to be in the form Eq. (3.29), the rotation can angle can be found this way. The squeezing parameter $z$ is easily extracted from the singular values $\vec{s}$. Using this method it was possible to extract the SVD parameters for all single-mode symplectics encountered.

### 2.6.3 Singular value decomposition – two-mode

The SVD of the two-mode symplectic was used extensively in the numerical study of Section 3.6. The `scipy.linalg.svd` can be used to decompose the two-mode symplectic as $S = UZV$, but as in the single-mode case discussed above, the singular values are returned in descending order and hence the component matrices need to be permuted so that $Z$ is in the format of 3.59. The $U, V$ matrices will be orthogonal, but not necessarily rotational, either before or after the permutation. Some combination of $U$ row and $V$ column sign switching will typically find a combination of $U, V$ where both are symplectic, which can be checked by $U^{\mathsf{T}}\Omega U = \Omega$. Note that the row / column sign switching was found not to work if $z_1 = z_2$, as the $U, V$ matrices returned from the SciPy could not be manipulated in this way to be symplectic. However, the $z_1 = z_2$ would likely only occur in some manufactured scenario, and is most unlikely for some randomly evolved $S$. Clearly it is trivial to extract the squeezing parameters $z_1, z_2$ from this decomposition. However, it is not immediately apparent how to extract the component angles of the two-mode rotation matrix Eq. (3.60).

If the matrices $U, V$ are symplectic and orthogonal, then this makes them rotation matrices. So, working on the assumption then that $U = R_1, V = R_2$ of Eq. (3.58), some attempt can be made to extract the $\alpha, \beta, \theta, \phi$ for each of them. Multiplying out Eq. (3.60) leads to

$$\begin{bmatrix} \cos\left(\frac{\theta}{2}\right)\cos\left(-\alpha+\frac{\beta}{2}+\frac{\phi}{2}\right) & -\sin\left(-\alpha+\frac{\beta}{2}+\frac{\phi}{2}\right)\cos\left(\frac{\theta}{2}\right) & \sin\left(\frac{\theta}{2}\right)\cos\left(\alpha-\frac{\beta}{2}+\frac{\phi}{2}\right) & \sin\left(\frac{\theta}{2}\right)\sin\left(\alpha-\frac{\beta}{2}+\frac{\phi}{2}\right) \\ \sin\left(-\alpha+\frac{\beta}{2}+\frac{\phi}{2}\right)\cos\left(\frac{\theta}{2}\right) & \cos\left(\frac{\theta}{2}\right)\cos\left(-\alpha+\frac{\beta}{2}+\frac{\phi}{2}\right) & -\sin\left(\frac{\theta}{2}\right)\sin\left(\alpha-\frac{\beta}{2}+\frac{\phi}{2}\right) & \sin\left(\frac{\theta}{2}\right)\cos\left(\alpha-\frac{\beta}{2}+\frac{\phi}{2}\right) \\ -\sin\left(\frac{\theta}{2}\right)\cos\left(\alpha+\frac{\beta}{2}-\frac{\phi}{2}\right) & -\sin\left(\frac{\theta}{2}\right)\sin\left(\alpha+\frac{\beta}{2}-\frac{\phi}{2}\right) & \cos\left(\frac{\theta}{2}\right)\cos\left(\alpha+\frac{\beta}{2}+\frac{\phi}{2}\right) & \sin\left(\alpha+\frac{\beta}{2}+\frac{\phi}{2}\right)\cos\left(\frac{\theta}{2}\right) \\ \sin\left(\frac{\theta}{2}\right)\sin\left(\alpha+\frac{\beta}{2}-\frac{\phi}{2}\right) & -\sin\left(\frac{\theta}{2}\right)\cos\left(\alpha+\frac{\beta}{2}-\frac{\phi}{2}\right) & -\sin\left(\alpha+\frac{\beta}{2}+\frac{\phi}{2}\right)\cos\left(\frac{\theta}{2}\right) & \cos\left(\frac{\theta}{2}\right)\cos\left(\alpha+\frac{\beta}{2}+\frac{\phi}{2}\right) \end{bmatrix}. \tag{2.9}$$

From this 8 unique equations can easily be identified:

$$R_{11} = \cos\left(\frac{\theta}{2}\right)\cos\left(-\alpha + \frac{\beta}{2} + \frac{\phi}{2}\right) \tag{2.10a}$$

$$R_{21} = \sin\left(-\alpha + \frac{\beta}{2} + \frac{\phi}{2}\right)\cos\left(\frac{\theta}{2}\right) \tag{2.10b}$$

$$R_{14} = \sin\left(\frac{\theta}{2}\right)\sin\left(\alpha - \frac{\beta}{2} + \frac{\phi}{2}\right) \tag{2.10c}$$

$$R_{24} = \sin\left(\frac{\theta}{2}\right)\cos\left(\alpha - \frac{\beta}{2} + \frac{\phi}{2}\right) \tag{2.10d}$$

$$R_{31} = -\sin\left(\frac{\theta}{2}\right)\cos\left(\alpha + \frac{\beta}{2} - \frac{\phi}{2}\right) \tag{2.10e}$$

$$R_{41} = \sin\left(\frac{\theta}{2}\right)\sin\left(\alpha + \frac{\beta}{2} - \frac{\phi}{2}\right) \tag{2.10f}$$

$$R_{34} = \sin\left(\alpha + \frac{\beta}{2} + \frac{\phi}{2}\right)\cos\left(\frac{\theta}{2}\right) \tag{2.10g}$$

$$R_{44} = \cos\left(\frac{\theta}{2}\right)\cos\left(\alpha + \frac{\beta}{2} + \frac{\phi}{2}\right). \tag{2.10h}$$

It should be possible to simplify these down to four equations, however, it is not immediately apparent how to do this. Also not apparent is an analytical method to find solutions to these equations – attempts with Mathematica and SymPy were not successful. Neither was an attempt to numerically solve using SymPy.

It is possible to solve for the angles in these equations using a gradient based, steepest descent method. Defining a cost function that is the sum of the residuals of these equations, and corresponding gradient functions with respect to each angle, allows efficient minimisation of the cost using the L-BFGS-B algorithm. Typically 1 to 20 random initial angle sets are needed before a solution is found, within a cost target of $10^{-3}$.

Using the method described above, it was possible to extract the parameters $\alpha_1, \beta_1, \theta_1, \phi_1$, $z_1, z_2$, $\alpha_2, \beta_2, \theta_2, \phi_2$ for all randomly evolved two-mode symplectic transformations. Eqs. (2.10) were differentiated using `sympy.diff`, and then `sympy.lambdify` was used to generate Python functions from the symbolic representation. The Python functions execute far more efficiently than passing values to the SymPy symbolic representation. This made it possible to analyse the distribution of parameters for a large sample. The two-mode symplectic SVD also made it possible to investigate the potential to reach the so-called 'shifted targets', which are described in Section 3.6.3.

## 2.7 *In situ* optimisation numerics

### 2.7.1 Determining control pulse parameters

There are clear advantages to using the shortest possible gate time $T$ and minimum possible number of control pulse timeslots $N_{\text{ts}}$. In terms of computational effort in classical simulation, the gate time $T$ is not a factor. However, for a physical realisation, this is of paramount importance for minimising decoherence. In both the classical simulation and *in situ* optimisation $N_{\text{ts}}$ is a direct scaling factor. Consequently, a significant amount of computational and personal effort was expended in finding satisfactory values for specific topologies, interaction types and system sizes.

The method used was to trial optimisations for a range of $N_{\text{ts}}$ and $T$, and find the proportion of successful operations $p_{\text{succ}}$. This task was completed using the standard gate fidelity, as optimisations using this are much faster. Parameters found to be successful were then verified using the local gate fidelity, and invariably $p_{\text{succ}}$ was equivalent with both fidelity measures. For the smaller systems it was possible to compare the full grid of $N_{\text{ts}}$ and $T$ values. For the larger systems ($n \geq 5$), only $N_{\text{ts}}$ ranges with specific $T$ and *vice-versa* was possible. Using this method, it was found that $N_{\text{ts}} = 12$ and $T = \pi$ resulted in $p_{\text{succ}} \lesssim 1$ for Ising chains of all lengths, but $T$ scaled exponentially with system size for Heisenberg chains. These parameter sizing investigations used nearest-neighbour gate qubits. It was later found that $N_{\text{ts}} = 48$ and $T = 4\pi$ were sufficient for Ising chains and rings with separated gate qubits, but still no scaling with system size.

A further investigation was made into the optimisation parameters for Heisenberg interacting system by visualising the 'infidelity landscape'. The landscape is defined by the minimum infidelity found through optimisation with the $N_{\text{ts}}$ and $T$ parameter values. The results for the 4 qubit Heisenberg chain are shown in Fig. 2.4. The plot with the broad range of parameters shows an interesting feature at $T \approx 3$. For this reason the higher resolution plot was made, based on additional optimisation results. The higher resolution plot clearly shows a region $T \approx \pi$, $N_{\text{ts}} \gtrsim 10$ where the infidelity $\mathcal{E} \approx 0$. This gave hope for finding a similar region for all lengths of Heisenberg chain,

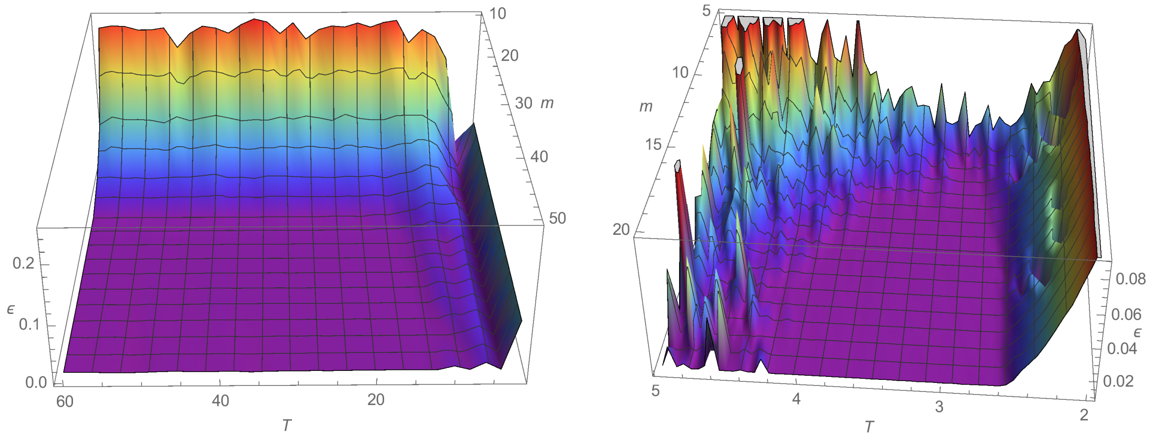


**Figure 2.4: Surface plots of the infidelity landscape for the 4 qubit Heisenberg chain.** The axes are the number of timeslots $N_{\text{ts}}$ (labelled $m$) and allowed gate time $T$. The height of the surface is the minimum infidelity found for these parameters in 20 optimisation attempts. The plot on the left is based on a broader range of the parameters: $2 \leq T \leq 60, \Delta T = 2$ and $2 \leq N_{\text{ts}} \leq 60, \Delta N_{\text{ts}} = 5$. The plot on the right focuses on a smaller range, but with a higher resolution: $2 \leq T \leq 5, \Delta T = 0.05$ and $2 \leq N_{\text{ts}} \leq 20, \Delta N_{\text{ts}} = 1$.

but unfortunately the $\mathcal{E} \approx 0$ region for short $T$ was not found with larger systems.

### 2.7.2 Numerical accuracy threshold search

A key part of the investigation into the scalability of the *in situ* optimisation of quantum gates was the scaling of the required measurement precision, which is presented in Section 4.6.3. The main relationship of interest is with system size, but a relationship with the target fidelity is also conjectured. This required a lot of different scenarios to be considered: 25 in total, 5 different fidelities and 5 system sizes. For each of these a threshold for the precision $A_{\text{num}}$ was sought that would result in a proportion $p_{\text{succ}} = 0.5$ of successful optimisations. To gain a statistically reliable value for $p_{\text{succ}}$ would take many optimisation attempts, and for the larger systems each optimisation takes a lot of computing resource. The manual process of trialling different $A_{\text{num}}$ to get closer to $p_{\text{succ}} = 0.5$ was time consuming and prone to human error, so the process was partially automated.

The aim of the automated part of the method is to produce sufficient $p_{\text{succ}}$ for given $A_{\text{num}}$, such that an interpolation can be made to determine the value of $A_{\text{num}}$ that gives $p_{\text{succ}} = 0.5$, and the uncertainly in the result. The values of $A_{\text{num}}$ that give $p_{\text{succ}} \simeq 0$ and $p_{\text{succ}} \simeq 1$ are of no use in the interpolation, and so the method seeks to perform the repeated optimisations where $p_{\text{succ}}$ will be within a useful range $[p_{\text{lb}}, p_{\text{ub}}]$, for which $[0.1, 0.95]$ was found to work well. For each $A_{\text{num}}$ scenario a number of repeated optimisations are performed from which $p_{\text{succ}}$ is calculated.

As input parameters, the method takes initial estimates for the $A_{\text{num}}$ values that correspond to these $p_{\text{succ}}$ range bounds. The method first looks to find $A_{\text{num}}$ values that give $p_{\text{succ}}$ just inside the $[p_{\text{lb}}, p_{\text{ub}}]$ bounds. $A_{\text{lb}}$ is the highest $A_{\text{num}}$ for which $p_{\text{succ}} < p_{\text{ub}}$, and $A_{\text{ub}}$ is the lowest $A_{\text{num}}$ for which $p_{\text{succ}} > p_{\text{lb}}$. Scenarios are attempted to locate these boundaries until there are $p_{\text{succ}}$ results either side of the bound. If $p_{\text{succ}} < p_{\text{lb}}$ for $A_{\text{lb}}$, then it is halved until there are $p_{\text{succ}}$ results either side of $p_{\text{lb}}$. If $p_{\text{succ}} > p_{\text{ub}}$ for $A_{\text{ub}}$, then it is doubled until there are $p_{\text{succ}}$ results either side of $p_{\text{ub}}$.

When the boundaries have been located, the method looks to find $p_{\text{succ}}$ values for $A_{\text{num}}$ evenly distributed between the bounds. For this stage all scenarios where $A_{\text{num}} < A_{\text{lb}}$, except the one closest to the bound, are excluded. Similarly, all scenarios where $A_{\text{num}} > A_{\text{ub}}$, except the one closest to the bound, are excluded. New scenarios are then tried for $A_{\text{num}}$ where the difference between previously tried scenarios is greatest. Note, the $A_{\text{num}}$ boundaries may be extended during this stage of the method as well. This stage continues until a predetermined maximum number of scenarios have been tried, which completes the automated part of the method.

The method outlined above is illustrated in Fig. 2.5. This shows a simplified version of the algorithm, shown as would be processed by a single thread. It is possible to complete some of the steps in parallel, as described below. Further details on the steps from Fig. 2.5 are given here.

- boundary search steps
  - estimate $A_{\text{lb}}$ and $A_{\text{ub}}$: these are best estimates based on observances from tests for the specific infidelity target.
  - calc $p_{\text{succ}}$: Repeated GRAPE control pulse optimisation with randomised initial pulses, using $A_{\text{num}}$ for the numerical precision. $p_{\text{succ}}$ is the proportion for which the optimisation is successful.
- gap filling steps
  - max attempts?: a maximum number of $A_{\text{num}}$ samples is set. 50 were found to be sufficient for the interpolation method.

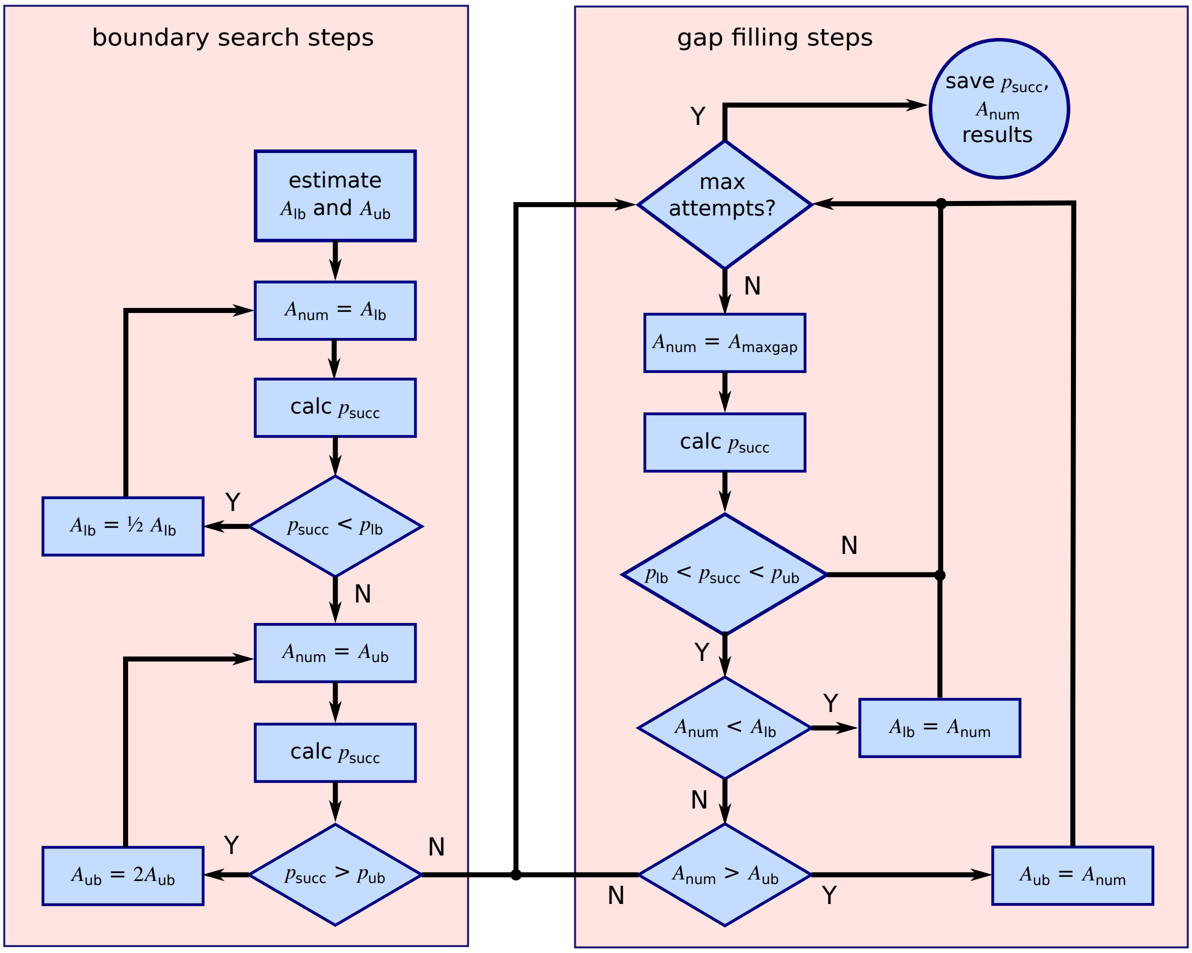

boundary search steps
estimate $A_{\text{lb}}$ and $A_{\text{ub}}$
$A_{\text{num}} = A_{\text{lb}}$
calc $p_{\text{succ}}$
$p_{\text{succ}} < p_{\text{lb}}$
Y
$A_{\text{lb}}$ = ½ $A_{\text{lb}}$
N
$A_{\text{num}} = A_{\text{ub}}$
calc $p_{\text{succ}}$
$p_{\text{succ}} > p_{\text{ub}}$
Y
$A_{\text{ub}} = 2A_{\text{ub}}$
N
gap filling steps
Y
save $p_{\text{succ}}$, $A_{\text{num}}$ results
max attempts?
N
$A_{\text{num}} = A_{\text{maxgap}}$
calc $p_{\text{succ}}$
$p_{\text{lb}} < p_{\text{succ}} < p_{\text{ub}}$
N
Y
$A_{\text{num}} < A_{\text{lb}}$
Y
$A_{\text{lb}} = A_{\text{num}}$
N
$A_{\text{num}} > A_{\text{ub}}$
N
Y
$A_{\text{ub}} = A_{\text{num}}$

**Figure 2.5:** Flow diagram of numerical accuracy threshold search.

- $A_{\text{num}} = A_{\text{maxgap}}$: If $\mathfrak{A}_{\text{all}}$ is the set of all attempted $A_{\text{num}}$, with elements in ascending order labelled $A_i, i \in \{1, 2, \ldots\}$, and $A_{\text{lb}} = A_N$ and $A_{\text{ub}} = A_M$, then the *active* set is $\mathfrak{A}_{\text{active}} = \{A_{N-1}, \ldots, A_{M+1}\}$. $A_C = A_j$ for which $\max(A_{j+1} - A_j) \forall A_j \in \mathfrak{A}_{\text{active}}$, then $A_{\text{maxgap}} = (A_C + A_{C+1})/2$. That is, $A_{\text{maxgap}}$ is the midpoint of the largest gap between those $A_{\text{num}}$ considered active.

There are clear opportunities for parallel processing in this method. Most obvious is the repeated optimisations for each $A_{\text{num}}$ scenario. For the results presented in Fig. 4.6 the number of repeats was between 50 and 200, depending on the system size. It is also possible to run multiple scenarios in parallel. In the boundary search stage it only makes sense to run two scenarios in parallel, as there are only two boundaries. In the gap-filling stage any number of scenarios could be run in parallel. As the $A_{\text{num}}$ value is determined by the method, then the multiprocessing must be managed dynamically – that is the main process must spawn sub-processes to run the scenario optimisations with the given $A_{\text{num}}$ value. The Python `multiprocessing` module is ideal for this purpose. A process `Pool` can be created that calls a Python function with specific parameter values for each process.

The interpolation method, an example of which is shown in Fig. 2.6, is based on the assumption that the relationship between $A_{\text{num}}$ and $p_{\text{succ}}$ is approximately linear around the $p_{\text{succ}}$ threshold. The region of linearity is not necessarily $[A_{\text{lb}}, A_{\text{ub}}]$ that was found during the $p_{\text{succ}}$ threshold search. The bounds of the linear region are set manually – an interactive plot is used, for which there is a mode in Matplotlib. The bounds are chosen, leaving a some of the $A_{\text{num}}, p_{\text{succ}}$ scenario results included in the interpolation, while those outside the region are excluded. The linear fit of $A_{\text{num}}$ to $mp_{\text{succ}} + c$ is performed using the `numpy.polyfit` function with order one. Along with the gradient $m$ and $x$-intercept $c$, this also gives the variances in the same, ${\sigma_m}^2, {\sigma_c}^2$. Based on these, the threshold precision $A_{\text{thresh}}$ at $p_{\text{thresh}} = 0.5$ is calculated as $A_{\text{thresh}} = mp_{\text{thresh}} + c$, and the uncertainty as $\epsilon_A = \sqrt{(\sigma_m p_{\text{thresh}})^2 + {\sigma_c}^2}$.

The number of iterations $N_{\text{iter}}$ required for a control optimisation with $A_{\text{num}} = A_{\text{thresh}}$ is estimated, using the same Numpy function, by fitting $N_{\text{iter}} = mA_{\text{num}} + c$. The number of iterations at the threshold is then $N_{\text{thresh}} = mA_{\text{thresh}} + c$, with uncertainty $\epsilon_N = \sqrt{(\sigma_m p_{\text{thresh}})^2 + {\sigma_c}^2 + (m\epsilon_A)^2}$. It is clear from Fig. 2.6 that the additional repetitions possible with the small system led less variance in both the optimisation success proportion and number of iterations.

The codes that implement the automated search method and perform the interpolation were developed by the thesis author. They are available in the public repository [122]. The function `qsorun.run_qso_sims_numer_acc_limit` performs the automated search. The the interpolation is made in the script `numer_acc_interpolate.py`, and there is a Jupyter notebook `interpolate_num_acc_thresh.ipynb` that can be used for the same .

### 2.7.3 Large system gate optimisation

Initially it was found infeasible, with the available resources, to go beyond systems with more than 8 qubits when simulating optimisation of gates. Each optimisation would take near to the 2 day limit, and it was not obvious how to parallelise this without making significant developments in QuTiP control modules. However, it was found later on in the study, after a new release of SciPy (or

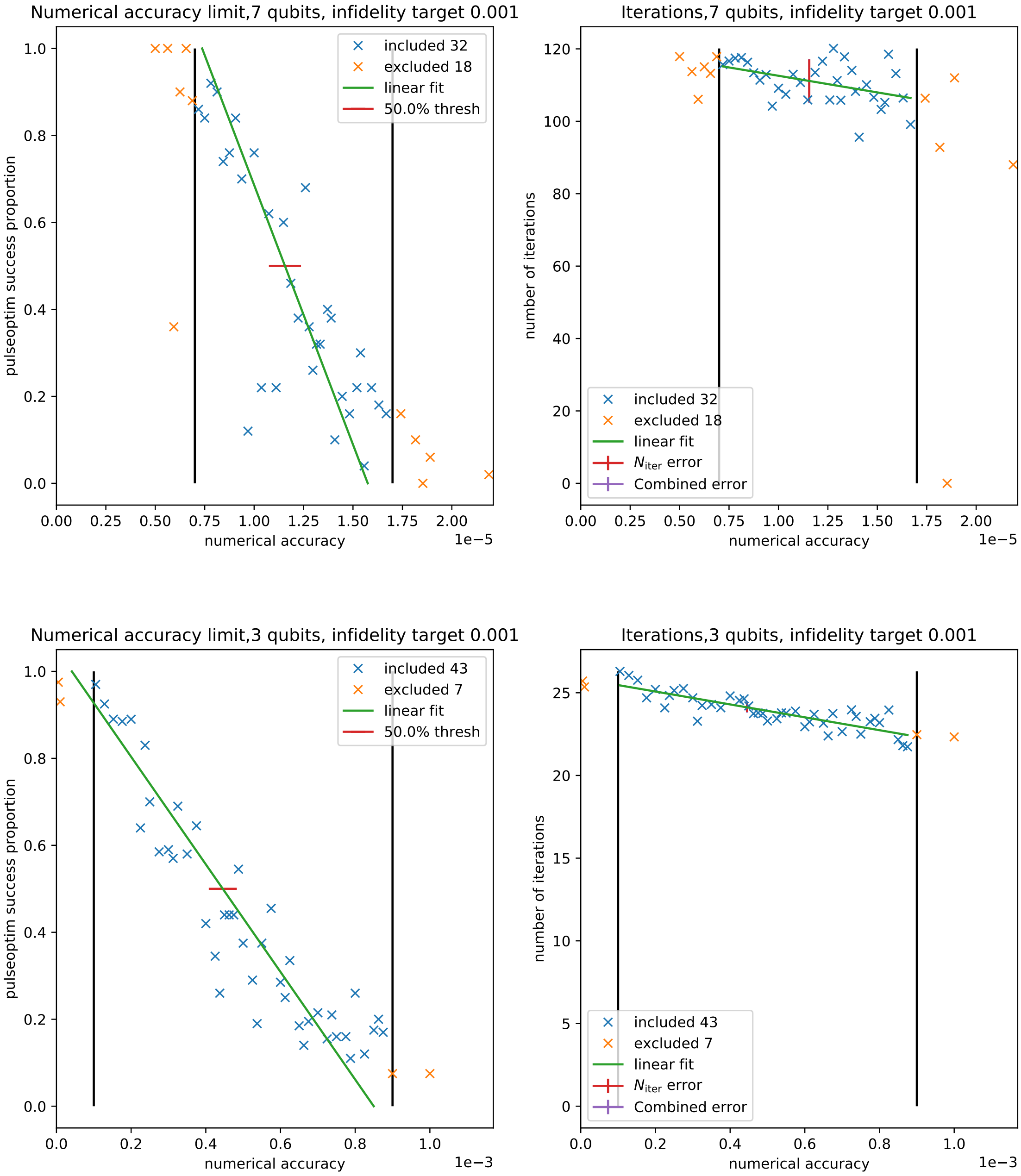


**Figure 2.6: Numerical accuracy and iteration interpolation.** Plots illustrating (left) the interpolation method used to determine the numerical accuracy required to achieve a 50% success rate for control pulse optimisation, and (right) the interpolation method used to estimate the average number of iterations required for the optimisation. The blue crosses are the points included in the interpolations. The black vertical bars indicate the boundaries used to exclude points not in the linear region. The green lines show the linear fits. The red bars show the uncertainty derived from the fit, and the purple bar in the interpolation plot shows the combined uncertainty from both fits. The upper plots are for a 7 qubit Ising chain (50 repetitions per scenario), and the lower plots a 3 qubit chain (200 repetitions per scenario).

a new version of the Intel MKL in Anaconda), that the matrix exponential `scipy.linalg.expm` and `scipy.linalg.expm_frechet` could utilise multiple cores. By allocating 4 cores to each control optimisation it was possible to bring the average run time for a 9 qubit gate optimisation (with infidelity target $10^{-3}$) to within two days. This is because, for large systems, the time-limiting step is the exponentiation of the generators to compute the propagators.

When using the SciPy library in Anaconda, it is possible to allocate multiple cores to maths functions using

```
import mkl
mkl.set_num_threads(num_threads)
```

where `num_threads` is the number of cores to be allocated. Note that only specific functions execute in parallel, and for even for supported functions, parallel processing is only used for matrices over a certain size. For instance, propagator computation was only found to use multiple cores for systems of 5 qubits or greater.

## 2.8 Open system dynamics simulation

### 2.8.1 Numerical solving of the master equation

The Lindblad master equation Eq. (5.11) or Eq. (5.13) can be solved for density operators or maps using the QuTiP function `mesolve`. This takes as parameters the initial state or map operator, along with the dynamics generator as either a Hamiltonian or a Lindbladian. The usage is very flexible, and it fully explained in the documentation [114] and with examples in [16], so only some key points will be explained here. A time-dependent Hamiltonian (or Lindbladian) can be passed (for example) as `[H0, [H1, h1_td], [H2, h2_td]]`. Where `H0, H1, H2` are `Qobj` (which is essentially a matrix), and the `h1_td, h2_td` are functions that take $t$ as a parameter and return a scaling factor. The Lindblad operators are passed as a list in the `c_ops` parameter. These are assumed to be $\sqrt{\gamma_j}V_j$. Again, any one of these can be passed in time dependent form `[V_j, g_j_td]`, where `g_j_td` is a function that returns $\sqrt{\gamma_j(t)}$.

### Control optimisation with Lindblad dynamics

To use the QuTiP control modules with Lindblad generators and density operator or maps, it is necessary to pass parameters as superoperators and vectorised density matrices. QuTiP provides functions for these in the `superoperator` module. The `liouvillian` function can be used to create a Lindbladian superoperator from a Hamiltonian and a list of Lindblad (collapse) operators, to give the $L_0$ of Eq. (5.15). The same function, without collapse operators, can be used to create the control dynamics generators $L_j$. A density matrix can be vectorised using `operator_to_vector`, or for map optimisation, the `sprepost` function will give the superoperator representation of initial and target operators. For instance `sprepost(identity(d), identity(d))` gives the identity map for a system of dimension $d$.

Using parameters of this type for `drift` and `control` dynamics generators, and `initial` and `target` states or maps, the `qutip.control.pulseoptim` function can be used to simulate control optimisation in systems with Lindbladian dynamics. The other methods described in Subsection 2.3.2 are also valid for this use. There is an example notebook 'Optimal Control, Lindbladian' in the QuTiP tutorials [120] and some example scripts in Qtrl examples repository [121]. The example script:

```
ctrl_pulse_optim-example-Lindbladian-compare_map_dm_evo.py
```

compares state and map control optimisation.

## 2.8.2 QuTiP implementation of the HEOM

The HEOM solver code is organised in an object oriented manner. The code can be found in `qutip.nonmarkov.heom`. The base class `HEOMSolver` is intended as a container for any methods common to HEOM solver types. The `HSolverDL` is based on the Drude-Lorentz spectral density description of the bath correlations. Solver objects of this `HSolverDL` class were used in this study.

The HEOM are solved in QuTiP by numerically integrating the equation

$$\frac{\mathrm{d}}{\mathrm{d}t}|\rho_{\mathsf{HE}}\rangle = \mathcal{L}_{\mathsf{HE}}\,|\rho_{\mathsf{HE}}\rangle\,. \tag{2.11}$$

where $|\rho_{\mathsf{HE}}\rangle$ is a vector made from the system and ancillary density operator elements, and $\mathcal{L}_{\mathsf{HE}}$ is a superoperator that contains all the operations described by Eq. (5.29). For system of dimension $d$, the first $d^2$ vector elements are those of the system density matrix, as in Eq. (5.9). The $d^2$ elements of each ancillary density matrix follow sequentially after, in column stacking order. These ancillary density matrix elements are all initialised as 0.

The superoperator $\mathcal{L}_{\mathsf{HE}}$ is constructed in stages. The order is unimportant, as the stages are all additive. They are described here in the order they are done in `HSolverDL.configure`. Firstly the dissipating components, which thanks to the linear form of Eq. (5.29), can be added to the $\mathcal{L}_{\mathsf{HE}}$ one at a time. Each hierarchy component is a $d^2 \times d^2$ matrix. The number and arrangement of the hierarchy components is determined by the index lists $\mathsf{n}$ of Eq. (5.29).

Due to the potentially very large number of combinations arising from these lists, as discussed in Section 5.4, limiting the number of bath states is key to minimising the number of ancillaries, and hence the size of the HEOM superoperator. This is achieved using excitation number restricted (ENR) states. For this purpose, each Matsubara term is considered a component in a multipartite system, and the bath cutoff parameter $N_c$ as the total number of excitation across all components. Using this equivalence the `qutip.states.enr_state_dictionaries` function can be used to generate list of 'states' that are used as the index lists $\mathsf{n}$. The function returns a Python dictionary of these states with an integer key and the corresponding dictionary of indexes with the states as keys. The latter dictionary allows the $\rho_{\mathsf{n}\pm}$ ancilla to be identified. This leads to $N_{\mathsf{HE}} = (N_c + K)!/(N_c!K!)$ hierarchy elements that are the system and ancillary density matrices. As an example, for $K = 3$ Matsubara terms and bath cut off $N_c = 2$, the index lists $\mathsf{n}$ would be:

$$
\begin{aligned}
0 &: (0, 0, 0)\\
1 &: (0, 0, 1)\\
2 &: (0, 0, 2)\\
3 &: (0, 1, 0)\\
4 &: (0, 1, 1)\\
5 &: (0, 2, 0)\\
6 &: (1, 0, 0)\\
7 &: (1, 0, 1)\\
8 &: (1, 1, 0)\\
9 &: (2, 0, 0)
\end{aligned}
$$

A more detailed description of ENR states, and a demonstration of their intended purpose, can be found in this notebook [125].

As each hierarchy component $\mathcal{L}_{\mathsf{C}}$ is a small matrix, that needs to be added into the full $\mathcal{L}_{\mathsf{HE}}$

matrix in a specific location, then the `qutip.cy.heom.cy_pad_csr` function is used to create a matrix of size of $\mathcal{L}_{\mathsf{HE}}$ with $\mathcal{L}_{\mathsf{C}}$ in the specified location, with zeros elsewhere. This function was developed specifically for this purpose, because it is needed so many times in the construction of the hierarchy superoperator. The Cython package [126] is used to make a compiled code component to maximise efficiency. $\mathcal{L}_{\mathsf{C}}$ is stored as a `fast_csr_matrix`, which is a QuTiP specific version of the *compressed sparse row* matrix. This stores the non-zero data items in one dense array, and row start and column indexes in others. As the data is not changed by the function, only the indexes are updated by `cy_pad_csr`.

The hierarchy components that correspond to those in the first summation in Eq. (5.29) are added to the diagonal of $\mathcal{L}_{\mathsf{HE}}$. Each diagonal component is

$$\mathcal{L}_{\mathsf{C}} = \mathcal{L}_{\mathsf{n}} = \left( \sum_{k=0}^{K} n_k \gamma_k \right) \mathbb{I}_{d^2} \tag{2.12}$$

where the $n_k$ are the excitation numbers corresponding to this element, and $\gamma_k$ is the Matsubara frequency. The interaction components, which are those in the second summation in Eq. (5.29), are added offset from the diagonal. These components, for each Matsubara term, are

$$\mathcal{L}_{\mathsf{n}_{\mathsf{k}}^{+}} = -\mathrm{i}\, n_k \mathcal{C}_k \,, \quad \mathcal{C}_k := \eta_k \mathbb{I} \otimes Q - \eta_k^{*} Q^{\mathsf{T}} \otimes \mathbb{I} \tag{2.13}$$

$$\mathcal{L}_{\mathsf{n}_{\mathsf{k}}^{-}} = -\mathrm{i}\, \mathcal{Q} \,, \quad \mathcal{Q} := \mathbb{I} \otimes Q - Q^{\mathsf{T}} \otimes \mathbb{I} \,, \tag{2.14}$$

where $\eta_k$ is the Matsubara coefficient, and $Q$ is the system-bath interaction operator. The hierarchy element index (and hence position in the matrix) for corresponding $\rho_{\mathsf{n}_{\mathsf{k}}^{\pm}}$ is found using the $\mathsf{n}_{\mathsf{k}}^{\pm}$ list as a key in the ENR state index dictionary. If there is only one Matsubara term $K = 1$, then the interactions are with nearest-neighbour ancillae only. Otherwise there may be interactions with ancillae up to and including a separation of $K$ in the hierarchy. This arrangement of the hierarchy elements is illustrated in Fig. 2.7.

As $\mathcal{L}_{\mathsf{HE}}$ and all $\mathcal{L}_{\mathsf{C}}$ are stored as `fast_csr_matrix` the addition operation is quite efficient. It was for this kind of application that Paul Nation developed the QuTiP specific version of the CSR matrix, and associated functions, including `cy_pad_csr`. SciPy has a sparse matrix type

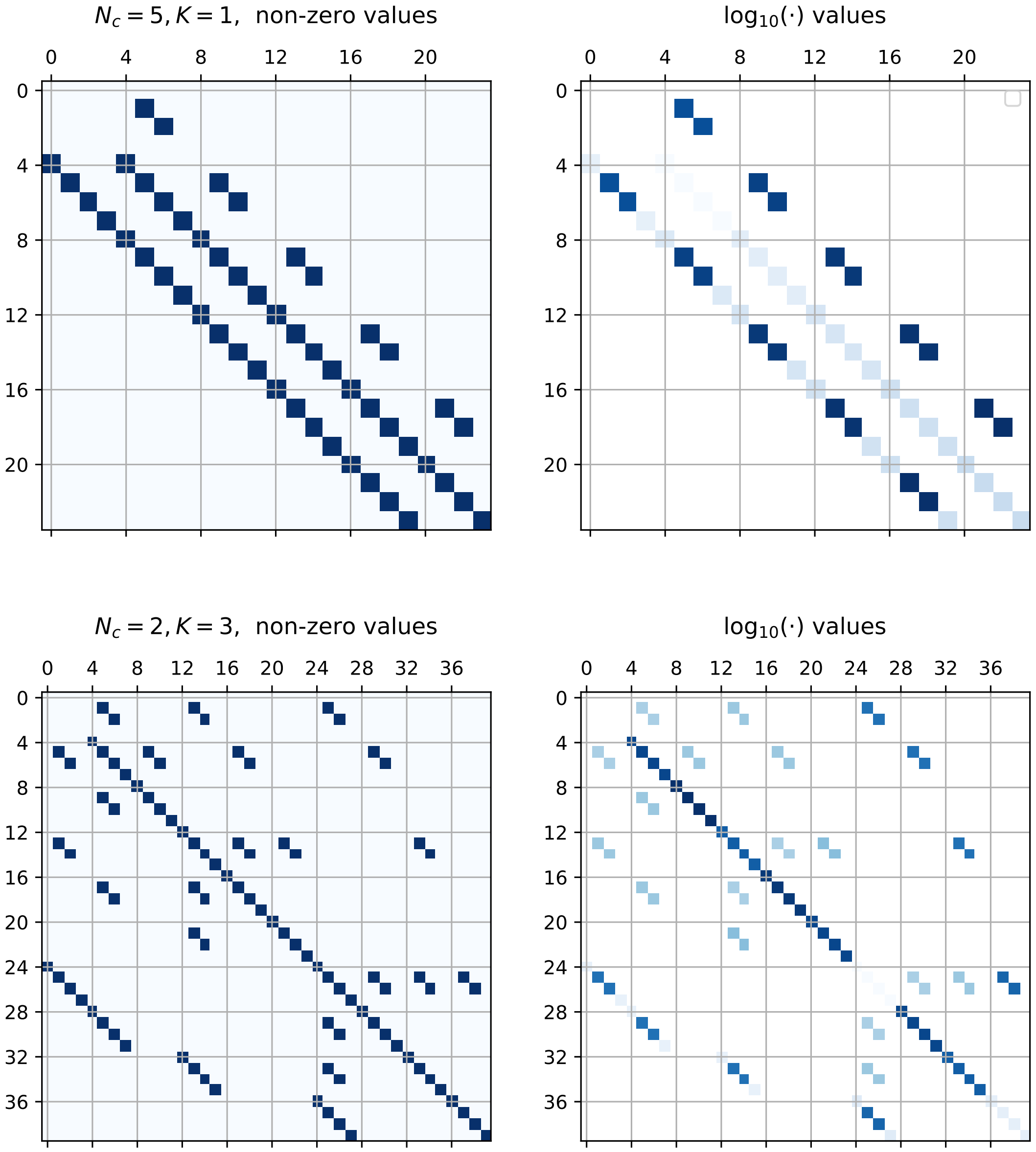


**Figure 2.7: Structure of the hierarchy.** Visualisations of the dissipative part of the HEOM superoperator $\mathcal{L}_{\mathsf{HE}}$, that is, without the system Hamiltonian components $\mathcal{L}_H$. The axes on the left show the non-zero elements of the matrix. The axes on the right give an indication of the relative absolute values of the elements, with the darkness of the blocks scaled by the $\log$ of the value. The upper axes, with $K = 1$ Matsubara terms, show the nearest-neighbour only interactions. The lower axes, where $K = 3$, show the additional interactions resulting from multiple Matsubara terms. These matrices are for a two-level system, hence the blocks of $d^2 = 4$ elements. The matrices correspond to Drude-Lorentz bath parameters $\omega_c = 0.01, \alpha = 0.5, T = 1$, see Section 5.4 for definitions.

`csr_matrix`, this does a lot of additional checking that `fast_csr_matrix` does not. Partly because `fast_csr_matrix` only supports the `numpy.complex128` data type.

The system Hamiltonian is added to all diagonal components by creating the superoperator

$$\mathcal{L}_H = \mathrm{i}\left(H^{\mathsf{T}} \otimes \mathbb{I}_d - \mathbb{I}_d \otimes H\right) \otimes \mathbb{I}_{N_{\mathsf{HE}}} \, , \tag{2.15}$$

using the QuTiP `liouvillian` and `zcsr_kron` functions. When it is added depends on whether $H$ is time-dependent. For time-independent $H$ this $\mathcal{L}_H$ is added to $\mathcal{L}_{\mathsf{HE}}$ at the end of the configuration stage, and the `cy_ode_rhs` function is set to evaluate the RHS of Eq. (2.11) for each ODE time step evaluation. This function is optimised for sparse matrix multiplication of dense vectors. For $H = H(t)$, the $\mathcal{L}_H$ is composed as

$$\mathcal{L}_H(t) = \mathcal{L}_{H_0} + \sum_{j=1}^{N_{\mathrm{td}}} u_j(t)\mathcal{L}_{H_j} \, , \tag{2.16}$$

where $\mathcal{L}_{H_0}$ is the constant component, and the $\mathcal{L}_{H_j}$ are the time-dependent components, of $\mathcal{L}_H$, the $u_j(t)$ are scalar functions, and $N_{\mathrm{td}}$ is the number of time-dependent components. $\mathcal{L}_{H_0}$ and the $\mathcal{L}_{H_j}$ are calculated from the components of the system Hamiltonian as per Eq. (2.15). The components are recombined as per Eq. (2.16) each time the RHS of Eq. (2.11) is evaluated. Hence, the HEOM solver can be used to investigate controlled dynamics with a non-Markovian noise model.

#### Optimisation of dynamical decoupling sequences

The optimisation of dynamical decoupling (DD) sequences was completed using a set of control optimisation modules currently in development stage. They are publicly available through the thesis authors fork of `qutip` [127] in the `qoc` branch. The motivation behind the development of these modules is to be able to utilise the wide variety of dynamics solvers in QuTiP, such as the HEOM solver in this case. The structure is much the same as Qtrl, except that there is no assumption of piecewise constant control functions, nor constant system dynamics generators generally. Hence they are more suitable for implementing algorithms such as CRAB and GOAT, although the main advantage will be to easily compare control optimisations using different noise models.

The optimisation of dynamical decoupling (DD) sequences is performed using the same L-BFGS-B implementation mentioned in Subsection 2.3.1. The optimisation variables in this case being the time intervals between DD pulses. It is not immediately apparent how exact infidelity gradients with respect to these variables can be calculated. Consequently, the gradients are approximated by the `scipy.optimize.fmin_l_bfgs_b` function itself. Use of approximate gradients also makes it trivial to add a cost penalty function.

It was found that the optimisation method tended to allow variable values that pushed some pulse start times beyond the evolution end time $t_f$. Once this happened in some iteration step, they did not tend to return during the optimisation process. This is likely due to the periodic form of the evolution of $|\rho_{12}(t)|$ when DD pulses are applied. It is not possible to disallow this through the bounds in the L-BFGS-B algorithm, as they can only be set for individual variables, and so a penalty function was used to avoid this happening. Taking $t_p$ to be the end time of the last pulse, the penalty is calculated as

$$f_p(t_p) = \begin{cases} 0 & t_p \leq t_f \\ w(t_p - t_f) & t_p > t_f \end{cases}, \tag{2.17}$$

where $w$ is some weighting, which was set at $w = 0.1$. The total optimisation cost then being $\mathcal{E}_{\text{pure}} + f_p$. This was found to constrain the pulse times to within the evolution time window.

# Chapter 3

# Symplectic operator control

The research in Section 3.5 was completed in collaboration with Uther Shackerley-Bennett, Marco G. Genoni, Alessio Serafini and Daniel K. Burgarth. The results were published in [128]. The research in Section 3.6 was completed in collaboration with Uther Shackerley-Bennett and Daniel K. Burgarth. The research in Section 3.7 was completed in collaboration with Daniel K. Burgarth.

Marco G. Genoni proposed the idea for using the isomorphism of $\mathrm{SU}(1,1)$ and $\mathrm{Sp}(2,\mathbb{R})$, and then using result from Ref. [12], in investigating the reachable operations of single-mode systems. The thesis author worked independently on a basis for visualisation of the single-mode reachable set. Uther Shackerley-Bennett translated the proof from Ref. [12], formalised the mathematics of the singular value decomposition and framed the proof in that basis. The thesis author developed the code for the numerical optimisation, and performed all the simulations and analysis thereof. Uther Shackerley-Bennett proposed the measure of squeezing for a $n$-mode symplectic operator and provided the example two-mode neutral and non-neutral systems. The thesis author developed the code for, and completed the numerical simulations and analysis of, the two-mode systems. The thesis author completed the numerical investigation of the minimum gate time in the oscillator chain model. Alessio Serafini and Daniel K. Burgarth supervised the research work of Uther Shackerley-Bennett and the thesis author respectively.

## 3.1 Introduction

Quantum computing with continuous variable (CV) systems was proposed by Lloyd and Braunstein [9], analogous to classical analogue computing in the same way that qubit-based quantum computers are analogous to digital classical computers. For universal quantum computing some non-linear gates are required, however a useful set gates are provided by symplectic transformations acting on the quadrature operators, as described below. The gates include the CV equivalent of the Clifford Group, meaning that error correction codes and teleportation protocols can be implemented solely using these gates [129].

Simulation of control optimisation with CV systems was first attempted by Wu et al., as reported in Ref. [14]. The authors proved that the control parameter space is smooth and free of traps. They also found that it was important to have control over interactions between system components. The symplectic group $\mathrm{Sp}(2n, \mathbb{R})$ is non-compact, meaning that the Lie algebra controllability criterion is not directly applicable. Controllability criterion have been given [10, 11, 12, 13], some of which are described later in this chapter. Previous control optimisation study has concentrated on systems where the controllability criterion is met. The reachable operations for systems where the criteria are not met are investigated herewithin.

Genoni et al. proposed a fully controllable model for a coupled oscillator based on the criterion the authors present in Ref. [13]. Investigating the limits on controllability in this model is where this study began for the thesis author. The characterisation of the reachable set of the single-mode system was embarked upon to better understand the importance of positive-definite dynamics generators. This coupled oscillator model is used to investigate scalability in optimisation of symplectic transformations, specifically, minimum times required to perform gates in CV systems of increasing size, hence addressing the thesis question in relation to this type of quantum system.

The codes used in Ref. [14] are not publicly accessible, therefore the control modules in QuTiP were extended by the thesis author to support the numerical part of this study. These are described in Chapter 2, including some of the challenges of working numerically with symplectic operator evolution discussed in Section 2.6.

This chapter is organised as follows. Section 3.2 gives some background on continuous variable systems, which are also called 'infinite dimensional', for the reason given therein. Section 3.3 gives the mathematical description of symplectic transformations, and Section 3.4 explains how control optimisation is performed with them. The reachable sets of operations on unstable single-mode systems is investigated in Section 3.5. Some attempt to extend this result to multi-mode systems is described in Section 3.6. The scalability investigation is described in Section 3.7. Some discussion of the results and outlook is given in Section 3.8

## 3.2 Infinite dimensional systems

Some quantum systems have a finite number of basis states. A prime example being those that can represent the qubit in quantum information, which is some two state system. For instance the up and down states of a spin-½ system or the energetic states of a two level atom. When the basis states are set of orthogonal *wavefunctions* that describe the probability of finding a particle at a given position $x$ there is typically an infinite number of them, each with increasingly quantised energy levels, with no theoretical ceiling. One example of this is the quantum harmonic oscillator (QHO), which will be the focus of this chapter. Other examples, for which there are analytical solutions, are the particle-in-a-box and the hydrogen atom. As there are an infinite number of basis states, then the Hilbert space in which they can be represented is infinite dimensional. Hence the term *infinite dimensional systems*.

The Hamiltonian of the particle in a QHO is

$$\hat{H} = \frac{p^2}{2m} + \frac{1}{2}m\omega^2 x^2 \tag{3.1}$$

where $m$ is the mass of the particle, $\omega$ is the angular frequency of the system, $\hat{x} = x$ is the position operator and $\hat{p} = -\mathrm{i}\frac{\partial}{\partial x}$ is the momentum operator. The solutions to the Schrödinger operator equation $\hat{H}|\psi\rangle = E|\psi\rangle$ for this are

$$\langle x|\psi\rangle = \frac{1}{\sqrt{2^n n!}} \cdot \left(\frac{m\omega}{\pi}\right)^{1/4} \cdot \mathrm{e}^{-\frac{1}{2}m\omega x^2} \cdot \mathrm{H}_n(\sqrt{m\omega}x), \quad n = 0, 1, 2, \ldots \tag{3.2}$$

where $\mathrm{H}_n$ are the Hermite polynomials

$$\mathrm{H}_n(u) = (-1)^n \,\mathrm{e}^{u^2} \frac{\mathrm{d}^n}{\mathrm{d}u^n}\left(e^{-u^2}\right) . \tag{3.3}$$

The corresponding eigenvalues are

$$E_n = \omega\left(n + \frac{1}{2}\right) \tag{3.4}$$

where $n = 0, 1, 2, \ldots$ produces the quantised energy levels and the infinite set of eigenfunctions of the Hamiltonian. These solutions are studied by all physics undergraduates and hence there derivation can be found in many text books. It should be remembered here that any superposition of quantum states is also a valid quantum state. Meaning simply that the linear combination of any of these wavefunctions, with normalised scale factors, is also a solution to the Schrödinger equation with the QHO Hamiltonian.

For this study the shape of the wavefunction is unimportant, it is sufficient that they are orthogonal. Consequently they can represented as set of orthogonal basis vectors on a Hilbert space. In the Dirac notation these are $|0\rangle, |1\rangle, \ldots, |n\rangle, \ldots$, collectively known as the Fock basis.

These operators decrease ($\hat{a}$) or increase ($\hat{a}^\dagger$) the quantised energy level

$$\hat{a} = \sqrt{\frac{m\omega}{2}}\left(\hat{x} + \frac{\mathrm{i}}{m\omega}\hat{p}\right), \quad \hat{a}^\dagger = \sqrt{\frac{m\omega}{2}}\left(\hat{x} - \frac{\mathrm{i}}{m\omega}\hat{p}\right) \tag{3.5}$$

consequently they are called the *lowering* and *raising* operators, or the *annihilation* and *creation* operators, respectively. The position and momentum operators can be expressed in terms of them

$$\hat{x} = \sqrt{\frac{1}{2m\omega}}\left(\hat{a}^\dagger + \hat{a}\right), \quad \hat{p} = \mathrm{i}\sqrt{\frac{1}{2m\omega}}\left(\hat{a}^\dagger - \hat{a}\right) \tag{3.6}$$

Commutation relations can be derived from $[\hat{x}, \hat{p}] = \mathrm{i}$

$$[\hat{a}, \hat{a}^\dagger] = 1 \,. \tag{3.7}$$

Combined these give the Hamiltonian in this form

$$\hat{H} = \omega\left(\hat{a}^\dagger\hat{a} + \frac{1}{2}\right) \tag{3.8}$$

The number operator $\hat{N} = \hat{a}^\dagger\hat{a}$ is the energy operator of the system in terms of the number of quanta, *i.e.* $\hat{N}|n\rangle = n|n\rangle$ where $|n\rangle$ is the energy eigenstate of the system.

In the quantum field theory used in quantum optics the annihilation and creation operators remove or add a photon to the field. The quadrature operators, defined as

$$\hat{q} = \frac{1}{\sqrt{2}}\big(\hat{a}^\dagger + \hat{a}\big), \quad \hat{p} = \frac{\mathrm{i}}{\sqrt{2}}\big(\hat{a}^\dagger - \hat{a}\big) \tag{3.9}$$

where $\hat{q}, \hat{p}$ are the dimensionless equivalent of $\hat{x}, \hat{p}$ in Eq. (3.6), and are the real and imaginary components of the field amplitude respectively. The Hamiltonian of the field can be expressed as

$$\hat{H} = \omega\big(\hat{q}^2 + \hat{p}^2\big) \tag{3.10}$$

The number operator $\hat{N}$ relates to the number of photons in the field.

Due to the similarity of form $\hat{q}$ is referred to as position-like and $\hat{p}$ as momentum-like. The Hamiltonians Eq. (3.8) and Eq. (3.10) are equivalent. Hence QHO systems and quantum fields can be studied as equivalent systems.

Although care must be taken, there are methods to work analytically with infinite dimensional systems. Numerically, however, it is impossible to perform computations that include the entire Fock basis, as this would require infinite size vectors and matrix operators. When working numerically with the Fock basis a cutoff point must be chosen, making an assumption that the oscillators will never reach energies beyond the cutoff. It is typically considered a safe assumption, taking into account a certain temperature ceiling. A model using such a cutoff is described in Chapter 5.

It can be seen that the ground state wavefunction $n = 0$ (also known as the *vacuum state*) is Gaussian in the position operator $\hat{x}$, as the Hermite polynomial for $n = 0$ is $\mathrm{H}_0 = 1$, and hence Eq. (3.6) is Gaussian in $\hat{x}$ for $n = 0$. It can also be shown that it is Gaussian in the momentum operator $\hat{p}$. There are other state classes, notably the *thermal states* and the *coherent states*, that are also Gaussian. These three are an important set of states for QHO systems, and important herein because they are preserved during symplectic transformation [130].

In the Gaussian states the quadrature operators of a single mode can be completely described by two parameters, their mean and variance. As these can be calculated from quadratic components only, Hamiltonians that are no more than quadratic in these operators preserve the Gaussian nature of these states. For this reason, these quadratic systems, although infinite dimensional, can be treated exactly analytically and numerically.

This Gaussian formalism has been used widely as a research tool as it can describe many classes of system – quantised light fields [131], the motional degrees of freedom of trapped ions [132], opto- and nano-mechanical oscillators [133] and superconducting Josephson junctions [134]. These systems offer the components of emerging quantum technologies, such as entanglement generation, squeezing, cooling and quantum communication protocols [130, 135].

## 3.3 Symplectic transformations

For $n$ mode systems with quadratic Hamiltonians, evolution in the Heisenberg picture can be constructed using a vector of the quadrature operators

$$\vec{\mathbf{r}} = (\hat{q}_1, \hat{p}_1, \ldots, \hat{q}_n, \hat{p}_n)^\mathsf{T} \,. \tag{3.11}$$

As the subscripts refer to separate modes, then the quadrature operators obey these commutation relations $[\hat{r}_l, \hat{r}_m] = \mathrm{i}\delta_{lm}$. The Hamiltonian can then be defined as

$$\hat{H} = \frac{1}{2}\sum_{l,m} H_{lm}\hat{r}_l\hat{r}_m \tag{3.12}$$

It can be seen that here $\hat{H}$ will be at most quadratic in the quadrature operators. The $2n \times 2n$ matrix $H$ is real and symmetric, which ensures the bilinearity of $\hat{H}$. The hat notation is retained for $\hat{H}$, the Hamiltonian operator to distinguish it from $H$, the Hamiltonian generating matrix.

The real symplectic non-compact Lie group $\mathrm{Sp}(2n, \mathbb{R})$ is the set of $2n \times 2n$ matrices $S$ such that

$$S\Omega S^{\intercal} = \Omega, \quad \text{where} \quad \Omega := \bigoplus_{i=1}^{n} \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix}. \tag{3.13}$$

All symplectic matrices have determinant $1$. Symplectic matrices resulting from the Hamiltonian generating matrices, as described below, have only real elements, and hence this chapter is only concerned with real symplectic matrices. The group has dimension $n(2n+1)$. The exponential map from the Lie algebra $\mathfrak{sp}(2n, \mathbb{R})$ to the group $\mathrm{Sp}(2n, \mathbb{R})$ is not surjective. However, all elements in the group can be generated by multiplication of two exponentiated algebra elements. That is $\forall S \in \mathrm{Sp}(2n, \mathbb{R})$ there exists $A, B \in \mathfrak{sp}(2n, \mathbb{R})$ such that $S = \mathrm{e}^{A}\mathrm{e}^{B}$. A set of symplectic matrices are *symplectically similar* to another if one symplectic matrix maps one set to the other.

For a unitary evolution operator $\hat{U} = \mathrm{e}^{\mathrm{i}\hat{H}t}$, defined on the infinite-dimensional space of the quadrature operators in $\vec{\mathbf{r}}$, the Heisenberg evolution is given by $U^{\dagger}\vec{\mathbf{r}}U = S\vec{\mathbf{r}}$ where $S$ is a symplectic matrix [136, 137]. For real, symmetric $2n \times 2n$ matrices $H$, matrices $-H\Omega$ are in the Lie algebra $\mathfrak{sp}(2, \mathbb{R})$ of the symplectic group $\mathrm{Sp}(2n, \mathbb{R})$. The quantum system will then evolve over time according to $S = \mathrm{e}^{-H\Omega t}$. This correspondence between $S$ and $\hat{U}$ means that the Schrödinger operator equation Eq. (1.6) for these systems can be written as

$$\dot{S} = XS\,, \tag{3.14}$$

where $X = -H\Omega$ and can be time-dependent. The controlled dynamics operator equation, taking the form $X = A + u(t)B$, is hence

$$\dot{S}(t) = \left(A + \sum_{j} u_j(t) B_j\right) S(t), \quad S(0) = \mathbb{I}. \tag{3.15}$$

This is the equivalent to the general closed system control dynamics equation Eq. (1.23). Here the control functions $u_j(t)$ are real valued functions defined for all positive times $t$ and $A, B \in \mathfrak{sp}(2n, \mathbb{R})$.

## 3.4 Symplectic operator control optimisation

The GRAPE algorithm (see Subsection 1.6.4) was used to produce some of the results in this chapter. Some of the principles are recast here in the context of symplectic operators.

The evolution in each time slice is given by

$$S_k = e^{-H_k \Omega \Delta t}. \tag{3.16}$$

The full evolution is given by

$$S(T) = S_{N_t} S_{N_t - 1} \cdots S_k \cdots S_2 S_1. \tag{3.17}$$

The cost function for the optimisation is the 'trace difference' infidelity defined in Eq. (1.48). In terms of symplectic operators it is

$$\mathcal{E} := \lambda \operatorname{Tr}[(S(T) - S_{target})^{\mathsf{T}} (S(T) - S_{target})] \,, \tag{3.18}$$

with $\lambda = 1/(4n)$ for a $2n \times 2n$ matrix. It is clear that $\mathcal{E} = 0$ if $S(T) = S_{\text{target}}$. However, the units are difficult to quantify, as there is no upper bound due to the non-compact nature of the symplectic group. Some of the consequences of this are discussed in Section 2.6. Exact gradients can be calculated for the symplectic propagators, and hence the fidelity error, using the Fréchet derivative method. This is described for general generator matrices in Section 1.6).

Note Section 3.5 takes the propagator to be $S = e^{\Omega H \Delta t}$. Using $\Omega^{-1} = -\Omega$, hence $\Omega^2 = -\mathbb{I}$

$$(-\Omega)(-H\Omega)\Omega = \Omega H \Omega^2 \tag{3.19}$$

$$= -\Omega H \,, \tag{3.20}$$

therefore

$$\Omega^{-1} e^{-H\Omega\Delta t} \Omega = e^{-\Omega H \Delta t} \,. \tag{3.21}$$

As $\Omega^{-1}(\cdot)\Omega$ is a symplectic similarity transform, then the results are equivalent with $t = -t$.

## 3.5 The reachable set of single-mode quadratic Hamiltonians

In this section single-mode systems will be used to investigate the controllability of systems with quadratic Hamiltonians. The controlled dynamics operator equation Eq. (3.15) for a single mode with one control operator is

$$\dot{S}(t) = (A + u(t)B)\, S(t), \quad S(0) = \mathbb{I}_2, \tag{3.22}$$

where the control function $u(t)$ is a real valued function defined for all positive times $t$ and $A, B \in \mathfrak{sp}(2, \mathbb{R})$.

### 3.5.1 Controllability and the reachable set

In finite dimensional systems the Lie algebra rank criterion (LARC) is a necessary and sufficient condition to determine the controllability of a quantum system, see Section 1.4. Satisfying the criterion shows that all directions in the Hilbert space can be followed and therefore, given sufficient time, the entire space is reachable due to its bounded nature and dynamical recurrence. For infinite dimensional systems LARC is also a necessary condition. Again it shows that all directions are accessible, but as the group is non-compact, then dynamical recurrence cannot be assured. In quantum optics terms, there is the possibility that despite all phase-rotation directions being possible squeezing will dominate the dynamics. These systems are referred to here as 'unstable'.

It has been shown that for systems with quadratic Hamiltonians that if the combined Hamiltonian matrix of $H = H_A + \sum_j u_j(t) H_{B_j}$ is positive(negative)-definite, then this combined with LARC is sufficient for controllability [13]. This property of $H$ corresponds to *neutrality* of the symplectic generator. A neutral matrix has all imaginary eigenvalues, which in the context of dynamics generators, leads to recurrent motion. The existence of combined control Hamiltonians that provide recurrence was shown to be a sufficient condition generally for non-compact Lie groups by Jurdjevic and Sussmann [10].

It has been shown that neutrality is a sufficient and necessary condition for controllability in single-mode systems by Wu, Li, Zhang and Tarn [12]. They consider systems evolving according to the group $\mathrm{SU}(1,1)$, rather than the symplectic $\mathrm{Sp}(2,\mathbb{R})$ group. These groups are isomorphic and so the result applies in both cases. $\mathrm{SU}(1,1)$ is used to describe the dynamics of systems such as Bose-Einstein condensates [138] and spin wave transitions [139]. The symplectic group is used more prevalently in quantum optics [130].

As the full controllability of the single-mode system is only achievable when this neutrality condition is met, then the reachable set of operations cannot be the entire set of single-mode symplectic transformations. However, it is clear a subset of the symplectic transformations is reachable. A model where the LARC are met, but the neutrality condition cannot be met, and therefore unstable, was used to determine that there is a bound on the reachable set and hence system is not controllable [12]. By translating the model into the symplectic group it is possible to gain a physical interpretation for the bound, and to discover some other characteristics of the reachable set.

Given $A$ and $B$ from Eq. (3.22), the subset $\Xi$ of $\mathfrak{sp}(2,\mathbb{R})$ with elements of the form $A+vB$, $v \in \mathbb{R}$, is called the set of accessible dynamical generators of the system. To define the reachable set for symplectic operations:

**Definition:** *Reachable set*. The union of all sets of elements $S(t)$ that solve Eq. (3.22) for some choice of control function $u(t)$ is called the reachable set of Eq. (3.22) and is denoted $\mathcal{R}$.

The symplectic generator can be categorised, with the elements of $M$ of $\mathfrak{sp}(2,\mathbb{R})$ being one of

- Parabolic, if $\mathrm{Tr}[M^2] = 0$,
- Hyperbolic, if $\mathrm{Tr}[M^2] > 0$,
- Elliptic, if $\mathrm{Tr}[M^2] < 0$.

The names provide some intuition on the action of the generators. The elliptic generators are neutral resulting in recurring dynamics. Whereas parabolic and hyperbolic generators produce non-recurring trajectories.

Given that the rank criterion with neutrality is sufficient for controllability, any set $\Xi$ that satisfies the LARC and includes elliptic generators would result in a controllable system. It may seem therefore that a set $\Xi$ that excluding all elliptic generators would be sufficient to make a system uncontrollable, since the set $\mathrm{e}^X$ for all $X$ non-elliptic is not $\mathrm{Sp}(2,\mathbb{R})$. However, the time evolved solution to Eq. (3.22) is a product of exponentials, so it is not apparent that this would be the case. It is certain though that any uncontrollable system cannot include elliptic generators.

Neither hyperbolic nor parabolic generators have the recurring properties that would provide full controllability. If $\Xi$ contains parabolic elements, then either the rank criterion is not satisfied or the system is controllable [12]. Therefore, a model that satisfies the rank criterion, but is not controllable, will have $\Xi$ with hyperbolic generators only.

These operators specify a basis of $\mathfrak{sp}(2,\mathbb{R})$:

$$K_x = \frac{1}{2}\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \quad K_y = \frac{1}{2}\begin{pmatrix} -1 & 0 \\ 0 & 1 \end{pmatrix}, \quad K_z = \frac{1}{2}\begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix}, \tag{3.23}$$

with these commutation relations

$$[K_x, K_y] = -K_z, \quad [K_y, K_z] = K_x, \quad [K_z, K_x] = K_y. \tag{3.24}$$

The generator $K_z$ is elliptic. The group elements $\mathrm{e}^{K_z t}$ are 2-dimensional rotations, belonging to $\mathrm{SO}(2)$, a subgroup of $\mathrm{Sp}(2,\mathbb{R})$. In quantum optics experiments, these correspond to phase-plates or phase-shifters that rotate the optical phase of a field. They preserve the field excitations, for which they are know as ‘passive’ or ‘energy-preserving’. The generators $K_x$ and $K_y$ are hyperbolic, and they generate squeezing operations, which do not preserve the number of field excitations. As an example of how this basis can be used, the linear combination $aK_y + K_z$, with $a \in \mathbb{R}$, is elliptic for $|a| < 1$, parabolic for $|a| = 1$, and hyperbolic for $|a| > 1$.

Any linearly independent combination of 3 generators would either subsume the drift term or would imply that elliptical generators were accessible [12]. Hence a control system with only a single control operator, *i.e.* in the form of Eq. (3.22), needs to be considered in a study of general controllability of single-mode unstable systems

### 3.5.2 Uncontrollability of unstable systems

The uncontrollability of unstable single-mode systems was proven in [12]. This proof has been repeated for the $\mathrm{Sp}(2,\mathbb{R})$ group and is provided in full for completeness in Appendix A. The outline of the proof, its key premises and strategy is given here.

An unstable system is one that contains only hyperbolic elements. It can be shown, for instance by the example Eq. (3.40), that a $\Xi$ satisfying this hyperbolic only condition is capable of generating $\mathfrak{sp}(2,\mathbb{R})$. Therefore the LARC will also be satisfied when $\Xi$ contains all hyperbolic elements.

**Lemma 3.1.** *If $\Xi$ only contains hyperbolic elements then Eq. (3.22) is similar, via a symplectic transformation, to*

$$\dot{S}(t) = (-K_x + bK_z + u(t)K_y)S(t), \quad S(0) = \mathbb{I}_2, \tag{3.25}$$

*where constant $b \in \mathbb{R}$, $|b| < 1$.*

Therefore, the reachable set of any purely hyperbolic system will be symplectically similar to any other purely hyperbolic system, hence any conclusion about one will be applicable to all. The set of accessible controls of Eq. (3.25) is denoted by $\widetilde{\Xi}$, with elements of the form $-K_x + bK_z + vK_y$, $v \in \mathbb{R}$, and its reachable set by $\widetilde{\mathcal{R}}$. The strategy of the proof is to show that for the purely hyperbolic system Eq. (3.25) some part of the single mode symplectic space is unreachable, that is $\widetilde{\mathcal{R}} \neq \mathrm{Sp}(2,\mathbb{R})$.

A general $2 \times 2$ real matrix can be written as

$$X = \begin{pmatrix} x_1 + x_3 & x_2 + x_4 \\ x_4 - x_2 & x_1 - x_3 \end{pmatrix}, \quad x_i \in \mathbb{R}. \tag{3.26}$$

This form allows the definition of the following lemma.

**Lemma 3.2.** *If $X \in \widetilde{\mathcal{R}}$ then the function*

$$f(X) := (x_1 - x_4)^2 - (x_2 - x_3)^2 \tag{3.27}$$

*satisfies*

$$\begin{aligned} f(X) &\geq 1, \\ \dot{f}(X) &\geq 0, \\ \dot{f}(X = \mathbb{I}) &\geq 1, \end{aligned} \tag{3.28}$$

*for any choice of* $u(t)$ *in Eq. (3.25).*

As the initial ($t = 0, X = \mathbb{I}$)) value of $f(X)$ is 1, and for the purely hyperbolic system Eq. (3.25), $f$ is always non-decreasing, then $f(X)$ will always be greater than 1. As there exists $X \in \mathrm{Sp}(2, \mathbb{R})$ for which $f(X) < 1$, then the reachable set is not the entire symplectic group, that is $\widetilde{\mathcal{R}} \neq \mathrm{Sp}(2, \mathbb{R})$. It is possible to use this bound to gain a physical interpretation in a quantum optical setting, which is given below.

### 3.5.3 Singular value decomposition

Any symplectic matrix can be decomposed into the product of two rotation and one squeezing matrices, or a single rotation where there is no squeezing. This is a singular value decomposition (SVD) for symplectic transformations. This is often referred to as the Euler [140] or Bloch-Messiah [141] decomposition.

**Definition:** *Singular value decomposition.* Define

$$\mathrm{SO}(2) := \left\{ \begin{pmatrix} \cos[\theta] & -\sin[\theta] \\ \sin[\theta] & \cos[\theta] \end{pmatrix} \middle| \, \theta \in \mathbb{R} \right\} \tag{3.29}$$

and

$$\mathcal{Z}(2, \mathbb{R}) := \{ \mathrm{diag}(1/z, z) \mid z \in \mathbb{R}, z \geq 1 \} \ . \tag{3.30}$$

So matrices in $\mathrm{SO}(2)$ can be described by a single parameter $\theta$, which is the phase rotation, and matrices in $\mathcal{Z}(2, \mathbb{R})$ can be described by a single parameter $z$, which is the level of squeezing. Any $X \in \mathrm{Sp}(2, \mathbb{R})$ can be decomposed as either

$$X = R_\theta Z R_\phi \quad \text{or} \quad X = R_\theta, \tag{3.31}$$

where $R_\theta, R_\phi \in \mathrm{SO}(2)$ and $Z \in \mathcal{Z}(2, \mathbb{R})$.

Note that this decomposition is not unique. It is trivial to see that any additional rotation of $2\pi$ will be equivalent for instance. Slightly more subtle is that matched additional rotations of $\pi/2$ in $\theta$ and $-\pi/2$ in $\phi$ combined with the anti-squeezing is also equivalent.

For the singular value decomposition to be unique, the allowed angles must be bounded such that

$$-\pi + \theta_0 \leq \theta < \pi + \theta_0, \quad -\frac{\pi}{2} + \phi_0 \leq \phi < \frac{\pi}{2} + \phi_0, \tag{3.32}$$

where $\theta_0$ and $\phi_0$ are arbitrary but fixed. See Appendix B.1 for a justification of these bounds. Values of $z < 1$ result in equivalent squeezing of $1/z$ in the orthogonal squeezing plane, and so only values of $z \geq 1$ need to be considered.

The decomposition can be physically interpreted as a phase rotation, followed by some squeezing, and then another phase rotation. As mentioned previously the phase rotations are passive (energy preserving), whereas the squeezing operations are not. If the entire symplectic transform is passive, and hence there is no squeezing operation, then it can be described by a single phase rotation.

Note that the SVD of a matrix generally does not require that left-singular vectors and right-singular vectors ($R_\theta$ and $R_\phi$ in this case) are in $\mathrm{SO}(2)$, simply that they are unitary. There are other decompositions therefore where the left-singular vectors and right-singular vectors are orthogonal, but not necessarily rotations. This needs to be considered when using general SVD algorithms with symplectic matrices, as discussed in Subsection 2.6.2.

The singular value decomposition allows symplectic matrices to be characterised in terms of the parameters $\theta$, $\phi$ and $z$. These are convenient for the physical interpretation of the reachable set, and in its visualisation. The bound on the reachable set can be redefined in these coordinates. The derivation of the coordinate transformation is given in Appendix B.2. Lemma 3.2 can be restated as:

**Corollary 3.1.** *If $X \in \widetilde{\mathcal{R}}$ then the function*

$$g(X) := \cos[2\theta]\cos[2\phi] - \lambda(z, \phi)\sin[2\theta]\,, \tag{3.33}$$

*where*

$$\lambda(z,\phi) := \frac{1}{2}\left(z^2 + \frac{1}{z^2}\right)\sin[2\phi] - \frac{1}{2}\left(z^2 - \frac{1}{z^2}\right), \tag{3.34}$$

*satisfies*

$$g(X) \geq 1, \tag{3.35}$$

$$\dot{g}(X) \geq 0, \tag{3.36}$$

$$\dot{g}(X = \mathbb{I}) \geq 1. \tag{3.37}$$

*for any choice of $u(t)$ in Eq. (3.25).*

### 3.5.4 Reachable sets of unstable systems

The Corollary 3.1 can be used to provide some bounds on $\widetilde{\mathcal{R}}$ for unstable systems in the SVD coordinate system. By considering all possible trajectories of the evolution, starting from the identity, it is possible to prove that pure passive operations (those with no squeezing component) cannot achieved for hyperbolic systems. Although the identity is included in the set passive operations, and it is trivial to see that at $t = 0$ this is reached, however all other passive rotations are not reachable. This is formalised in the following theorem:

**Theorem 3.1.** *If Eq. (3.22) is restricted to hyperbolic dynamical generators then its reachable set does not contain any elements of $\mathrm{SO}(2)$ except for $\mathbb{I}$.*

The full proof of this theorem is given in Appendix C. In practical terms this means that for unstable systems no manipulation of available controls can achieve, at any time, any optical phase-shift operation. As a reminder, unstable systems are those with only the potential for hyperbolic generators, but which are able to generate the whole symplectic algebra. This result demonstrates that, for single-mode systems, it is not possible to achieve the equivalent of an elliptic generator by any sequence of hyperbolic generators. This suggests that positive squeezing is unavoidable for systems evolving under purely hyperbolic generators.

### 3.5.5 Example system: controlled squeezing Hamiltonians

The reachable operations of single-mode systems with quadratic Hamiltonians, with restrictions on the types of available generator (elliptic, parabolic, hyperbolic), can be explored using a model with drift and control Hamiltonians as squeezing operations along different directions.

$$\hat{H} = \hat{H}_A + u(t)\,\hat{H}_B, \tag{3.38}$$

where the drift Hamiltonian $\hat{H}_A$ and the control Hamiltonian $\hat{H}_B$ are defined as

$$\begin{aligned}\hat{H}_A &= \frac{(1-c)\hat{x}^2 - (1+c)\hat{p}^2}{2},\\ \hat{H}_B &= -\frac{\hat{x}\hat{p}+\hat{p}\hat{x}}{2}.\end{aligned} \tag{3.39}$$

These can be represented in the Hamiltonian generator matrix format (see Sec. 3.3) as

$$H_A = \begin{pmatrix} 1-c & 0 \\ 0 & -c-1 \end{pmatrix}, \quad H_B = \begin{pmatrix} 0 & -1 \\ -1 & 0 \end{pmatrix}, \tag{3.40}$$

By applying $\Omega$ this gives the open-loop control problem in terms of symplectic generators

$$\dot{S}(t) = (A + u(t)B)S(t), \tag{3.41}$$

where

$$A \begin{pmatrix} 0 & -(1+c) \\ -(1-c) & 0 \end{pmatrix}, \quad B = \begin{pmatrix} -1 & 0 \\ 0 & 1 \end{pmatrix}. \tag{3.42}$$

This example will be used to provide an investigation into of the reachable operations. It be can easily configured to explore the three generator types, as:

$|c| < 1 \implies$ hyperbolic

$|c| = 1 \implies$ parabolic

$|c| > 1 \implies$ elliptic

For investigating purely hyperbolic systems the obvious case is where $c = 0$, as this is furthest from the elliptic region. However values of $c$ near boundary of the elliptic-hyperbolic region are also worthy of investigation.

### Reachable set visualisation

The visualisation uses the coordinate system from the singular value decomposition defined in Sec. 3.5.3 and uses $(z, \theta, \phi)$ as graph axes. For the volume to uniquely contain the symplectic group then bounds must be set on the rotation angles by setting fixed values for $\theta_0$ and $\phi_0$. These values are most suitable for clearly displaying the symmetry in the reachable operations

$$(\theta_0, \phi_0) = (0, \frac{\pi}{2}), \tag{3.43}$$

which results in the bounds

$$-\pi \leq \theta < \pi, \quad 0 \leq \phi < \pi. \tag{3.44}$$

All symplectic group operations can be represented by a point in this cubic space. Note that through the definition of $\mathcal{Z}$ in Eq. (3.30) $z > 1$ this provides a unique mapping. It is unbounded above as there is no theoretical limit on squeezing, making the symplectic group non-compact, and is related to the infinite set of energy eigenstates for QHO systems. The $z = 1$ plane contains non-unique symplectic operations, as any combination of $\theta$ and $\phi$ that sum to the same angle are equivalent to a single rotation of $\theta + \phi$. However, the main point of interest in this plane is the identity, as this is the start point of the operator evolution trajectory.

Although any reconstruction of a symplectic operation from the SVD parameters where $z = 1, \theta = -\phi$ is the identity, a specific 'singular value decomposition of identity' can be found by considering the limit $t \rightarrow 0$ for some known reachable element. For example, from Eq. (3.41) with $S(0) = \mathbb{I}, c = 0, u(0) = 0$

$$\exp\left[\begin{pmatrix} 0 & -1 \\ -1 & 0 \end{pmatrix} t\right] \tag{3.45}$$

as $t \to 0$. Consider $t = \frac{1}{n}$, where $n \in \mathbb{N}$.

$$\begin{aligned} \exp\left[\begin{pmatrix} 0 & -1 \\ -1 & 0 \end{pmatrix} \frac{1}{n}\right] &= \left( R_{-\frac{3\pi}{4}} \begin{pmatrix} \frac{1}{e} & 0 \\ 0 & e \end{pmatrix} R_{\frac{3\pi}{4}} \right)^{\frac{1}{n}} \\ &= R_{-\frac{3\pi}{4}} \begin{pmatrix} \frac{1}{e} & 0 \\ 0 & e \end{pmatrix}^{\frac{1}{n}} R_{\frac{3\pi}{4}}. \end{aligned} \tag{3.46}$$

In the limit as $n \to \infty$ the singular value decomposition of the identity is singled out as

$$\mathbb{I} = R_{-\frac{3\pi}{4}} R_{\frac{3\pi}{4}}. \tag{3.47}$$

The analysis used in the proof of the reachable set in Appendix C can be applied to give a further bound for unstable systems. The proof of Lemma C.1 states that $\sin[2\theta] > 0$ for $S \neq \mathbb{I}$ for unstable systems. In the range chosen for symplectic uniqueness Eq. (3.44), this implies

$$-\pi < \theta < -\frac{\pi}{2}, \quad 0 < \theta < \frac{\pi}{2}. \tag{3.48}$$

Given that the identity, where the operator evolution starts at $t = 0$, has been shown to be located at $\theta = -3\pi/4$, and that there can be no discontinuity in the SVD parameters during the time evolution, then

$$-\pi < \theta < -\frac{\pi}{2}. \tag{3.49}$$

This is further illustrated in the visualisation of the reachable set in Fig. 3.1 where there appear to be inescapable buckets, so starting at $\theta = -3\pi/4$ there could no continuous time evolution that would reach the $0 < \theta < \frac{\pi}{2}$ range.

### Reaching operations through optimal control

Previously in this section it has been shown that for unstable systems some region of the symplectic space is unreachable. This does not imply that all the symplectic transformations outside of the unreachable region are reachable operations. It is clear that for any system evolving under

Eq. (3.15) that some symplectic operations are achievable, as any choice of $u(t)$ will give some symplectic operation at all values of $t$. Quantum optimal control (QOC) methods, as introduced in Section 1.6, can be used determine whether a specific operator can be realised, by attempting to find some function $u(t)$ that achieves this. Finding such a function shows that the operator is reachable. The failure of a QOC algorithm to find a control function that achieves the objective does not necessarily imply that one does not exist. However, a systematic use a QOC algorithm will show points the SVD coordinate space of the single-mode symplectic group that can be reached. To gain some indication of the boundaries of the reachable set many operator optimisations must be attempted, hence the computational efficiency of the algorithm is important, making GRAPE a good choice.

The GRAPE algorithm requires that the control function is piecewise constant, so there is a discrete set of parameters (amplitudes in the timeslots) that can be optimised. This in itself is a constraint on the function that may (or may not) exclude some operations. The evolution time $T$ is split into $M$ equal time slices of length $\Delta t$ with the time at the beginning of each slice $t_k$. The amplitude $u(t_k)$ is constant throughout the time slice, hence the piecewise constant control function $u(t)$ corresponds to a set of $M$ real values. Setting $M = 10$ provides a balance between computational efficiency and limiting the constraint on $u$. Thus GRAPE can be used to find a set of $u(t_k)$ that results in an $S_T$ that is sufficiently close to $S_{\text{target}}$ for it to be considered 'reached' and therefore in the reachable set. If the control parameter optimisation for $S_{\text{target}}$ fails it is 'unreached' and in this subsection is assumed outside of the reachable set. The target is considered reached in this case if $\mathcal{E} < 10^{-3}$.

The Hamiltonian generator matrices are of the form

$$H_k = H_A + u(t_k)H_B, \quad u(t_k) \in \mathbb{R}. \tag{3.50}$$

where $H_A$ and $H_B$ are as given in Eq. (3.40).

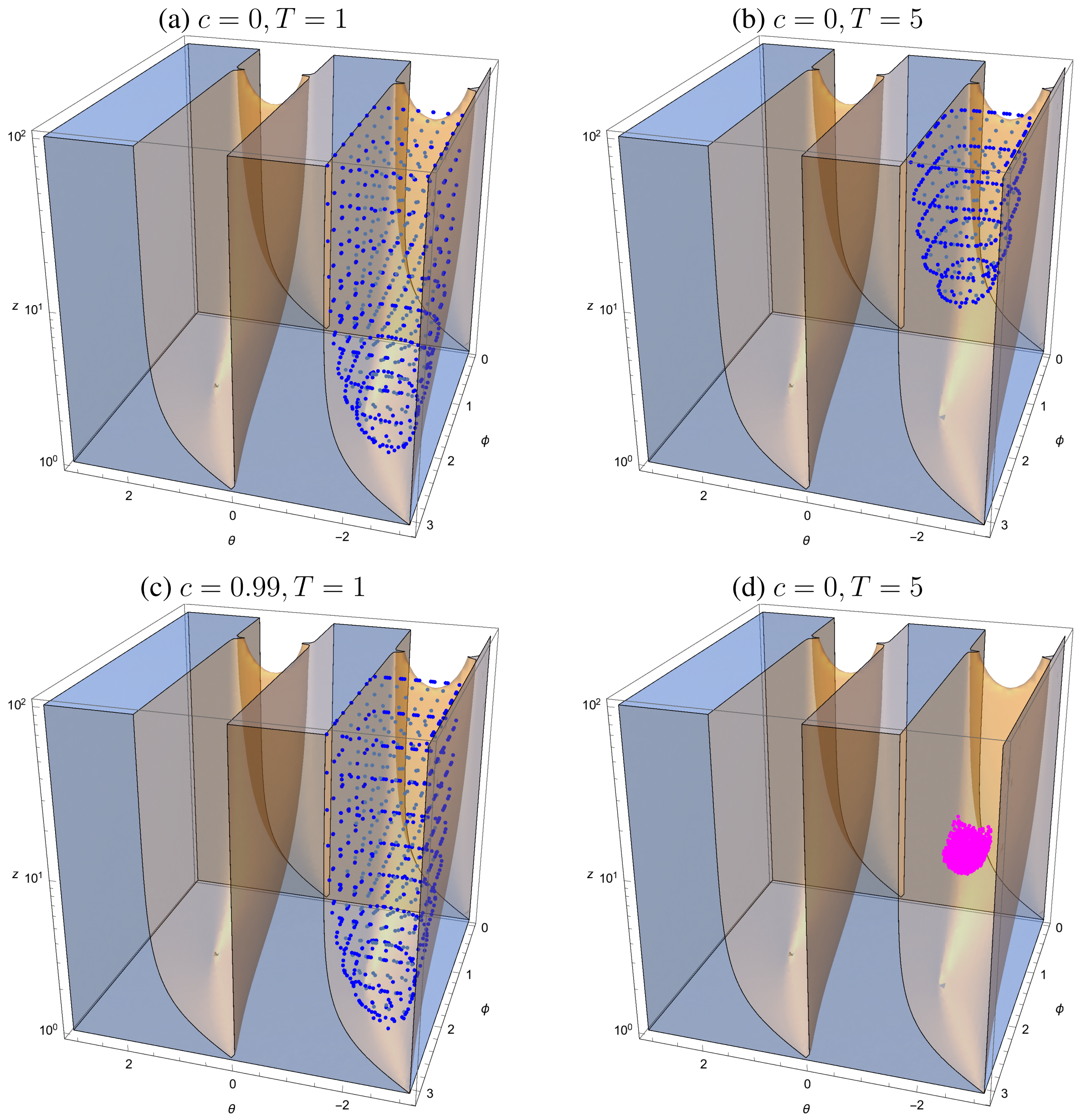
A discretisation scheme in the $(z, \theta, \phi)$ space is required for a set of possible target symplectics. To produce the results here points at $\pi/12$ intervals in the angular directions and 10 logarithmically equal intervals between $z = 1$ and the arbitrary upper bound of $z = 100$. Between any two reached and unreached points there exists a reachable set boundary. A point closer to this boundary can be found by interpolating the coordinates for another point between them. See Section 2.6.1 for more detail on how the boundary reached points were found.

The region proved unreachable, given by Eq. (3.35) and the numerical results of the tests for unstable systems are shown in Fig. 3.1. Note that the points shown are those reached specifically *at* evolution time $T$ rather than up to time $T$. The reached points are restricted to a set centred around $(\theta, \phi) = (-3\pi/4, 3\pi/4)$, which are the coordinates of the identity. The boundary points form a basket or bowl shape with the lowest point above the identity. This indicates that rotations are accessible, but only at the expense of additional squeezing. The reached point are all within the bounds of $-\pi < \theta < -\pi/2$, as predicted by the analytics Eq. (3.49). The basket is symmetrical, giving other bounds of $\pi/2 < \phi < \pi$. This suggests that the reachable set is tighter than the complement of the unreachable region. Due to the symmetry of the singular value decomposition, it is likely that there is a lemma of similar form to Lemma 3.2 that would provide these bounds in $\phi$.

The volume not found unreachable is translationally symmetric in the $0 < \theta < \pi$ region to the $-\pi < \theta < 0$ region. However, reachable points are only found in the $-\pi < \theta < 0$ region. One may ask why there is no basket of reachable points in the volume in the $0 < \theta < \pi$ region. As explained above, all operator evolution trajectories start at $\mathbb{I}$, and the identity lies on the $\theta = -\phi, z = 1$ line. All points on this line for $\theta > 0$ are in the proven unreachable region, hence no trajectories can start there, and there are no continuous evolution paths into this region, meaning there are no reachable points in the $0 < \theta < \pi$ region.

The points reached by random evolution of the system shown in plot (d) of Fig. 3.1. These are the end points of the evolution based on samples where the piecewise constant amplitudes were uniformly distributed in $[-1, 1]$ with 10 timeslots. 10,000 samples were made, and the 1,000 most spatially separated are plotted. The Hamiltonian parameters are identical to those in plot (b), and so the reachable set is equivalent. The reached points all appear in the bottom of the basket. This demonstrates the advantage of control optimisation in exploring the bounds of the reachable set.

In tests with stable systems, with $|c| \geq 1.1$ all points were reached. For $|c| = 1.01$ the optimiser was unable to find a suitable control function for some $S_{\text{target}}$. These unreached points were


(a) $c = 0, T = 1$

(b) $c = 0, T = 5$

(c) $c = 0.99, T = 1$

(d) $c = 0, T = 5$

**Figure 3.1:** The enclosed blue region is unreachable by Eq. (3.35). The blue points are reached operations in the $(z, \theta, \phi)$ basis after time $T$. The boundary points, found by interpolation, are depicted as the darker blue points. The unreached points are not shown. Specific parameters: (a) $c = 0, T = 1$; (b) $c = 0, T = 5$; (c) $c = -0.99, T = 5$, (d) $c = -0, T = 5$. The magenta points in plot (d) were reached by random control pulses.

predominantly in the region found reachable for $-1 < c < 1$. However, it is most likely that this is due to the constraints placed on the pulse optimisation, and demonstrates the difficulty of finding a solution near the edge of stability. In the case for $c = -0.99$ the reachable set is broader – the sides of the bowl are less steep. This broadening of the reachable set as $c$ goes near the boundary of stability indicates that the squeezing at the expense of rotations is less, and the control system has become in some sense more stable. For the parabolic system $|c| = 1$ the reached points were constrained to a plane, as would be expected because the generators do not fulfil the rank criterion.

## 3.6 Multi-mode systems

Having the recurring elements of the generating algebra, referred to as neutral, available to the control system has been shown sufficient, with the rank criterion, for controllability on $n$ modes [23]. In the previous section neutrality was proven necessary for control generators to produce passive operations for single-mode systems. Extending this result to any number of modes is clearly desirable, however it remains a long-standing open problem in mathematical control theory [10, 142].

To prove the necessity of neutral generators for controllability it would suffice to show that there is no way of accessing the subgroup of passive operations because the non-neutral generators induce irreversible squeezing. It is possible that an analytical approach such as that in used in section 3.5.2 could be applied to $n$-modes, though clearly this would be more challenging. The numerical approach used in section 3.5.5 is applicable to at least two modes. With a single mode, reachable points, when elliptic generator elements are not accessible, the system is restricted to unbounded squeezing within a small angular region. A similar finding for two modes would motivate an analytical study into trajectories with irreversible squeezing.

As, for single-mode, unstable systems the inaccessibility of passives is due to squeezing being unavoidable, then it is reasonable to conjecture that this also true for $n$-modes. Given some general $n$-mode symplectic operator $S$, acting the vector of quadrature operators Eq. (3.11), a measure of the squeezing involved is given by

$$\xi = \max \operatorname{eig}[SS^{\intercal}] - 1, \tag{3.51}$$

which only reaches zero when $S$ is a passive element. For single-mode, unstable systems the only passive element in the reachable set is the identity.

### 3.6.1 Example two-mode systems

The controlled dynamics operator equation Eq. (3.15) for a two-mode system with one control operator is

$$\dot{S}(t) = (A + u(t)B)\, S(t), \quad S(0) = \mathbb{I}_4, \tag{3.52}$$

where the control function $u(t)$ is a real valued function defined for all positive times $t$, $A, B \in \mathfrak{sp}(4, \mathbb{R})$, and $S$ is a symplectic operator $\mathrm{Sp}(2, \mathbb{R})$ acting on the quadrature operator vector Eq. (3.11) with $n = 2$.

It is not trivial to find a control model where a single parameter can be used to set the neutrality of the system, as in the single-mode example. However, there are neutral and non-neutral examples with some physical interpretation.

#### Non-neutral system example

For a non-neutral example a combination of generators is required such that $A + vB$ is non-neutral for all values of $v \in \mathbb{R}$

In the $H$-matrix form, defined in section 3.3

$$H_A = \begin{pmatrix} 0 & 0 & d & c \\ 0 & 0 & c & -d \\ d & c & 0 & 0 \\ c & -d & 0 & 0 \end{pmatrix}, \quad H_B = \begin{pmatrix} a & -b & 0 & 0 \\ -b & -a & 0 & 0 \\ 0 & 0 & e & -f \\ 0 & 0 & -f & -e \end{pmatrix}, \tag{3.53}$$

where $H_A$ corresponds to two-mode squeezer and $H_B$ corresponds to two single-mode squeezers.

These provide

$$A = \begin{pmatrix} 0 & 0 & c & -d \\ 0 & 0 & -d & -c \\ c & -d & 0 & 0 \\ -d & -c & 0 & 0 \end{pmatrix}, \quad B = \begin{pmatrix} -b & -a & 0 & 0 \\ -a & b & 0 & 0 \\ 0 & 0 & -f & -e \\ 0 & 0 & -e & f \end{pmatrix}. \tag{3.54}$$

The elements $A$ and $B$ generate the Lie algebra $\mathfrak{sp}(4, \mathbb{R})$, as can be be verified using the LieTree algorithm. The method is described in section Section 2.5

Some specific values are required for a numerical treatment, for instance:

$$A = \begin{pmatrix} 0 & 0 & 1 & -2 \\ 0 & 0 & -2 & -1 \\ 1 & -2 & 0 & 0 \\ -2 & -1 & 0 & 0 \end{pmatrix}, \quad B = \begin{pmatrix} -2 & -1 & 0 & 0 \\ -1 & 2 & 0 & 0 \\ 0 & 0 & -4 & -3 \\ 0 & 0 & -3 & 4 \end{pmatrix}. \tag{3.55}$$

**Neutral system**

For a controllability comparison a neutral example is also required

$$H = \begin{pmatrix} -2 & 0 & 0 & 0 \\ 0 & -1 & 1 & 0 \\ 0 & 1 & -7 & 0 \\ 0 & 0 & 0 & -4 \end{pmatrix} \tag{3.56}$$

implies

$$A_{\text{neutral}} = \begin{pmatrix} 0 & 2 & 0 & 0 \\ -1 & 0 & 0 & -1 \\ 1 & 0 & 0 & 7 \\ 0 & 0 & -4 & 0 \end{pmatrix}, \tag{3.57}$$

As the drift $A$ in this case has all imaginary eigenvalues, then this guarantees access to neutral elements for the system when combined with $B$ as above.

### 3.6.2 Two-mode singular value decomposition

In order to systemically probe the two-mode symplectic space the singular value decomposition (SVD) can be used to construct an $S_{\text{target}}$ based on component parameters.

A general two-mode symplectic may be decomposed as

$$S = R_1 Z R_2, \tag{3.58}$$

where

$$Z = \text{diag}(z_1, 1/z_1, z_2, 1/z_2), \tag{3.59}$$

and

$$R = R_\alpha R_\beta R_\theta R_\phi \tag{3.60}$$

with

$$\begin{aligned}
R_\alpha &= \begin{pmatrix} R_{\text{O2}}(-\alpha) & 0 \\ 0 & R_{\text{O2}}(-\alpha) \end{pmatrix} \\
R_\beta &= \begin{pmatrix} R_{\text{O2}}(\beta/2) & 0 \\ 0 & R_{\text{O2}}(-\beta/2) \end{pmatrix} \\
R_\theta &= \begin{pmatrix} \cos[\theta/2] & 0 & \sin[\theta/2] & 0 \\ 0 & \cos[\theta/2] & 0 & \sin[\theta/2] \\ -\sin[\theta/2] & 0 & \cos[\theta/2] & 0 \\ 0 & -\sin[\theta/2] & 0 & \cos[\theta/2] \end{pmatrix} \\
R_\phi &= \begin{pmatrix} R_{\text{O2}}(\phi/2) & 0 \\ 0 & R_{\text{O2}}(-\phi/2) \end{pmatrix}
\end{aligned} \tag{3.61}$$

where

$$R_{\text{O2}}(x) := \begin{pmatrix} \cos x & -\sin x \\ \sin x & \cos x \end{pmatrix}, \tag{3.62}$$

hence

$$S = S(\alpha_1, \beta_1, \theta_1, \phi_1, z_1, z_2, \alpha_2, \beta_2, \theta_2, \phi_2) \tag{3.63}$$

### 3.6.3 Numerical study using optimal control

Quantum optimal control can be used to determine if specific symplectic operations can be reached. The approach is described in Section 3.5.5. For the piecewise control function, 24 equal timeslots should provide sufficient optimisation parameters, as the $\mathrm{Sp}(2, \mathbb{R})$ group has dimension $n(2n + 1) = 10$, and therefore 24 variables should be sufficient in the optimisation algorithm. Evolution time of $T = 1$ avoids the elements the symplectic exceeding computational limits. In all tests discussed below optimisations were initialised with randomised control functions with amplitudes in the range $[-1, 1]$. No bounds were placed on the amplitudes during the optimisation.

#### two-mode space sampling

As there are 10 SVD parameters in the two-mode symplectic, any deterministic approach to examining the entire space evenly with respect to these parameters gives rise to an infeasibly large number of combinations for a control optimisation to be attempted on all of them. Primarily of interest are the squeezing degrees of freedom. A feasible approach is to evenly span the squeezing parameters space and use random sampling of the angular parameters to explore what can and cannot be reached.

Tests were completed using control optimisation to determine the reachability of points for all combinations of $z_1, z_2$ with 10 equally values in $[1, 10]$. For each of the angular parameters, values were selected randomly for the full range of $[-\pi, \pi]$ at intervals of $\pi/4$. These were used to form some $S_{\text{target}}$ based on the composition Eq. (3.58). For each pair of squeezing parameters 100 random angle combinations were attempted.

In the vast majority of scenarios, with the neutral example generators of Eq. (3.55), the $S_{\text{target}}$ was found to be reachable. However, with the non-neutral example of Eq. (3.57), none of the targets were found to be reachable. As only a limited set of points were tested, these tests suggest that potentially the reachable set is very narrow in one or more parameters. This is investigated by seeing if controls can be optimised to reach targets close to known reachable points.

### Shifted targets

It is possible calculate on operation that is very close to another that is known to be in the reachable set. This can be done by evolving the system with some $u(t)$ resulting in some $S(T)$ at $t = T$, as per Eq. (3.17). A small change can introduced in any of the 10 parameters ($\alpha_1, \beta_1, \theta_1, \phi_1$, $z_1, z_2$, $\alpha_2, \beta_2, \theta_2, \phi_2$) and a new operator calculated as a result. This can be used as some target $S_{\text{target}}$ in control optimisation. The SVD parameters can be extracted from $S(T)$ using a method described in Subsection 2.6.3. From this the target can be constructed using Eq. (3.58), with

$$\begin{aligned} S_{\text{target}} = S(&\alpha_1 + \delta\alpha_1, \beta_1 + \delta\beta_1, \theta_1 + \delta\theta_1, \phi_1 + \delta\phi_1, \\ &z_1 + \delta z_1, z_2 + \delta z_2, \\ &\alpha_2 + \delta\alpha_2, \beta_2 + \delta\beta_2, \theta_2 + \delta\theta_2, \phi_2 + \delta\phi_2)\,, \end{aligned} \tag{3.64}$$

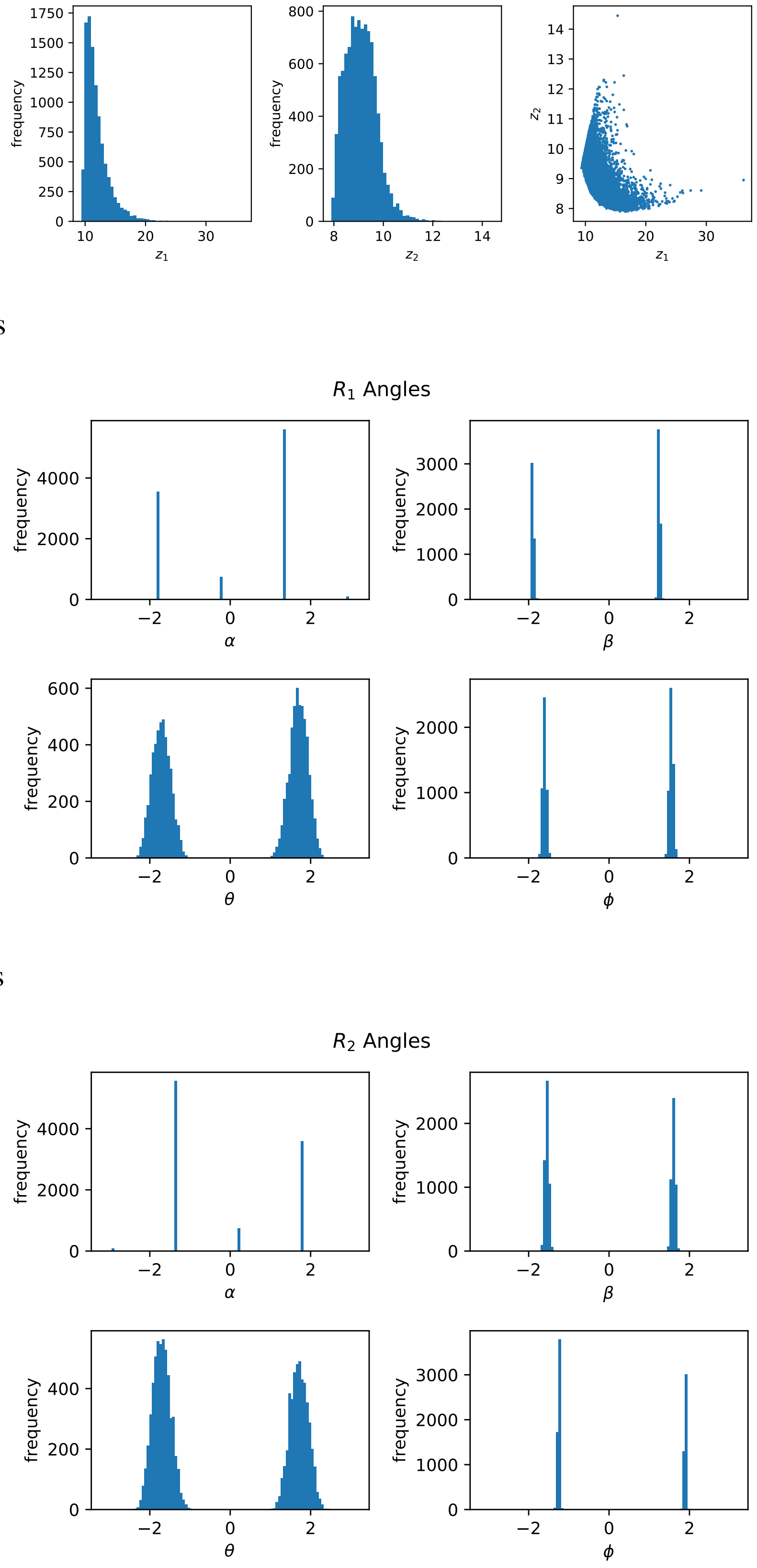
where the $\alpha_1, \beta_1, \ldots$ are the SVD parameters of $S(T)$ and the $\delta\alpha_1, \delta\beta_1, \ldots$ are the small changes, for which it typically makes sense have only one non-zero.

Control optimisation attempts were made for targets shifted, as described above, from the evolved operator $S(T)$. The initial control pulse for the optimisation was the random pulse that evolved $S(T)$. Using the non-neutral example it was possible to find shifted targets (to within an infidelity target of $10^{-5}$) where the change in squeezing ($\delta z_1 + \delta z_2$) between that randomly evolved $S(T)$ and target $S_{target}$ was in the range $[-1, 2]$.

Angle shifted targets were more difficult to reach. A change of $\delta\theta$ in range $[\pm 0.05]\text{rad}$ is reachable with with an infidelity target of $10^{-3}$, but $\delta\theta = \pm 0.01$ is not reachable with an infidelity target of $10^{-5}$. The other angles were found to be more sensitive. $\delta\phi = \pm 0.01$ was reachable with an infidelity target of $10^{-3}$. Whereas $\alpha, \beta$ did not accept any shifting $> 0.0001$, even for infidelity target of $10^{-3}$.

## 3.6.4 Numerical study using random evolution

It is clear that the reachable set for the non-neutral example is not zero-volume. The system evolving under the generators must result in some transformation. Using the same piecewise constant control function parameters, with the non-neutral example generators of Eq. (3.57), the system was



(a) Squeezing parameters

(b) $R_1$ parameters

(c) $R_2$ parameters

**Figure 3.2:** Decomposition of 10000 random pulse evolutions. All evolved for $T = 1$, with 24 timeslots.

| Measure | Freq |
|---|---|
| $z_1 + z_2$ | 349 |
| $\|Z\|_2$ | 759 |
| $z_1$ | 4415 |
| $z_2$ | 16 |
| $z_1$ & $z_2$ | 10 |

**Table 3.1:** Non-monotonicity of squeezing for 10000 random pulse evolutions. All evolved for $T = 1$, with 24 timeslots.

evolved 10000 times using different random control functions.

### SVD parameter distribution

The evolved $S(T)$ were decomposed into the 10 SVD parameters, the 4 angles of $R_1$, the two squeezing parameters, and the 4 angles of $R_2$. For details of how these parameters were extracted see section 2.6.3. Fig. 3.2 shows the distribution of the SVD parameters. It should be noted that $z_1 > z_2$ for any single $S(T)$ due to the decomposition method, which explains some features of the distributions of Fig. 3.2(a), such as $z_1 > z_2$ on average and the hard-edge $z_1 = z_2$ in the rightmost plot.

The distributions show narrow angle ranges, in particular for $\alpha, \beta, \phi$. This provides some explanation as to why the systematic search for the reachable set of the non-neutral example found no reachable targets. The width of the angle ranges is greater than that found achievable in the shifted target attempts. This hints at potentially a similar feature to that seen in the single-mode unstable system, where rotations were only accessible at the expense of squeezing.

### Anti-squeezing

A key conjecture of the uncontrollability of unstable systems is that squeezing is inevitable during the operator evolution over time. In a piecewise constant setting this can investigated by examining the degree of squeezing in the operator evolved in each timeslot. In two-mode systems the individual mode squeezing $z_1, z_2$ may not be expected increase monotonically, but as the level of squeezing may be transferred between modes, then the most valid measure would be

$$\|Z\|_2 = \mathrm{Tr}\,[Z^T Z], \tag{3.65}$$

If the SVD has already been completed, then this is more efficient than calculating Eq. (3.51).

The main objective is to show that passive operations are inaccessible for two-mode unstable systems. One approach is to show that squeezing is unavoidable. The 10,000 random operator evolutions can be analysed to see if they contain anything to counter this conjecture. Table 3.1 shows the number of instances of non-monotonicity, by various measures, for the squeezing parameters. Note that $z_1$ & $z_2$ non-monotonic does not imply that they both decreased in the same timeslot, just that they each decreased during some timeslot of the evolution. The focus henceforth will be on the $\|Z\|_2$ measure, and any timeslot where $\|Z_{k+1}\|_2 < \|Z_k\|_2$ will be referred to as an *anti-squeezing* timeslot. There are then some combinations of $S_{0:k}$, the time evolution up to timeslot $k$, and $S_k$, the propagator in timeslot $k$ (such that $S_{0:k+1} \equiv S_k S_{0:k}$), that result in anti-squeezing. As shown in Table 3.1 there are 759 examples in the 10000 sample set where there is at least one anti-squeezing timeslot, as measured by the change in $\|Z\|_2$. The timeslot index in which maximum anti-squeezing occurs will be referred to as $k'$.

Given that anti-squeezing does occur in this unstable system, what are the features of the timeslot dynamics that give rise to it? Are there specific generators that always anti-squeeze, or does the system enter some state where squeezing can only be reversed, or does it require some combination of both? Fig. 3.3 a.) shows the frequency of maximum anti-squeezing in one timeslot for all the examples that had at least one anti-squeezing timeslot. In the vast majority $\min \Delta\|Z\|_2 > -1$. However some have much more anti-squeezing, with one example having $\min \Delta\|Z\|_2 \approx -22$, this specific example is referred to as the *prime* example. Fig.3.3 b.) shows the effect of the prime anti-squeezing propagator on all the pre anti-squeezing operators. Fig. 3.3 c.) looks at the effect of all anti-squeezing propagators on the prime pre anti-squeezing operator. These two histograms suggest that in the prime example the operator is particularly susceptible to anti-squeezing, or otherwise significant positive squeezing. That is, it is in some highly unstable stage of its evolution.

Fig. 3.4 looks at the effect of a.) the prime anti-squeezing propagator on some randomly

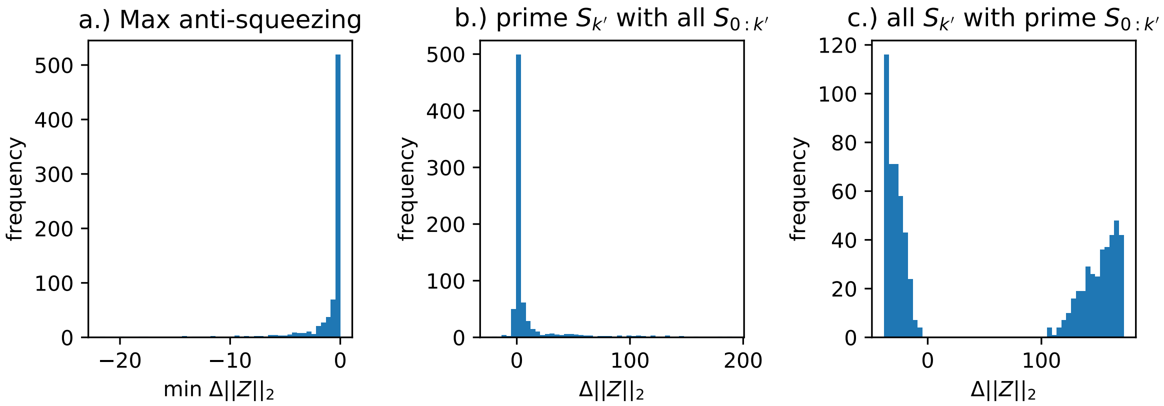


**Figure 3.3:** Analysis of the min $\Delta\|Z\|_2$ (maximum anti-squeezing) for those evolutions that contain at least one anti-squeezing timeslot. The anti-squeezing timeslot for the evolutions is denoted by $k'$. The propagator $S_{k'}$ and evolution operator $S_{0:k'}$ for the example that was found to produce the overall maximum anti-squeezing of all the examples, referred to as the *prime* example, are used for additional analysis. a.) Distribution of the maximum anti-squeezing; b.) max anti-squeezing propagator applied to all anti-squeezing evolutions; c.) all anti-squeezing propagator applied to max anti-squeezing evolution.

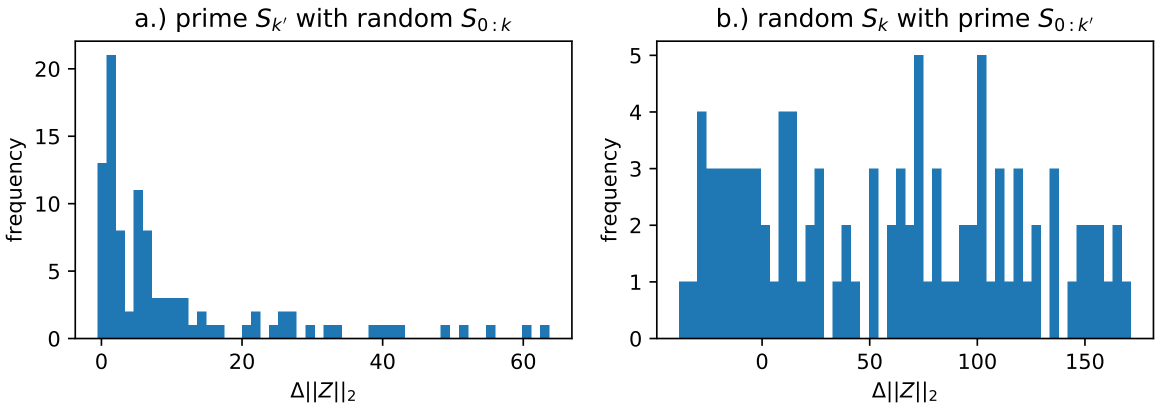


**Figure 3.4:** The prime (see Fig. 3.3) anti-squeezing propagator and evolved operator are combined randomly evolved operators and propagators.

evolved operator and b.) some random propagator on the prime pre anti-squeezing operator. The random amplitudes are sampled in the range $[-1, 1]$, and for the randomly evolved operator, a random timeslot up to the 24th is used. These histograms seem to demonstrate that the prime anti-squeezing propagator is not unusual, whereas the prime pre anti-squeezing operator is much more likely to experience anti-squeezing than an average operator.

As the original conjecture about the unstable system and its generators is based around their

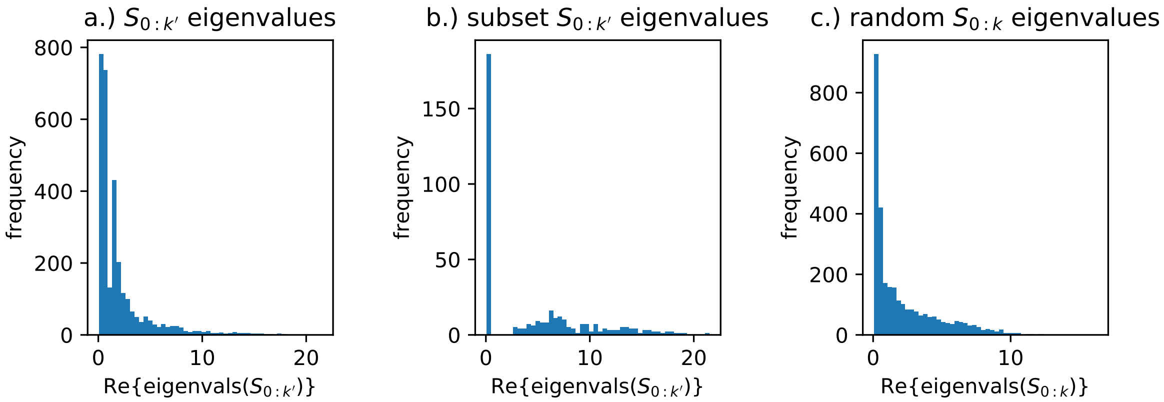


**Figure 3.5:** The eigenvalues of the evolved operator up to the anti-squeezing timeslot. The subset in b.) are those evolutions that have anti-squeezing greater than $2$. These are compared with c.) eigenvalues of randomly evolved operators

eigenvalues, then an examination of eigenvalues is justified when looking for some signature of instability that potentially results in anti-squeezing. Fig. 3.5 attempts find some signature in the eigenvalues of the pre anti-squeezing operators. The eigenvalues are all real for this non-neutral example. The distribution of eigenvalues for the full set of pre anti-squeezing operators, a subset that contains only the operators from the evolutions that have a $\Delta\|Z\|_2 < -2$, and an equivalent size set of eigenvalues from randomly evolved operators are compared. Whereas for the full set of pre anti-squeezing operators the distribution of eigenvalues appears indistinguishable from that of the randomly evolved operators, the subset seems to have a range where eigenvalues are missing from the distribution.

## 3.7 Coupled oscillators

The previous two sections focus on the necessity of neutrality in the generators of a bilinear symplectic control system described by Eq. (3.15). The numerical studies in one and two modes have demonstrated that access to neutral generators combined with the Lie algebra rank criterion (LARC) is sufficient for achieving the full set of symplectic operations. This section looks extend this into many modes by considering a challenging control objective in many-body systems.

Genoni, Serafini, Kim and Burgarth proved that, given satisfaction of LARC, positive-(negative) definite Hamiltonian generating matrices are sufficient for controllability of systems with quadratic

Hamiltonians [13]. This is equivalent to the neutrality condition. In the same study it is proven that chains of coupled oscillators are fully controllable for specific controls acting on one oscillator only, and coupling constants within a specific range. This section looks to use this oscillator chain model to investigate some limits on control with these systems.

### 3.7.1 The oscillator chain model

The Hamiltonian for a chain of $n$ coupled oscillators can be described by

$$\hat{H}_0 = \omega \sum_{j=1}^{n} \left( \hat{a}_j^\dagger \hat{a}_j + \frac{1}{2} \right) + g_1 \sum_{j=1}^{n-1} (\hat{a}_j \hat{a}_{j+1}^\dagger + H.c.) + g_2 \sum_{j=1}^{n-1} (\hat{a}_j \hat{a}_{j+1} + H.c.). \tag{3.66}$$

where all the oscillators modes have the same frequency $\omega$, $\hat{a}_j$ and $\hat{a}_j^\dagger$ are the bosonic mode operators for the $j^{\text{th}}$ mode, and $g_1$ and $g_2$ are coupling constants, which in this model is the same between all adjacent modes.

Control Hamiltonians, which act only on the first mode of chain,

$$\begin{aligned} \hat{H}_1 &= \omega_1 \left( a_1^\dagger a_1 \right), \\ \hat{H}_2 &= \chi (a_1^2 + a_1^{\dagger 2}) \end{aligned} \tag{3.67}$$

are local phase rotation and local squeezing operators respectively. Combined these give the full Hamiltonian of the system

$$\hat{H}(t) = \hat{H}_0 + u_1(t)\hat{H}_1 + u_2(t)\hat{H}_2 \tag{3.68}$$

where $u_1(t)$ and $u_2(t)$ are real valued control functions. The LARC is shown to be satisfied for this combination of Hamiltonians in [13] and its supplementary material. Also therein it is shown that the positive-(negative) definite condition is met so long as $g_1 + g_2 < \omega/2$.

In the terms of the quadrature operators the drift Hamiltonian can be expressed as

$$\hat{H}_0 = \frac{\omega}{2} \sum_{j=1}^{n} \left( \hat{q}_j^2 + \hat{p}_j^2 \right) + g_q \sum_{j=1}^{n-1} \hat{q}_j \hat{q}_{j+1} + g_p \sum_{j=1}^{n-1} \hat{p}_j \hat{p}_{j+1} \tag{3.69}$$

where $g_q = g_1 + g_2$ and $g_p = g_1 - g_2$.

The elements Hamiltonian generator matrix (defined in section 3.3) are therefore

$$H^{(0)}_{l,m} = \begin{cases} \omega & l = m, \\ g_q & |l - m| = 2, l \text{ even}, \\ g_p & |l - m| = 2, l \text{ odd}, \\ 0 & \text{otherwise}. \end{cases} \tag{3.70}$$

as the quadrature operators for different modes commute. For two modes this is

$$H_0 := \begin{pmatrix} \omega & 0 & g_q & 0 \\ 0 & \omega & 0 & g_p \\ g_q & 0 & \omega & 0 \\ 0 & g_p & 0 & \omega \end{pmatrix} \tag{3.71}$$

In terms of the quadrature operators the control Hamiltonians are

$$\begin{aligned} \hat{H}_1 &= \frac{\omega_1}{2}({q_1}^2 + {p_1}^2) \\ \hat{H}_2 &= \chi({q_1}^2 - {p_1}^2) \end{aligned} \tag{3.72}$$

and the corresponding $H$ matrices can easily be recognised from these.

### 3.7.2 Optimal control of the chain

An interesting target for control optimisation is the square root of a boson swap between ends of the chain, as achieving this would clearly demonstrate control of the whole chain, and would induce an entangled state of the first and last oscillator modes. As this transformation is an exchange of information on a length-scale, assuming a model of physical oscillators, then some inference about the *quantum speed limit* could also be made. The Hamiltonian used to generate this target is

$$\hat{H}_{\text{swap}} = \hat{a}_1 \hat{a}_n^\dagger + \hat{a}_n \hat{a}_1^\dagger = \frac{1}{2}(\hat{q}_1\hat{q}_n + \hat{p}_1\hat{p}_n + \hat{q}_n\hat{q}_1 + \hat{p}_n\hat{p}_1) = \hat{q}_1\hat{q}_n + \hat{p}_1\hat{p}_n\,. \tag{3.73}$$

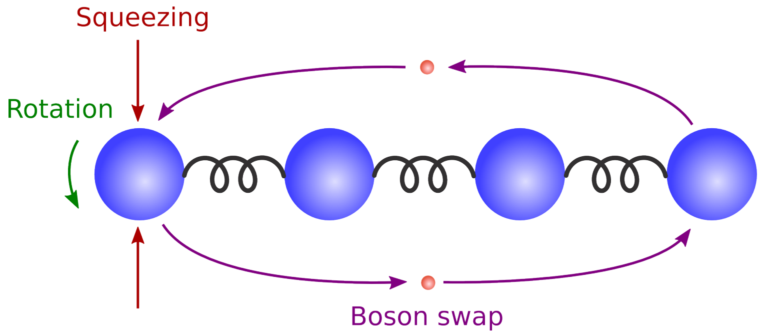


**Figure 3.6:** Schematic of the control model. A chain of coupled oscillators with squeezing and phase-rotation controls acting on the first oscillator mode. The control target is the square root of the operation that swaps a field excitement (boson) from one end of the chain to the other. This entangles the system.

The corresponding Hamiltonian generator matrix can be used to calculate the symplectic operator as

$$S_{\sqrt{\text{swap}}} = e^{-H_{\text{swap}}\Omega\pi} \tag{3.74}$$

A schematic of the full control set up is shown in Fig. 3.6.

Intuition suggests that for longer chains the 'difficulty' of achieving this $S_{\sqrt{\text{swap}}}$ target should increase. A measure of the difficulty is the time required to effect this transformation. This can be interpreted as some kind of quantum speed limit, as in physical systems such as ion traps or coupled nano-oscillators, the number of oscillators in the chain translates to a length scale. A bound is set for the amplitude of the control function, to model the limitation that would exist in a physical experiment.

The GRAPE algorithm in QuTiP was used to find piecewise constant control functions for $u_1$ and $u_2$ that achieve the target operation, using the process described in Section 3.4. Optimisations were attempted for specific total evolution times $T$, and the minimum fidelity error was recorded. Optimisation was repeated for $T$ in increasing increments until the operation could be achieved with an fidelity error of less than $0.02$ for 10 consecutive $T$. More details on the control optimisation process, including the challenges of optimal control in many-body oscillator systems and techniques for overcoming them, are given in Section 2.6.1

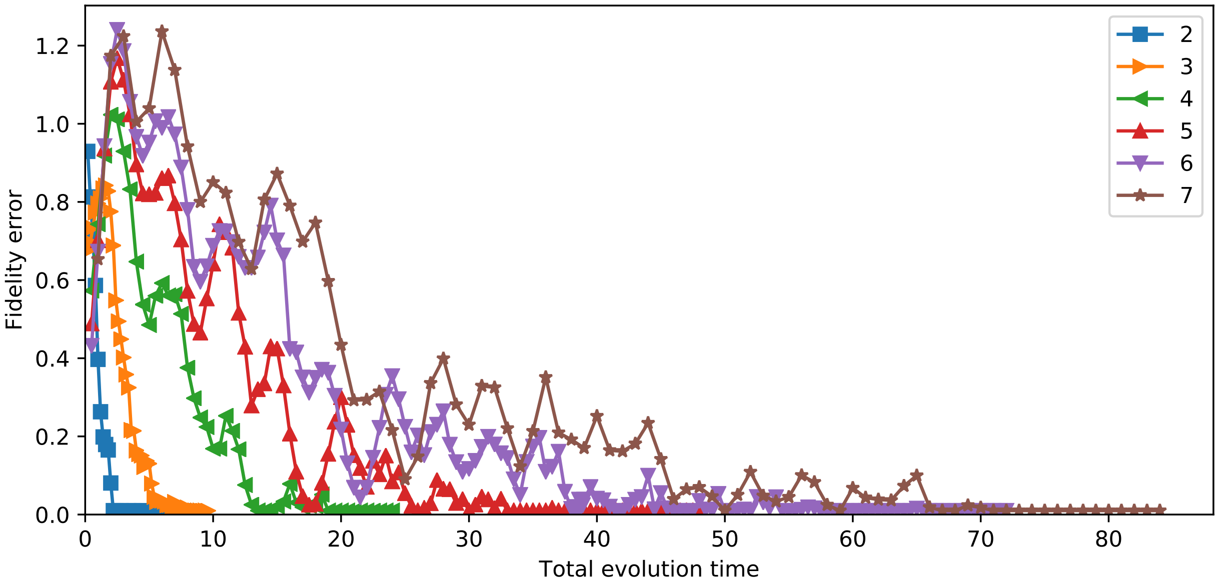
Specific parameters for the drift Hamiltonian were $\omega = 2$, $g_1 = 0.3$, $g_2 = 0.2$. These were chosen to fulfil the positive definite requirement. Although $g_1 = g_2$ would be more typical for a coupled QHO system, they were chosen different as momentum coupling does not occur in classical oscillator systems, and therefore having this makes the chains in some way more ‘quantum’. The control Hamiltonian parameters were set at $w_1 = 1$ and $\chi = 0.5$. These are somewhat arbitrary, except in the context of of the amplitude bounds, which were set at $[-10, 10]$. The number of timeslots for the piecewise constant controls was set based on the number of oscillators at $2n^2$. This is based on theory from [78] and was confirmed by experimentation. As the number of degrees of freedom in this system increases, so number of parameters in the optimisation method must also, unless there are symmetries that effectively reduce the degrees of freedom. The cost of computing the time evolution grows exponentially with the number of oscillators. Combined with the quadratic growth in optimisation parameters this made it infeasible to extend the numerical results beyond 7 oscillators.

The plots in Fig. 3.7 show the results of the attempt to find the minimum evolution time $T$ required to perform the symplectic operation, known as the *minimum gate time*. There are two measures of this. The first, $T_{g_1}$, is a specific time that is the smallest $T$ found to be sufficient to effect the operation, the second, $T_{g_\mathrm{A}}$, is a lower bound on $T$ such that the operation can be reliably achieved for all $T > T_{g_\mathrm{A}}$. The trends of both $T_{g_1}$ and $T_{g_\mathrm{A}}$ are shown in Fig. 3.8 for the $S_{\sqrt{\mathrm{swap}}}$ target. There is nothing in the plot to suggest a scaling of any worse than linear for either of the minimum gate time measures. Similar results found for spin chains were reported in [143].

To verify that $T_{g_1}$ and $T_{g_\mathrm{A}}$ for $S_{\sqrt{\mathrm{swap}}}$ are not significantly different for other operations, due to some symmetry or other feature, the minimum gate time optimisations were repeated with a randomly generated symplectic target. The target was kept the same for each $T$, but there was no relation between the targets for different chain lengths. Plots Fig. 3.7 a.) & b.) show a very similar trend for $T$ with the minimum fidelity error for the different chain lengths. The trends for minimum gate time with the random symplectic target are not plotted, because the targets for each $n$ are not related.

The fidelity error Eq. (3.18) is unbounded from above, so it is no surprise that $\mathcal{E} > 1$ is seen in Fig. 3.7. The minimum infidelities (or unreachable targets) for random symplectic targets are typically higher than those for the $S_{\sqrt{\mathrm{swap}}}$ target in the same size system, and this increases with $n$.

(a) $S_{\sqrt{\text{swap}}}$ target



(b) Random symplectic target

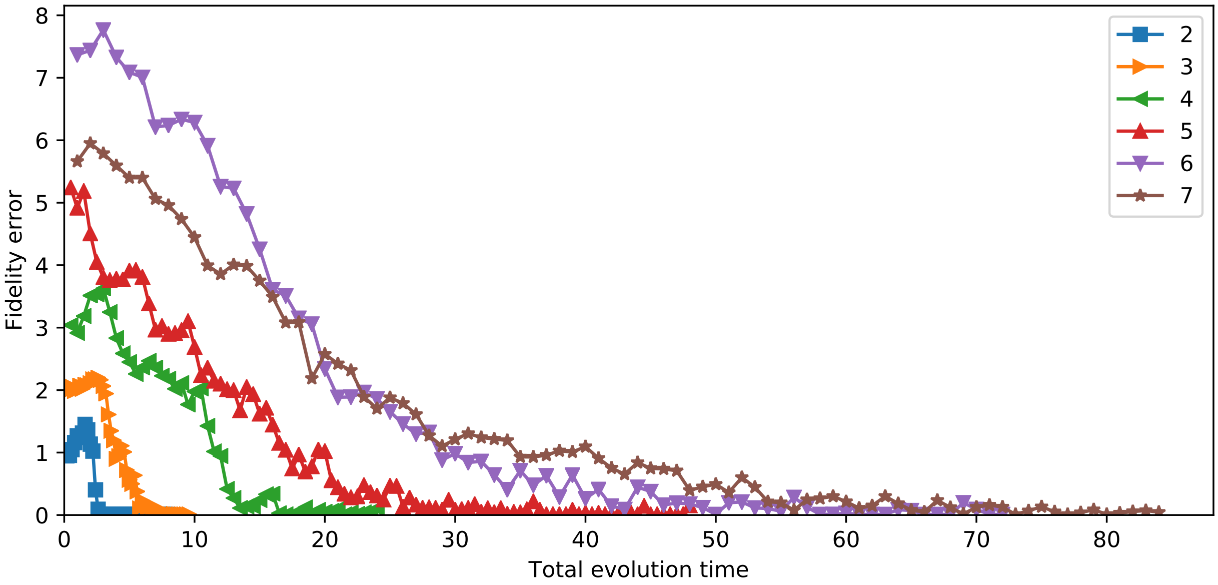


**Figure 3.7:** The minimum fidelity error is plotted for different evolutions times $T$. This is repeated for different chain lengths $n = \{1, 2, 3, 4, 5, 6, 7\}$. In (a) the target is the square root of the boson swap, which is described in the main text. In (b) the target is some random symplectic operation of corresponding $n$.

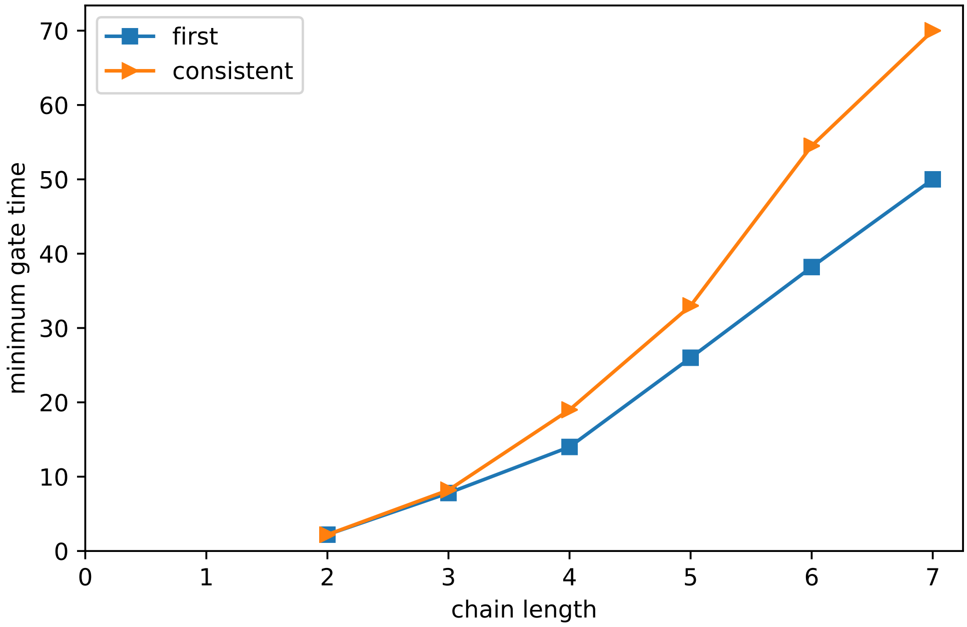


**Figure 3.8:** Plot showing the trend in minimum total evolution time $T$ found to be required to effect the operation. The blue squares indicate $T_{g_1}$, the smallest time such that the operation can be achieved to within a fidelity error of $\mathcal{E} < 0.01$. The orange triangles indicate $T_{g_{\mathrm{A}}}$, the time that $\mathcal{E} < 0.02$ consistently for any greater $T$.

The elements of the $S_{\sqrt{\mathrm{swap}}}$ target matrix are predominantly zeros, which is not necessarily true for a random symplectic. The fidelity error at $t = 0, S = \mathbb{I}$ will likely be much higher for the random symplectic, and when the target in not reachable for $t = T$, this is likely to remain that way.

## 3.8 Conclusion

The Lie algebra rank criterion is necessary and sufficient for controllability on compact Lie groups. It is not sufficient for the non-compact symplectic group, which is used to represent the operators on Gaussian quantum systems. The sufficiency of neutrality for controllability of the generators has long been known. Specifically for the single-mode case this has been equated with the geometrical interpretation of elliptic. Although previous work has proved the necessity of elliptic generators in an isomorphic group, this study clearly illustrates this for the symplectic group in a single mode. A region of the symplectic space has been shown to be unreachable when the control system cannot access elliptic generators. Relating this to physical systems, such as in quantum-optics, passive, or energy-preserving operations have been shown impossible to implement for unstable single-mode systems. This is due to squeezing, which is not energy preserving, being unavoidable while attempting to drive the passive operations, which are phase rotations.

Further characterisation of the reachable set was made using numerical methods. Control optimisation finds reached points in the symplectic operator space that are symmetrical about the identity in coordinates provided by singular value decomposition. This hints at the potential for finding an analytical solution for the set that includes only reachable points.

Systems with hyperbolic generators, having operations accessible through control, exist in several experimental set-ups, both optical and mechanical. They have potential for beating decoherence times due to exponential speed-ups – for example [144], where such operations are proposed to achieve this aim in the context of superconducting quantum magnetomechanics. The Hamiltonians generating these are referred to as 'repulsive potentials'.

The two-mode reachable set investigation perhaps raises more questions than it answers. Certainly it remains open ended, with many avenues worthy of further investigation. The narrowness of the reachable set for the specific unstable example, particularly in the angle parameters, is most interesting. It is not immediately clear what the physical relevance of these angles are. Perhaps the mean values could give some indication. It may be possible to find other unstable examples that can access wider ranges of rotations. Why is there an upper limit on $\delta z_1 + \delta z_2$ for the shifted targets? As the controlled generator is a two single-mode squeezer, and the control function is unbounded, then there should be no upper limit squeezing. It may be interesting to plot the trajectory of the operator evolution in terms of these generators.

The inevitable squeezing conjecture appears to have been disproved with many counter examples. However, no examples were found where the overall squeezing was zero. The eigenvalue signatures perhaps give some clue as to how a counter-example might be constructed. Only one counter-example is necessary to disprove the conjecture about passives being inaccessible with non-neutral systems. It would be sufficient to find one set of generators, combined with a control function, where any squeezing is completely reversed in the final evolution,

The neutrality condition can be used to select valid parameters for a simulation or experiment to ensure controllability. The reachable set investigation showed that it not necessary to exclude control parameters that give non-neutral generators, as although they do not provide full controllability, they do provide a set of potentially useful operations. Access to some neutral generators is sufficient, as demonstrated in the coupled oscillator chain simulations, where what intuitively seems like very limited control over the system, but shown to be sufficient through the Lie algebra rank criterion and the neutrality condition, was found capable of achieving challenging control objectives. Although only a limited set of data was possible, no evidence for scaling of gate times with number of modes worse than linear was found. Numerical quantum control optimisation was clearly demonstrated to be a useful tool in investigating both the reachable set of operations, and the scaling of minimum gate times with system size, for continuous variable quantum systems.

# Chapter 4

# Upgrading quantum simulators to computers

The results presented in this chapter were produced as part of a collaborative project with Benjamin Dive, Florian Mintert & Daniel Burgarth, they have been published in Ref. [145]. The application of the local estimator fidelity to *in-situ* optimisation and its implementation as a QuTiP control 'custom fidelity' were completed by B. Dive. The thesis author developed the algorithm for locating the fidelity precision threshold, ran the control optimisation simulations and performed analysis for the required updates and fidelity precision scaling, and ran the simulations and performed the analysis for the topological and interaction type comparison.

## 4.1 Introduction

From the description of a gate type quantum computer given in Subsection 1.5.4, building a digital quantum computer sounds like an achievable engineering feat – one just needs a sufficiently large register of coherent qubits, that can be addressed individually for the preparation and readout of quantum states, and upon which two different single qubit and one two qubit gates can be effected. There are wide range of physical systems that can act as qubits, along with methods for controlling their state evolution. However, effecting two-qubit gates to high-levels of fidelity in itself is hard, and finding a physical platform which provides *scalability* in all of these ingredients is proving elusive. And so the objective of building a device that can demonstrate quantum supremacy, let alone a digital quantum computer that perform useful tasks in science and the widely society, remains unachieved.

There are many challenges that still remain to be overcome in the development of a scalable quantum computer, as discussed in Subsection 1.5.4. Quantum simulators are a rapidly developing technology, and today there are devices with hundreds of interacting coherent qubits. If single and two qubit gates can be effected on these devices, initial states can be reliably set, and final states be efficiently read out, then they will be able to execute quantum algorithms, and hence act as a quantum computer.

For a given quantum simulator, if there is sufficient control over individual qubits and the necessary interactions, then the many-body quantum system is fully controllable, meaning that any unitary evolution can be effected upon it, and this clearly includes gates on single and pairs of qubits. Assuming this, how can the controls which perform these gates be determined? If an accurate model could be built, then the quantum control optimisation methods introduced in Section 1.6 could be used to determine the control parameters required to perform a gate on a specific qubit or qubits. There are two insurmountable challenges with this. Firstly, it is impossible to model the system accurately enough that gate fidelities would be sufficient. Secondly, the computing power required to calculate the evolution of an $n$ qubit system scales exponentially with $n$, making it practically impossible to simulate any system than could perform a useful quantum computation. In this chapter a scheme is presented whereby the controls that perform gates can be determined *in situ*, that is within the device itself.

Suitable quantum simulators for the scheme exist, or at least are close to being realised. These are discussed in Section 4.2. The idea of optimising controls *in situ* is not new, in fact it has a long history that is outlined in Section 4.3. The crux of the scheme is that it must scale favourably with system size, that is no worse than polynomially in $n$. Central to the scheme is a method for determining how closely the controls effect the unitary that performs the desired gate(s), which must not only effect the gate(s) on the selected qubit(s), but must also act trivially on the rest of the system, and hence is a full system unitary operation. A fidelity measure for this purpose that does not scale with the number qubits is presented in Section 4.5. The control optimisation method itself

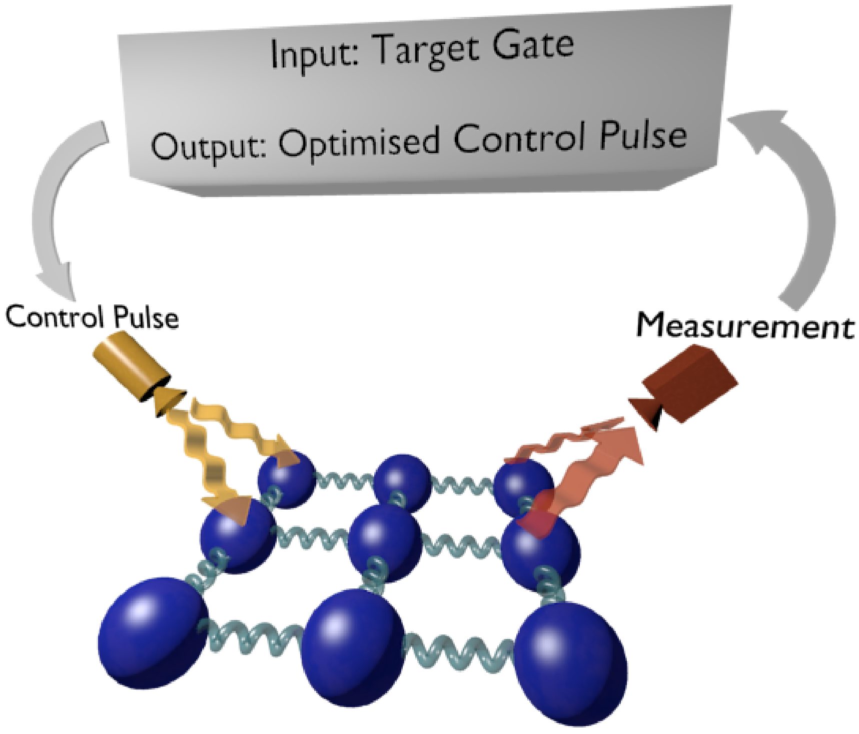


**Figure 4.1:** Illustration showing *in-situ* gate optimisation. A classical computer iteratively optimises a control pulse based on measurements performed to estimate the fidelity. [graphic produced by B. Dive]

must also scale acceptably. The numerical study in Section 4.6 demonstrates this by simulating the scheme up to the limits of (available) classical computing power, which is for systems of up to 9 qubits, in which it is shown to scale polynomially for systems with Ising interactions in various topologies. Thus demonstrating the useful application of numerical control optimisation in developing quantum information processing technology.

### 4.1.1 The Choi-Jamiolkowski isomorphism

The Choi state will be used in the definition of the gate fidelity, hence it is defined here. There is a one-to-one correspondence between CPT maps (introduced in Section 1.2.3) and quantum states on the doubled space of the system $\mathcal{H}_{\mathrm{CJ}} = \mathcal{H}_S \otimes \mathcal{H}_S$. This is the Choi-Jamiolkowski isomorphism, defined as

$$\rho_M = (M \otimes \mathcal{I})|\Phi\rangle\langle\Phi| \qquad \text{, where} \quad |\Phi\rangle = \frac{1}{\sqrt{d}} \sum_i^d |i\rangle \otimes |i\rangle \tag{4.1}$$

where the $|i\rangle$ form a basis of $\mathcal{H}_{\mathrm{CJ}}$, the definition of $|\Phi\rangle$ is that of the maximally entangled state, and $\mathcal{I}$ is the identity map. The Choi state $\rho_M$ is therefore the CPT map $M$ acting on one half of the maximally entangled state of the doubled space, and it encodes all properties of the map $M$. If $M$ is a unitary channel, then $\rho_M$ will be a pure state.

## 4.2 Quantum simulator for computing

An overview of analogue quantum simulators, and the wide range of applications envisioned for them, is given in Section 1.5.3. As mentioned there, one of the key applications is in the investigation of magnetism at a quantum level. Some of the simulators being developed for this purpose also happen to be suitable for implementing the scheme for upgrading them to quantum computers that is detailed in this chapter. The specific technologies and their applicability are discussed below.

Ions trapped by electromagnetic potentials have long been employed in the experimental study of quantum behaviour due to the ability to isolate individual particles [146]. Qubit states are encoded in the internal electronic states. Laser cooling is used to limit this to the ground and first energetic state, giving a two level system. Laser pulses are also used to drive transitions for state read out, and couple ions through motional states, making two-qubit gates possible [43]. It is a mature technology and quantum entanglement has been demonstrated in over 200 qubits in ion trap simulators [45].

The flexibility of generating artificial lattice structures using ultra-cold atoms makes them particularly suited to many applications in condensed matter physics, such as superfluidity and superconductivity [147]. There have been recent advances in generating two dimensional arrays to simulate coupled spin-½ systems [46]. Although this study only reported success with arrays of up to 30 spins, which is insufficient for a useful quantum computation, it is a rapidly developing field. It could provide a platform for the *in-situ* gate optimisation scheme as a proof of principle, as it provides the controllability and measurement criteria.

The nuclear spins of atoms within molecules have long been used as qubits. Their use in quantum simulation has been proposed for some time [148], with some successful realisations

[149]. The scaling of using NMR with liquid phase molecules has always been an issue, as beyond tens of qubits it becomes difficult to address specific nuclei within the molecule. However, the nuclear spins of impurities in diamond hold great potential for quantum simulation and computing, as their locked positions in the lattice can be used to address individual nuclei. Architectures for large scale simulators based on this idea have been proposed [150].

Superconducting circuit qubits, with states as direction of current flow or harmonic states of oscillators, and couplings via capacitance or photonic cavities, offer one a the most scalable platforms due to the well established lithographic techniques that are used to fabricate them. Tuning for specific parameters can be achieved simply through geometry. Successful digital quantum simulations have been made with these devices [151]. There are challenges in individual qubit control and addressing with larger scale devices [58].

The simulation of quantum spins with Ising type interactions is one of the prime applications for these systems, as this is the model used to investigate magnetism at a quantum level. Simulators configured in this way are called *Ising machines*. There examples with trapped ions [42], cold atoms [46] and optical processors [152]. The numerical study demonstrates that these systems can be upgraded to act as quantum computers, provided only that full control and addressing of individual qubits is available.

## 4.3 *In-situ* control optimisation

The idea of using the system itself to solve Schrödinger's equation in a control optimisation procedure was demonstrated in 1992 [153], in which a diatomic molecule was driven into a specific, thermally inaccessible state through laser pulses. The shape of the pulse is provided by the controlling computer, which runs a genetic algorithm to iteratively improve on the fitness based on input from the measurement device. The main motivation for using control *in situ* in this case was difficultly in accurately characterising the Hamiltonian of molecule under control. The Rabitz group have continued to developed this method of "learning control" and have used it to explore many aspects of quantum control [154, 155, 156, 157, 158, 159], with a particular focus on controlling chemical reactions.

In 2014 classical and *in-situ* control optimisation were combined using a method that the authors call adaptation by hybrid optimal control (Ad-HOC) [160]. A control pulse is optimised on a classical computer, running the Nelder-Mead algorithm, based on a model of the two coupled superconducting qubit (transmon) system with estimated parameters. Again, not being able to measure these parameters to sufficient precision is cited as the motivation for a hybrid approach [161]. The partially optimised pulse is used as the initial condition for a closed-loop *in-situ* optimisation. The authors report that fidelities are improved by an order of magnitude by the second stage optimisation.

In the same year the Santa Barbara group were using a process called optimised randomized benchmarking for immediate tune-up (ORBIT) to improve gate fidelities [5]. Analytically derived gate control pulses are optimised *in situ* for transmon qubits, again using the Nelder-Mead algorithm. Randomised benchmarking [162, 163] is used to determine the gate fidelity. Improvements in single and two-qubit gate fidelities are reported, with the main result being an increase in fidelity from 0.984 to 0.993 for the two-qubit entangling controlled-phase gate. This brings it within the limit required for fault tolerant computing [61]. Recently the Santa Barbara group has also used *in-situ* optimisation [6] to remove qubit drift errors in arrays of 9 transmon qubits.

The efficiency of the hybrid classical-*in-situ* optimisation has been demonstrated experimentally in a 7 qubit NMR type quantum simulator [164]. Their aim was to overcome the challenge highlight in this chapter's introduction: the cost of computing evolution under the Schrödinger equation for large scale many-body quantum systems, due to the exponential scaling of the Hilbert space. They found success in preparing certain states, but recognised the limitations of NMR simulators in terms or scalability and speed.

So far *in-situ* optimisation has not been used to determine gate controls in large scale quantum simulators. A significant challenge is the difficulty of measuring the gate fidelity in many-body systems, for which a solution is proposed by this scheme.

## 4.4 The Scheme

The proposed scheme is to take a quantum simulator, comprising locally coupled qubits, and optimise the gates required for a quantum computing algorithm *in situ*, that is within the device itself. As discussed in previous sections, this has already been achieved in two-qubit devices. This scheme would allow this to be extended into larger systems, making a scalable quantum computer. To study the scalability of the scheme, a model is assumed for $n$ qubits, coupled by some interaction, such that all are connected either directly or indirectly. Full control over individual qubits is assumed, meaning all single qubit unitary operations are possible. The decoherence time is assumed to be long enough such that it is insignificant relative to the gate error threshold in the gate operation timescale. Preparation and readout in a full basis set is also assumed for individual qubits. It is almost certain that the scheme could operate with less stringent assumptions. What is not assumed is any control over the interactions, nor any prior understanding of the effect of control parameters on individual qubits.

The control assumptions are sufficient to implement any gate in the quantum simulator (see Section 1.4), given sufficient couplings between qubits [165, 166, 167]. However, as no prior understanding of how the control parameters will drive the system is assumed, then it is impossible to design a control pulse functions to perform the gates. It is possible though to use the control optimisation methods described in Section 1.6 to find the effective pulse parameters in the control landscape, so long as a cost function can be found to evaluate the success of the control attempt. A method for measuring the fidelity of the evolution with respect to the target operation is given in the next section.

The steps for performing the gate optimisation are outlined in Fig. 4.2. It is assumed that there is a discrete set of parameters to drive the system over a given time $T$ and that the gate can be effected within this time. In reality there would be some understanding of how these parameters may drive the system. As here no assumption is made about this, then any selection is a guess, and hence random is as good as any. The system is prepared in some state, evolved under the control parameters, and the state at time $T$ read out, these steps (which are described in detail in Section 4.5) are repeated until the gate fidelity is estimated. Depending on the optimisation algorithm in use, they may also be repeated to estimate the gradient with respect to each of the control param-

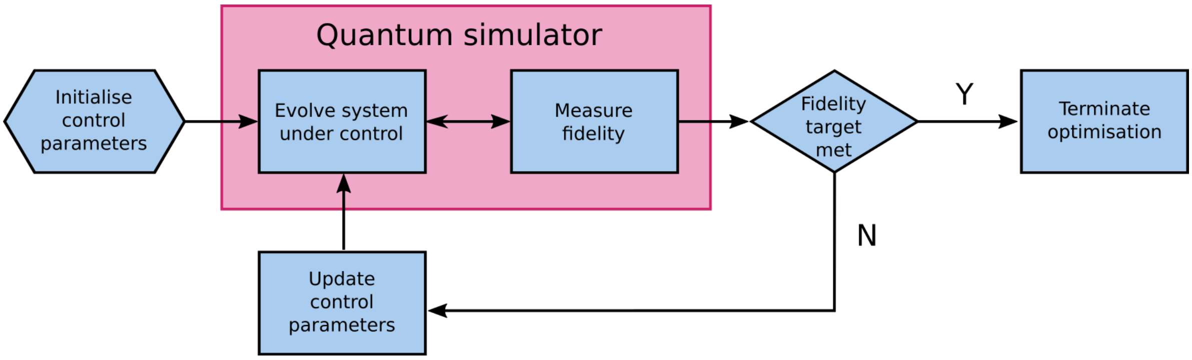

Quantum simulator
Initialise control parameters
Evolve system under control
Measure fidelity
Fidelity target met
Y
N
Terminate optimisation
Update control parameters

**Figure 4.2:** Outline of the process for optimising quantum gates *in situ* using the quantum simulator where the gate will be used for quantum computing algorithms. The process is identical to that performed in a classical computer simulation, except that the system evolution step is completed by the simulator and the measure of the fidelity would take a number of repeats of this propagation, which would be controlled by a classical computer along with the other steps.

eters. The fidelity is then compared with the target threshold, which it is not expected to meet until after numerous repetitions. If it meets the threshold then the optimisation process is complete, as a set of controls have been found that effect the desired gate, otherwise the optimisation algorithm determines a new set of control parameters and the process repeats.

The system evolution step would be impossible to simulate classically for a quantum simulator capable of executing a quantum algorithm that outperforms a classical counterpart, as the Hamiltonian matrix to integrate would be exponentially large in $n$. It would also be pointless, as the Hamiltonian parameters could not be determined to sufficient precision for required gate fidelities to be met, as the couplings between qubits cannot be measured. The *in-situ* evolution step solves both of these problems, as the simulator evolves naturally and is a perfect model of itself. The scheme assumes that although the system and control Hamiltonians are not known, that they do not change over time, so that the controls determined through the optimisation process will perform the gate when applied again. Therefore, using this scheme, controls can be determined that will perform all the gate combinations required for some quantum algorithm, and hence can be run in sequence to implement the said algorithm.

As the control assumptions state that all single qubit operations are possible, then it is trivial to see that controls could be found to perform single qubit gates. Control sequences for these would need to be optimised in the quantum simulator. However, as described in Section 1.5.4, two qubit entangling gates are also required for universal quantum computing. The C-NOT gate is the canonical two qubit gate for quantum computing. In physical systems these are much more difficult to effect, as they rely upon the interactions between qubits, and typically have much higher gate errors. Although it is not necessary to be able to effect these between all pairs of qubits in order to perform algorithms, the option leads to shorter circuits and hence more efficient algorithms. Therefore the focus of the numerical study is to demonstrate that this scheme is scalable with $n$ in optimising C-NOT gates between arbitrary qubits in the simulator; those which are directly and indirectly connected in various topologies.

## 4.5 Local gate fidelity

Evaluating the fidelity of the evolved operator with respect to the target operation is easy in a classical simulation of a gate optimisation, as the full evolution operator has to be computed. The computation of this matrix is the expensive part (both in CPU time and memory requirements), due to the exponential growth of the matrix with system size, hence part of the motivation for the *in-situ* scheme. The evolution operator is not directly accessible in the *in-situ* optimisation, only the states before and after the gate implementation attempt are available through measurement. The standard method to calculate the operation effected by the controls is through a form of process tomography called *certification*, which requires a repeated procedure of preparing the system in some state, evolving the system (under controls), and performing measurements to estimate the evolved state. The number of repetitions scales as $O(d^2) = O(2^{2n})$, that is exponentially with system size [168], making it infeasible in large systems such as the quantum simulators proposed for implementing this scheme. So it is not possible to measure the actual gate fidelity efficiently, but it is possible to measure a tight bound efficiently, as described in this section.

The gate fidelity $\mathcal{F}_{\text{PSU}}$ introduced in Section 1.6.1, whether applied to dynamic maps or unitary operators as targets, requires the full evolution for comparison with the target. The gate fidelity applied to some CPT map $M$, which includes any dissipative or decohering effects, and the target unitary evolution operator $U$ can be defined in terms of Choi states as

$$\mathcal{F}(M,U) = \langle\psi|\rho_M|\psi\rangle\,, \tag{4.2}$$

where $|\psi\rangle = (U \otimes \mathbb{I})|\Phi\rangle$ and $\rho_M$ is the Choi state of $M$ defined in Eq. (4.1) – where the maximally entangled state $|\Phi\rangle$ is also clearly defined. The fidelity is upper bounded at $1$, with $\mathcal{F} = 1$ when $M(\cdot) = U(\cdot)U^\dagger$, and is lower bounded at $0$. If the evolved map is also unitary, that is $M(\cdot) = V(\cdot)V^\dagger$, then it simplifies down to the square of the unitary gate fidelity $F(V,U) = \frac{1}{d^2}|\operatorname{Tr} V^\dagger U|^2$.

As the gate acts only on a small number of qubits, maximum two if the only multi-qubit gate is the C-NOT, then the target is not some arbitrary unitary, but rather one that can be expressed as the tensor product of gates acting on subsystems, which in this case are single or pairs of qubits. That is $U = \bigotimes U_i$, where $i$ is a subsystem index. These unitaries could be any combination of single or (non-overlapping) multi qubit gates. This study focuses on product gates with a single C-NOT on two qubits and the identity on the rest, that is

$$U = \bigotimes U_i = \text{C-NOT}_1 \otimes \mathbb{I}_2 \otimes \mathbb{I}_3 \otimes \ldots\,. \tag{4.3}$$

Note that the subsystem index $1$ here does not imply the first pair of qubits, the subsystem could be any pair of qubits, adjacent in terms of topology or otherwise. So long as the target can be decomposed as Eq. (4.3), then the fidelity is bounded by the *local estimator* $\mathcal{F}_{\text{LE}}$ as

$$\mathcal{F}(M,U) \geq \mathcal{F}_{\text{LE}}(M,U) = 1 - \sum_i (1 - \mathcal{F}(M_i, U_i)) \tag{4.4}$$

where $M_i(\rho_i) = M(\rho_i \bigotimes_{j\neq i} \frac{1}{d_j}\mathbb{I}_j)$ is the reduced dynamical map acting on subsystem $i$ and the other subsystems are initialised in the maximally mixed state. The proof of $\mathcal{F}_{\text{LE}}$ being a lower bound on the gate fidelity is based on methods from [169] and is given in Appendix D.

Whereas to estimate the true gate fidelity requires certification over the entire system, leading to the exponential scaling, the local estimator requires certification only over the small subsystems. That is the one and two qubits that the gates (or identity) act upon. These do not scale with the overall size of the system. The cost of measuring each subsystem is $O({d_i}^2)$. To calculate

$\mathcal{F}_{\text{LE}}$ each of the subsystem fidelities $\mathcal{F}(M_i, U_i)$ must measured, for which the $M_i$ need to be reconstructed. One method would be to initialise the state of one subsystem in a pure state, and the other subsystems in the maximally mixed state, then allow the system to evolve under the controls, then measure the specific subsystem in one basis. This would need to be repeated on the subsystem until every combination of preparation and measurements has been completed for a complete basis set, which is $O({d_i}^2)$ repetitions. As this would need to be repeated for each subsystem it leads to an overall scaling for the number of measurements $N_{\text{meas}} = O(\sum_i {d_i}^2) = O(n)$.

Although linear scaling for fidelity measurement is already a big improvement over exponential, intuitively, measuring only one subsystem for each application of the controls seems inefficient. Indeed, as the maximally mixed state is a random mixture of pure states, the fidelity of each subsystem can be measured concurrently by preparing each of them in a random pure basis state. This removes the scaling with system size, as the time required for the measurements scales as with that for the largest subsystem. So with this measurement procedure $N_{\text{meas}} = O(1)$.

Another important consideration for the local estimator to the fidelity is how tightly it bounds the true gate fidelity and how it scales with $n$ relative to the gate fidelity. These are important in determining how effective it will be when used in an optimisation algorithm. The minimum for the true gate fidelity is zero, whereas it is $1 - n$ for the local estimator. This is unlikely to hinder any optimisation algorithm, as minimising the infidelity $1 - \mathcal{F}_{\text{LE}}$ would be the typical approach, and hence there would just be a landscape scaling factor of $n$. The local estimator convergences on the true gate fidelity in the limit that $F(M, U) \to 1$, which is helpful, as this region is of primary interest when optimising gates. Less easy to show is that $\mathcal{F}_{\text{LE}}(M, U)$ increases monotonically as as $\mathcal{F}(M, U)$ does. This was investigated numerically for variety of quantum simulator models and gate targets. The two fidelity measures are plotted at each update stage of the optimisation algorithm in Fig. 4.3. In all four system scenarios the local estimator can be seen to track the true gate fidelity monotonically as the infidelity decreases, and the convergence of the two is shown to be rapid. Thus showing that the local estimator to the infidelity operates well as a cost function for minimising the gate infidelity in an optimisation algorithm.

The scaling of the local estimator bound with system size is also an important consideration. Due to it being a sum over the subsystem fidelities, the difference is expected to scale linearly

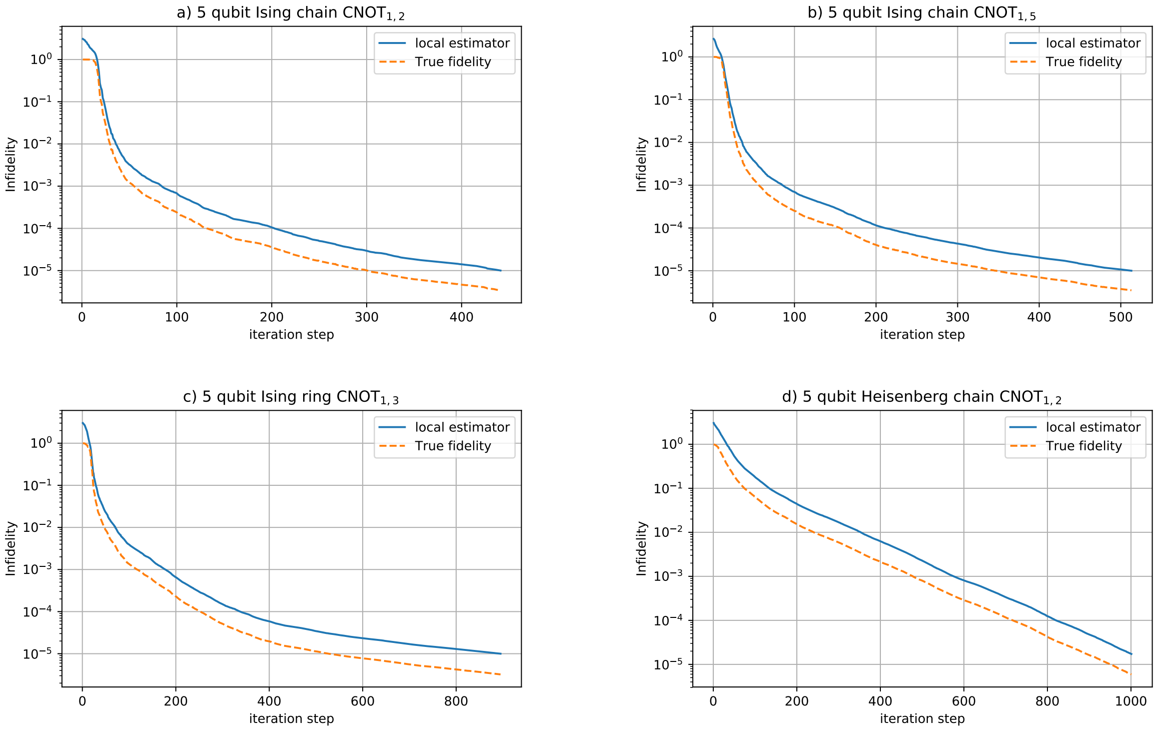


**Figure 4.3: Comparison of the local estimator with the gate fidelity during the optimisation procedure.** The gate infidelity $1 - \mathcal{F}(M, U)$ and the estimator to it $1 - \mathcal{F}_{\mathrm{LE}}(M, U)$ are plotted at each iteration step of one gate optimisation procedure. This is shown for different scenarios, all of which are with models of $5$ qubit systems, with the target operation being the C-NOT gate on two qubits and the identity of the rest. The scenarios are a.) an Ising chain with the C-NOT on the first two qubits, b.) an Ising chain with the C-NOT on the first and last qubits, c.) an Ising ring with the C-NOT on the first and third qubits, and d.) a Heisenberg chain with the C-NOT on the first two qubits. The topologies and interactions are described in Section 4.6. In all scenarios the local estimator can be seen to track and converge on the gate fidelity, demonstrating that maximising the local estimator also maximises the gate fidelity.

with $n$. This was investigated numerically for targets on up to $14$ qubits. A target unitary $U$ was formed for the C-NOT on the first two qubits and identity on the rest as per Eq. (4.3). Another operator for comparison was generated from this by taking a randomly generated Hamiltonian $H_\epsilon$ and propagating $U$ to give $U_\epsilon = e^{-\mathrm{i}H_\epsilon}U$. $H_\epsilon$ was generated by a Gaussian distribution and normalised such that $||H_\epsilon||_2 = 0.1$. The gate fidelity averaged over different $H_\epsilon$ remained at $99.5\%$ for $3$ to $14$ qubits. The local estimator fidelity dropped linearly by less than $0.2\%$ per qubit, confirming the intuition based on the form of $\mathcal{F}_{\mathrm{LE}}$ in Eq. (4.4).

## 4.6 Numerical investigation

In the previous section it was shown that a measurement scheme for determining a bound on the fidelity of a gate operation exists that does not scale with system size. In order to show that the entire *in-situ* scheme can be implemented efficiently it is necessary to show that other aspects scale no worse than polynomially with system size. Specifically, these are the time that is required to implement the gate, and time required to optimise the controls that effect the operation. Also the precision to which the fidelity must be measured must also not scale exponentially with system size. It has not been possible to find analytical expressions for these, and so a numerical approach is used. For systems up a certain size, the entire scheme, as outlined in Fig. 4.2, can be simulated on a classical computer.

One of the main drivers behind this scheme is that the system evolution computation step on a classical computers is known to scale exponentially with the number of qubits, both in terms of processing time and memory requirements. Due to this second limitation it was not possible to go beyond 9 qubits on the high performance computing cluster that was available. For the fidelity precision sensitivity trials, many optimisation attempts are required for reasonable statistics, and so it was only possible to get up to 7 qubits in that part of the investigation. Although the local estimator fidelity is exponentially more efficient to calculate than the true gate fidelity in an experimental setup, in a classical simulation it is actually significantly more expensive to compute. As the $M_i$ in Eq. (4.4) need to be extracted from the full evolved map or unitary (which scales with $2^n$) using the partial trace, which is an expensive operation in itself, calculating the fidelities at each update step is a significant proportion of the processing effort.

The average time required to find a set of controls that perform the desired gate using the optimisation process is given by:

$$T_{\text{total}} = T_{\text{run}}\ N_{\text{runs}}/p_{\text{succ}}\ . \tag{4.5}$$

$T_{\text{run}}$ is time taken for one run of the gate control sequence on the quantum device. It can be further broken down as $T_{\text{run}} = T_{\text{init}} + T_{\text{gate}} + T_{\text{meas}}$, with $T_{\text{init}}$ being the time required to prepare the system in its initial state, $T_{\text{gate}}$ the time set for evolving the system under the controls, and $T_{\text{meas}}$ the time needed for performing the measurement on the system. $T_{\text{init}}$ and $T_{\text{meas}}$ are specific to the system and are assumed to be constant. Not all gate optimisation attempts may be successful, $p_{\text{succ}}$ is the probability that one is. $N_{\text{runs}}$ is the average number of times that the gate must be attempted in a optimisation procedure, that is, reach the target fidelity or terminate for another reason. $N_{\text{runs}}$ can be decomposed as

$$N_{\text{runs}} = N_{\text{meas}}\ N_{\text{prec}}\ N_{\text{fids}}\ N_{\text{upds}}\,. \tag{4.6}$$

Here $N_{\text{meas}}$ is the number of times the gate controls are applied with different state preparations and measurement basis in order to calculate the local estimator gate fidelity. In the previous section this was shown to be $O(1)$. $N_{\text{prec}}$ is the number of times the fidelity must be measured to statistically reach the required precision. $N_{\text{fids}}$ is the number of gate fidelities needed for the optimisation algorithm, which depends on the algorithm in use. This would be $1$ per update for a gradient free method, such as Nelder-Mead. For gradient based methods running *in situ* a finite difference method must be used to calculate gradients (as per Eq. (1.58), hence $N_{\text{fids}} = N_{\text{cp}} + 1$, as an additional fidelity would be required for each of the $N_{\text{cp}}$ control parameters. $N_{\text{cp}}$ depends on the control model and control function parametrisation scheme. $N_{\text{upds}}$ is the number of control parameter updates, corresponding to the number of times the loop in Fig. 4.2 is executed. Again this is dependent on the choice of optimisation algorithm, for gradient based methods this would typically be at least an order of magnitude less than a gradient free method, as is discussed in Section 1.6.2.

The optimisation algorithm used in this numerical investigation is the GRAPE type in QuTiP introduced in Subsection 1.6.4. A ‘custom fidelity’ code module was developed specifically for the study. This allows for a cost function of $\mathcal{E}_{\text{LE}} = 1 - \mathcal{F}_{\text{LE}}$ to be used. The module also calculates the analytic gradients with respect to each of the control parameters. This scheme does not propose any method for calculating analytic gradients *in situ*, however the purpose of this numerical investigation was to see how the control optimisation scales with the number of qubits $n$, rather than produce the most faithful simulation of the scheme. Firstly to see how $N_{\text{upds}}$ scales with $n$. Using analytic gradients reduces the processing time to simulate the scheme by a factor of $N_{\text{fids}}$, which is a known value and hence can be included later, but it would not be expected to have much effect on $N_{\text{upds}}$, which is the unknown in Eq. (4.6).

For a GRAPE optimisation a piecewise constant parametrisation must be used for the control pulses. Hence the gate time $T_{\text{gate}}$ is split into $N_{\text{ts}}$ equal duration timeslots in which the control amplitude will be constant. For the quantum simulator to be classically simulated the Hamiltonians for the internal system interactions and the control pulse interactions must be defined – for the actual implementation of the *in-situ* optimisation scheme this would not be necessary. The full Hamiltonian is of the form

$$H = \sum_{j} \left[ u_x^j(t)\sigma_x^j + u_y^j(t)\sigma_y^j \right] + \sum_{<j,k>} g_{j,k} H^{j,k} \,, \tag{4.7}$$

where $\sigma_x^j$ and $\sigma_y^j$ are the Pauli operators acting on the $j^{\text{th}}$ qubit, modulated by the control functions $u_x^j(t)$ and $u_y^j(t)$. Any two independent rotations would give full control over individual qubits, they need not be the same for each qubit, but were chosen to be so in this study for convenience. Two controls per qubit means that with this model $N_{\text{ctrls}} = 2n$. The second summation is over all qubit pairs and represents the interactions between qubits. The form of $H^{j,k}$ determines the interaction type. The couplings $g_{j,k}$ define the topology; details of these are given as the results are presented. So, with two independent control Hamiltonians per qubit and each function described by $N_{\text{ts}}$ amplitudes, this gives a total number of control parameters $N_{\text{cp}} = N_{\text{ctrls}}\ N_{\text{ts}} = 2nN_{\text{ts}}$. A summary of the parameters used to describe the scheme are given in Table 4.1. The initial guess pulse for all controls used to produce the results presented comprised uniformly random amplitudes in the range $[-1, 1]$.

Fidelity, gradients and control parameters are purely classical values and therefore computations can be done classically. These scale at worst polynomially with system size and hence can be completed efficiently on a classical computer. They would also be done off-line, that is the control pulses required to perform gates would be determined before any quantum algorithm was run, not as part of it. As discussed in Section 1.5.5, research into fully quantum algorithms for multi-variable optimisation is a high-priority topic, and hence there is future potential for the computer to optimise its own gates. However, the focus here is a scheme that can be implemented with technology that currently exists.

| Parameter | Description |
|---|---|
| $T_{\text{gate}}$ | Evolution time for the gate |
| $N_{\text{ts}}$ | Number of timeslots for control pulse |
| $A_{\text{num}}$ | Accuracy of fidelity measurements |
| $\mathcal{F}_{\text{targ}}$ | Target fidelity for the desired gate |
| $N_{\text{runs}}$ | Number of (#) runs in total |
| $N_{\text{meas}}$ | # different preparation and measurement pairs |
| $N_{\text{prec}}$ | # repeats for required fidelity precision |
| $N_{\text{fids}}$ | # different fidelities to update controls |
| $N_{\text{upds}}$ | # control updates needed |
| $p_{\text{succ}}$ | probability of optimisation success |
| $N_{\text{ctrls}}$ | # independent controls on system |
| $N_{\text{cp}}$ | Total # of control parameters for the optimisation |

**Table 4.1:** Summary of parameters defined in the text. Those above the line are set by the experimentalist, and are input parameters for the *in situ* optimisation. Those below the line are a derived from parameters of the system and or the optimisation algorithm. They are used to quantify the efficiency, and hence assess the scalability of the scheme.

### 4.6.1 Topologies and interactions

In order to demonstrate that the scheme is applicable to the wide range of different simulation platforms that are available, a range of different topologies and interaction types were used. The *chain* topology is where all qubits have two nearest neighbour interactions, except the first and last which have only one interaction. The *ring* topology is the same except the first and last qubits have an interaction with each other. The *star* has a central qubit that interacts with all other qubits, which have no other interactions. The *fully connected* systems have interactions between all qubits. The *Ising* type interactions are $\sigma_z \otimes \sigma_z$ couplings, the *Heisenberg* type are $\sigma_x \otimes \sigma_x, \sigma_y \otimes \sigma_y, \sigma_z \otimes \sigma_z$ couplings. Unless otherwise stated the results presented are for equal coupling constants for all interactions.

Consistently the study found finding controls through optimisation was much easier for systems with Ising type interactions. For chains, rings and stars it was found that the time needed to effect the C-NOT gate does not need to scale with system size, neither did the number of timeslots. $T_{\text{gate}} = \pi, N_{\text{ts}} = 12$ was found to be sufficient for all systems up to $9$ qubits for these systems when the gate is performed on neighbouring qubits, and $T_{\text{gate}} = 4\pi, N_{\text{ts}} = 48$ sufficient for finding control pulses to perform C-NOT gates on any two qubits in the system. For Heisenberg systems the required gate time scaled badly with system size, potentially even exponentially with $n$.

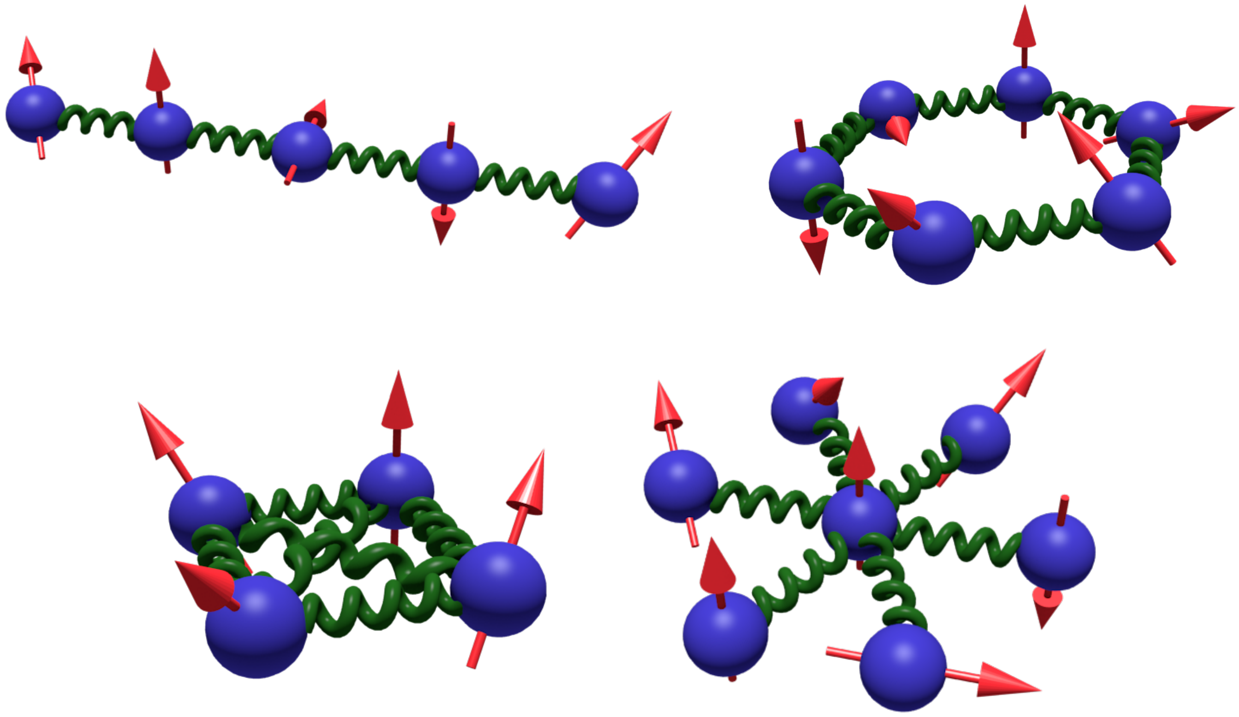


**Figure 4.4: Illustrations of quantum simulator model topologies.** The quantum spins are shown as blue spheres with the red arrows indicating their state vector in the Bloch sphere. The green springs indicate the interactions between qubits. The topologies shown are top left: chain; top right: ring; bottom left: fully connected; bottom right: star.

Consistently it was found that a timeslot duration $\Delta t \approx \pi/12$ provided best performance by the optimisation algorithm, and so $N_{\text{ts}}$ would also scale badly. Similarly difficult to find control pulses for were fully connected systems with Ising interactions of unequal coupling strength (chosen randomly). A summary of some of the results for different topologies and interactions are given in Fig. 4.2. The gate times and number of timeslots shown are the minimum for which a control pulse can be reliably found through optimisation. These results suggest that for $5$ qubit systems the scheme will work for a range of different simulators. However, to give an indication of relative effort, the 50 optimisations on the Ising chain cost under one hour of processing time, whereas the 50 optimisations on the Heisenberg chain cost over 36 days. For some reason the Heisenberg chain required a greater $T_{\text{gate}}$ than any of the topology and interaction type combinations; there are no clear intuition on why this should be.

Both the ease for which Ising system controls could be found and the difficulty in Heisenberg systems are surprising results. For Ising systems it was expected that number of control parameters

| Topology | Coupling | $T_{\text{gate}}$ | $N_{\text{ts}}$ | $N_{\text{upds}}$ |
|---|---|---|---|---|
| chain | Ising | $\pi$ | 12 | 60 |
| star | Ising | $\pi$ | 12 | 214 |
| fully connected | Ising | $12\pi$ | 160 | 295 |
| chain | Heisenberg | $16\pi$ | 160 | 585 |
| star | Heisenberg | $12\pi$ | 160 | 1043 |
| fully connected | Heisenberg | $12\pi$ | 160 | 881 |

**Table 4.2:** The average number of control updates $N_{\text{upds}}$ for $5$ qubit systems in a range of different topologies is shown. The topologies and interactions are described in the main text. The target operation is the C-NOT gate on the first two qubits (in the star case this includes the central qubit) and the identity on the rest. These simulations were done with full numerical precision. As the Hamiltonian used is that of Eq. (4.7) then $N_{\text{ctrl}} = 10$, and hence the total number of control parameters $N_{\text{cp}} = 10N_{\text{ts}}$. The target fidelity in all cases is $\mathcal{F}_{\text{targ}} = 0.999$. For each scenario 50 out of 50 attempts were successful in finding controls.

would scale polynomially with system size, possibly as $O(n^2)$, as discussed in Subsection 1.6.2. The intuition is that symmetries in the system allow for the gate to be effected, while state of the surrounding qubits remains unaffected, with relative easy. It has been shown that it is possible to do fast two qubit gates on $n$-qubit systems with any entangling interactions [166], and therefore the difficulties in finding control pulses in that work for Heisenberg interacting systems should be resolvable through optimisation techniques. The same problems were found with the standard gate infidelity used as a cost function, and so the problem is not related to the local estimator fidelity. It is possible that different parametrisations of the pulse would allow the dynamical decoupling operations that effectively remove the additional interactions of Heisenberg to be found by the optimisation process. Many different combinations of $N_{\text{ts}}$ and $T_{\text{gate}}$ were trialled in order to try and draw these out, however GRAPE does not appear to find these with standard piecewise constant parametrisations, and having a large number of timeslots greatly hinders to performance of the algorithm. Many variations of interaction types were also tested, including full $\sigma_z$ weak $\sigma_x$ and or $\sigma_y$ couplings, starting with weak $\sigma_x$ and or $\sigma_y$ couplings on qubit pairs. However, these all behaved more similarly to the full Heisenberg interactions than Ising.

The aim of many quantum simulator developments is to make Ising machines, and hence it is sufficient for now to show that the optimisation scheme works well for systems of different topologies with Ising interactions. However, it is clearly desirable to be able to apply the scheme to a wider range of systems, and also to better understand the reasons for the relative difficulties in finding effective pulses, and so this remains a very interesting area for future study in which numerical control optimisation is a valuable tool.

### 4.6.2 Optimisation effort scaling

The previous section described how neither the gate time nor number of timeslots scales with system size for Ising chains and rings. Consequently it was possible to investigate the scaling of the number of control updates required with system size using these system models with up to $9$ qubits, simulated on a classical computer. Rings, as well as chains, were chosen for study because there is no potential for boundary effects with a ring. As has been previously mentioned, quantum circuits are more efficient when two qubit gates can be implemented on arbitrary qubits, and so checking that the optimisation for the C-NOT gate can be effected anywhere on the simulator is also important. The results of simulated gate optimisations for $3$ to $9$ qubit systems are shown in Fig. 4.5 for Ising chains and rings. The trend indicated for the number of control updates required $N_{\text{upds}}$ is no worse than linear in the number qubits $n$. A sub-linear fit for these could even be feasible. A conservative linear extrapolation suggests that around 700 (400) updates would be required to optimise for C-NOT gates in 50 qubit Ising chains (rings), and 1500 (900) for 100 qubits. The effort for each update is dependent on hardware type (and $n$, as discussed later), and so converting this into any time units is non-trivial. However, it is not an untypically large number of updates in control optimisation, as is seen throughout this thesis. Reaching 50 qubits is considered an important milestone for quantum supremacy, as discussed in Section 1.5.4.

$N_{\text{Upds}}$ is slightly higher for the rings than the chains of equivalent size, although the gradient is lower, suggesting the two may converge for larger systems, where the boundary conditions at the ends of the chain will become less significant, which is further evidence for a sub-linear relationship. The $3$ qubit systems would not seem to fit any obvious model. The number of updates required is strikingly low compared to the other systems. One intuition for this is that the symmetry of the system allows for the qubit not part of the C-NOT gate to remain disentangled. In both topologies the number of updates required for randomly separated and neighbouring gate qubits are statistically indistinguishable. This is a somewhat surprising but welcome result, as

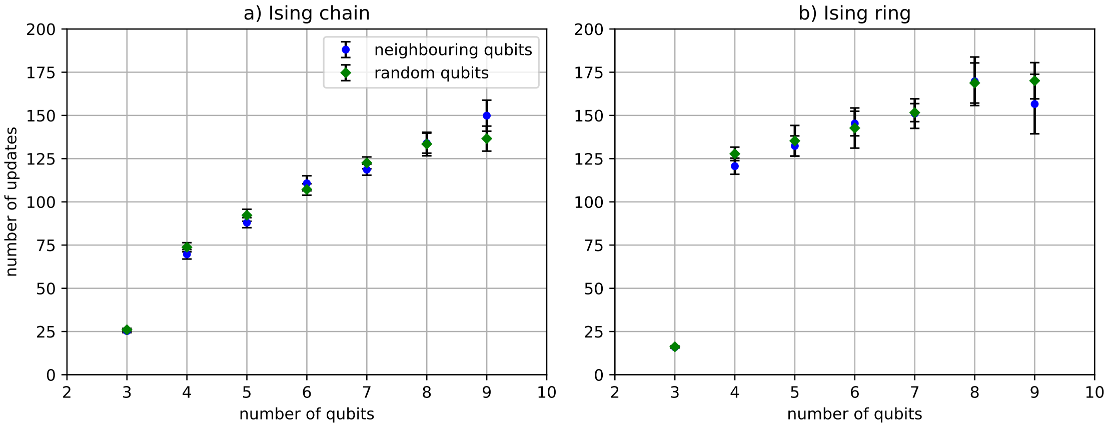


**Figure 4.5: Number of control pulse updates needed to optimise for C-NOT gate**. Plots showing the average number of control updates required for the optimisation procedure to find controls that perform the C-NOT gate on either neighbouring or randomly separated qubits. In both cases the first gate qubit is randomly selected. Plots are shown for a.) Ising chains of length $3$ to $9$ qubits, b.) Ising rings of $3$ to $9$ qubits. The topologies are described in more detail in the main text. In all cases a gate time of $T_{\text{gate}} = 4\pi$ was used, and the control parameters are the piecewise constant amplitudes of pulses of $N_{\text{ts}} = 48$ equal length timeslots. A local gate fidelity of $\mathcal{F}_{\text{LE}} \geq 0.999$ was achieved for each optimisation. Each data point represents repeated optimisations: 100 for $n < 8$; 96 for $n = 8$; 30 for $n = 9$. The number of successful optimisations is $p_{\text{succ}} > 90\%$ in all cases. The error bars shown are twice the standard error. As there is no obvious model to fit the number of updates required to the size of the system, no best-fit is shown.

intuitively one would expect the gate would become more difficult to effect with greater separation between the gate qubits. It is possible that the same symmetries that make the gate time and number of control parameters scale so favourably also allow for this ease of entanglement across many qubit interactions.

### 4.6.3 Fidelity precision scaling

Another important consideration is the sensitivity of the system to measurement noise. The number of repetitions required to achieve the necessary precision $N_{\text{prec}}$ needs to scale favourably with system size for the *in-situ* optimisation scheme to be viable. Concerns over the effort required to achieve high levels of precision in fidelity measures in many-body systems have been raised in other work, and so an important part of this investigation is to test how this required precision scales with system size. The effects of measurement noise are modelled in the simulations via a parameter $A_{\text{num}}$. In the simulations reported on in this subsection, each local fidelity measure is rounded to the precision of $A_{\text{num}}$. The process *in situ* would be to repeatedly measure the outcome until this level of precision has been achieved statistically. The central limit theorem states that the number of repetitions required scales quadratically with the desired precision, that is

$$N_{\text{prec}} \propto A_{\text{num}}^{-2} \tag{4.8}$$

If the precision is too coarse, that is $A_{\text{num}}$ is set too high, then the potential inaccuracy of the fidelity measurement hinders the optimisation algorithm, and can lead early termination of the algorithm due to its inability to determine a search direction. Setting the precision too fine would mean unnecessarily many repeated measurements. So by determining the value of $A_{\text{num}}$ that leads to a fixed probability of success $p_{\text{succ}}$ for the algorithm optimises $N_{\text{prec}}$, and hence the whole *in-situ* optimisation process. The method by which this was done in the classical simulations is described in Section 2.7, and would be applicable *in situ*. Values for $p_{\text{succ}} \approx 0.5$ are optimal.

The required $A_{\text{num}}$ is expected to scale as $O(\mathcal{E}_{\text{LE}}/n)$ as, to achieve gate infidelities below $\mathcal{E}_{\text{LE}}$, the fidelity would need to be measured to an accuracy of $O(\mathcal{E}_{\text{LE}})$. As the gate infidelity is calculated from a sum of subsystem infidelities, it is conjectured that each subsystem fidelity would need to be measured to an accuracy of $O(\mathcal{E}_{\text{LE}}/n)$. In the classical simulations the $A_{\text{num}}$ is applied at subsystem level, that is each of the infidelities $1 - \mathcal{F}(M_i, U_i)$ are rounded to the precision of $A_{\text{num}}$.

Plot a.) in Fig. 4.6 shows the fidelity precision required to achieve $p_{\text{succ}} = 0.5$. A number of different fidelity targets are used in order to provide more evidence for the scaling conjecture. Due to the many repeats of the optimisation procedure required to find the $p_{\text{succ}}$ threshold for each of the fidelity target scenarios, only the Ising chain system was studied, and it was only possible to work with up to 7 qubits. Separated gate qubits closest to the centre of the chain are used in an attempt show a result not dependent on a direct gate qubit interaction and to avoid boundary condition effects as far as possible within the simulation limitations. The $c/n$ fit curves for $A_{\text{num}}$ shows strong evidence for the conjectured $O(\mathcal{E}_{\text{LE}}/n)$ scaling for the required fidelity precision. The third plot in the figure provides some evidence for the linear relationship of the required $A_{\text{num}}$ with $\mathcal{E}_{\text{LE}}$. The emphasis here is on the scaling with $n$, which the fits shown in plot a.) strongly

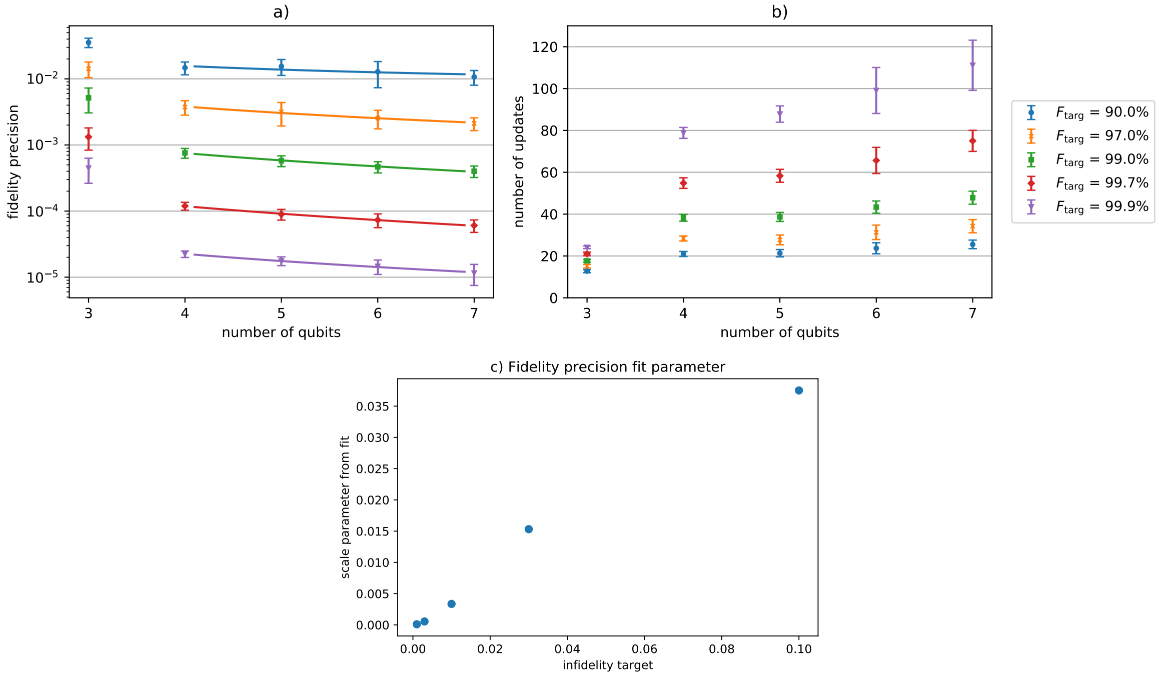


**Figure 4.6: Fidelity precision required for optimal scheme performance.** The required fidelity precision $A_{\mathrm{num}}$ for achieving a proportion $p_{\mathrm{succ}} = 0.5$ of successful optimisations for C-NOT gates is found for Ising chains of length $n$ being 3 to 7 qubits. This is repeated for different fidelity targets. Plots showing a.) the fidelity precision $A_{\mathrm{num}}$ and b.) the number of control updates $N_{\mathrm{upds}}$ for successful optimisations for each of the scenarios. The details of the topology and interactions of the Ising chains are given in the main text. Here the C-NOT gate optimisation is attempted between two qubits in the middle of the chain, separated by one other qubit. The target is the identity for the other qubits. The gate time is $T_{\mathrm{gate}} = 4\pi$, split into $N_{\mathrm{ts}} = 48$ timeslots for the control parametrisation. The fidelity accuracy for the $p_{\mathrm{succ}} = 0.5$ is estimated using an interpolation of $p_{\mathrm{succ}}$ values for a range of $A_{\mathrm{num}}$, with between 25 and 45 points used in the interpolation. Each of these points are the average over a number of repetitions: 200 for $n = 3, 4$; 100 for $n = 5$; 50 for $n = 6, 7$. The error in values is derived from the variance in the repetitions and the uncertainty in the interpolation method. In plot a.) the error bars show five times the standard error, in plot b.) twice the standard error. The fit line in plot a.) is based on a $c/n$ model. The $n = 3$ data are excluded from the fit for reasons discussed in the main text. Plot c.) shows the fit parameter $c$ for the different targets, shown as the infidelity as this is how the model is described in the main text.

support $O(1/n)$. The $n = 3$ data again suggests that the 3 qubit chain is a special case, it is the least significant in terms of a scaling relationship where the upper end is most of interest, and it is therefore excluded from the fits.

The average number of control updates $N_{\text{upds}}$ for the successful optimisations are shown in plot b.) of Fig. 4.6. The values for $\mathcal{F}_{\text{LE}} = 0.999$ can be compared with those in Fig. 4.5. Although the systems differ in terms of gate position and separation, this was seen to be not significant in the resulting $N_{\text{upds}}$ in Fig. 4.5. The $N_{\text{upds}}$ is typically lower in these simulations where $A_{\text{num}}$ is found to give a $p_{\text{succ}} = 0.5$ than those where the full numerical precision is used. This may at first seem a surprising result, as the optimisation algorithm is hindered by inaccurate fidelity measurements. However, $N_{\text{upds}}$ is only averaged for *successful* optimisations, and $50\%$ failed. It is likely that the $50\%$ that failed would have taken a greater number of updates than average if optimised using machine precision, assuming that the initial parameters were in a less favourable position on the fidelity landscape. The main point to note from Fig. 4.6 plot b.) is that, with the numerical accuracy of the fidelity measurement modelling included, the simulations still indicate a no worse than linear scaling of $N_{\text{upds}}$ with $n$.

### 4.6.4 Full *in-situ* scheme scaling

Numerical simulations of Ising systems have been used to estimate the scaling of those parameters used to calculate the time required to optimise a gate *in situ*, which are neither apparent from the configuration of the system itself, nor have elsewhere been analytically derived. It is therefore now possible to derive an estimate from the scaling of the *in-situ* gate optimisation scheme with the number of qubits $n$ in the system as follows. $N_{\text{meas}}$ for the local estimator fidelity was shown to be $O(1)$ in Section 4.5. $A_{\text{num}}$ was estimated to be $O(1/n)$ in simulations of Ising chains, which combined with Eq. (4.8) gives $N_{\text{prec}} = O(n^2)$. As the number of timeslots $N_{\text{ts}}$ remained constant for the Ising system simulations, the number of control parameters only scaled with $n$, and hence $N_{\text{fids}} = O(n)$ for the gradient based optimisation algorithm used. Simulations of optimisations in Ising chains and rings showed evidence for no worse than $N_{\text{upds}} = O(n)$. Substituting these into Eq. (4.6) gives $N_{\text{runs}} = O(n^4)$.

It is assumed that $T_{\text{run}} = O(T_{\text{gate}})$, and the simulations using Ising systems found that $T_{\text{gate}} =$

$O(1)$. The value $p_{\text{succ}} = 0.5$ is fixed in the estimate of $N_{\text{prec}}$ scaling. Hence the total optimisation time from Eq. (4.5) scales only with the number of runs required, that is $T_{\text{total}} = O(N_{\text{runs}}) = O(n^4)$. So based on numerical simulation, for systems corresponding to the Ising chain or ring model, the scaling for *in-situ* gate optimisation is polynomial.

## 4.7 Conclusion

The polynomial scaling of the *in-situ* gate optimisation scheme demonstrated numerically for some models of quantum simulators provides encouraging evidence for its viability. The limits of classical simulation of quantum system dynamical evolution prohibits extension of this kind of verification into the scales that would demonstrate more conclusively that it could be used to turn quantum simulators into quantum computers. This was known from the outset, and is one the prime motivations behind proposing the scheme. The linear (or potentially sub-linear) fit for the required number of control optimisation updates comes purely from numerical data, and there is no analytical model to back this up, hence other fits are possible. Experimental verification would be required to further validate the scalability scheme.

The results presented here are encouraging enough to suggest that high fidelity two qubit entangling gates could be effected in some types of quantum simulator with up to 50 qubits, which is potentially sufficient to demonstrate quantum supremacy. Specifically, simulators with linearly arranged qubits where the interactions are of the Ising type between nearest neighbours, although the scheme has been shown to work in models of various topologies and interaction, these were found easiest to optimise classically. The simulations of optimisations performed for this study could be repeated with parameters that more faithfully model a quantum simulator proposed for a trial of the scheme. More limited controls for instance could potentially still allow for sequences that perform the necessary gate operations – not all unitaries are necessary, just the single and two qubit gates required for a quantum algorithm. The state-of-the-art quantum simulators provide sufficient qubits and controls to extend the results of this study far beyond what is possible to simulate classically.

The GRAPE algorithm was chosen for the classical simulation of this scheme due to the efficiency that the gradients and system evolution can be calculated, and the resulting faster convergence speed. This chapter has again demonstrated that numerical control optimisation, and in particular GRAPE, to be a valuable tool for investigating theoretical scalability in quantum information process systems. GRAPE relies on the piecewise constant parametrisation of control pulses to gain these advantages. When operating the scheme in an experimental setting the efficiency of piecewise constant system evolution calculation does not exceed any other parametrisation scheme, and the overhead of the additional fidelity measures required to approximate the gradients may outweigh the reduced number of algorithm iterations. It is also likely that other parametrisation schemes would be easier to implement physically, such is one of main advantages of the CRAB algorithm (see Section 1.6.6). Therefore it is possible that using a simplex optimisation method such as the Nelder-Mead one used in CRAB may result in overall fewer control updates.

A stochastic simplex optimisation algorithm is compared with Nelder-Mead in another study into *in-situ* gate optimisation [170]. It was found to out-perform Nelder-Mead due to its resilience to measurement and control noise. Machine learning based optimisation algorithms (discussed in the thesis conclusion) may further increase the optimisation efficiency when applied to *in-situ* gate optimisation, by reducing the number of updates or being able to operate at lower fidelity precision. However, with the theory and experiment in Ref. [164] the authors demonstrate the ability for the fidelity and its gradient to be calculated *in-situ*, which could potentially reduce the overhead of computing gradients, making gradient based algorithms most suited to this application.

Alternative methods for parametrising the control functions to specifically promote the dynamical decoupling pulses that would negate or transform the unhelpful system interactions remains an interesting direction for further study. It would certainly have been a much more satisfying result to show that the scheme scales well in all models of many-body systems, regardless topology or interaction type. Certainly a lot of effort went into trying to optimise gates in Heisenberg chains of increasing length, which would be a promising first step. But, as a reminder, this does not stem from the *in-situ* scheme – it is encountered generally in gate optimisation simulations.

# Chapter 5

# Non-Markovian dynamics

The research for this chapter was completed in collaboration with Neill Lambert, RIKEN, Japan and Daniel Burgarth. Neill Lambert developed the original HEOM solver code, provided the pure-dephasing derivation and example code, dressed picture Lindbladian example code, and provided example code for the dynamical decoupling. Daniel Burgarth suggested the dynamical decoupling control objective. The thesis author improved upon the HEOM solver code, and implemented this and the Lindblad master equation solver in a control optimisation setting, for which more detail is given in Chapter 2. The thesis author developed the code, and ran all the simulations and performed the analysis, for the results presented in this chapter.

## 5.1 Introduction

The theory of open quantum systems, for which some background is given in Section 1.2.3, is used to predict the dynamics of quantum systems when couplings to the environment are taken into consideration. Clearly, in quantum control theory, which looks to predict how quantum systems can be manipulated given some external influences, the accuracy of the predictions depends on the validity of the model used to describe the environmental interactions. There is a great body of work investigating control in open quantum systems, including some in depth studies [171, 172]. The majority of this work assumes that the Born-Markov approximation is valid in the scenarios considered, as this allows dynamics to be computed through solving a linear differential equation.

This chapter investigates a method of solving open systems dynamics that does not make that assumption, and compares a control problem where the assumption cannot be used. Numerical control optimisation is used determine optimal dynamical decoupling pulses, which can be used to protect quantum information from decoherence, and potentially also negate unhelpful many-body system interactions, such as those discussed in Chapter 4.

The hierarchical equations of motion (HEOM) method for solving the open systems dynamics is described as exact, because it allows for correlations in the bath that persist for durations longer than the characteristic timescales of the system, that is, it does not make the Born-Markov approximation. As the name suggests, the dynamics are modelled through a hierarchy of coupled differential equations, for which the solution is only exact with an infinite depth to the hierarchy. The precision is determined by the truncation, which must be balanced against computational efficiency, making its application in control problems less trivial.

The HEOM has a long history, with the method originally proposed in 1989 by Tanimura and Kubo [173], and extended to a more general spectral density in Ref. [174]. The model was generalised to low temperatures by Ishizaki and Tanimura [175]. The model achieved its highest fame when Ishizaki proposed its use in describing the speculated evidence of quantum coherence in a biological system [176]. The proposed evidence of quantum coherence in the light harvesting complexes of a photosynthetic bacteria, the Fenna-Matthews-Olson (FMO) complex, aroused much excitement [177]. No other single model was able express the coherent, wavelike and incoherent, hopping behaviours of the biological system. The question of if and how the coherence plays a role the function of the light harvesting complexes attracted much research attention, with Ref. [177] attracting over 2600 citations, including this recent review [178]. Clearly, any research with potential practical applications in the harvesting of light energy causes great excitement, as these applications include artificial photosynthesis – the direction conversion of water and carbon dioxide to sugars, and hence clean fuels made from carbon captured from the atmosphere.

For the model to be used efficiently in control optimisation, it must be valid for a time-dependent system Hamiltonian, as driving the system towards some control objective typically results in a time-dependent system Hamiltonian of the form of Eq. (1.22). The derivations in the early papers are not specific about the time dependence, but they do not specify that it is assumed constant either. The derivation in Ref. [179] explicitly claims validity for systems driven by arbitrary time-dependent fields. This paper also describes how to quantify the precision of the method for particular truncation and some method for correction, as well as generalising the spectral density parametrisation. Some work with control in non-Markovian processes has been undertaken using the HEOM for computing the dynamics [180], where the authors report that driving the system leads to increased non-Markovianity in the bath response.

The HEOM need to be solved in order to compute the evolution of the reduced density matrix of the system. The original HEOM solver code in QuTiP by Neill Lambert was based on code developed specifically to compare the reaction coordinate model [181, 182]. The QuTiP solver code was reworked and documented by the author for this study. The computational efficiency of the hierarchy construction was improved ten-fold, which is highly beneficial in control optimisation, as the evolution needs to be computed many times, for reasons explained in Section 1.6. To the authors best knowledge, this is the only open source generic solver of the HEOM – there is another specific to the light-harvesting complexes [183]. The implementation is based on the Drude-Lorentz spectral density for a single bath. Development of a more general spectral density and support of multiple baths is well progressed, but not officially released at the time of this writing. The integration with control optimisation is publicly available in the `qoc` branch of QuTiP [113], but also not yet officially released.

This chapter starts by discussing Markovianty in quantum processes, and how non-Markovianty might manifest in the evolution of quantum systems. It then goes on to describe how the dynamics can be computed by solving the Lindblad master equation. For this there are many textbook resources, including Ref. [22], and these manuscripts provide good summary Refs. [171, 172], particularly in relation to control. The hierarchical equations of motion are described briefly in section Section 5.3, with a focus on their applicability in control optimisation. The specific model for the spectral density that is used in the QuTiP HEOM solver is outlined in Section 5.4. Section 5.5 validates the HEOM solver and investigates suitable hierarchy parameters using an analytical model for comparison. The Lindblad master equation and HEOM dynamics, as solved by numerical simulation, are compared in Section 5.6. A comparison by example of the two methods for computing driven system dynamics is given in Section 5.7, based on a control optimisation simulation. The optimisation dynamical decoupling pulse scheme with dynamics computed using the HEOM is described in Section 5.8. The results are summarised and discussed in Section 5.9, which also gives some outlook.

## 5.2 Markovianity and quantum processes

Markovianity was first introduced as a way to distinguish between different types of classical stochastic process, with those classed as Markov having the property that their behaviour going forward is not linked to past events [184]. The generalisation in to quantum systems needs to take into account that measurement affects the state of the system [185], and so the strict definitions from the study of classical systems cannot be directly applied in the quantum regime. One definition is based on the divisibility of the map

$$M(t_3, t_1) = M(t_3, t_2)M(t_2, t_1) \qquad \forall \quad t_3 \geq t_2 \geq t_1 \geq 0, \;, \tag{5.1}$$

as it states that the evolution from time $t_2$ to $t_3$ is independent of the evolution from time $t_1$ to $t_2$ for any choice of durations. This is therefore often used as a definition for Markovianity in quantum systems [185].

There are some issues with this definition of Markovianity in quantum systems. Firstly that it is binary, as a quantum process by this definition is either Markovian or not. Secondly, but related, is that it is difficult to measure in any actual quantum process. It would be useful to have some scale of non-Markovianity that could be measured in an experiment. A measure proposed in Ref. [186] is based on the premise that the trace distance (introduced as a measure of infidelity in Eq. (1.48)) between any two states should decrease monotonically during evolution under Markovian dynamics. The trace difference is interpreted as a measure of distinguishably, and when this decreases, information has been lost to the environment. Therefore a quantification of any increase in the trace distance during the evolution could be used as a measure of non-Markovianity, one of which is presented in Ref. [186]. Information decreasing monotonically is shown to be equivalent to divisibility of the process in Ref. [187].

There are many descriptions of non-Markovianity and how it can be measured, some of these are reviewed in Ref. [188], and it remains an active area of research. Typically it is assumed that the Markovianity or non-Markovianity of the dynamics is determined only by the inherent the system-environment interactions, however in Ref. [189] it is shown that in some cases it is also dependent on the how the system is driven, which is an important consideration when designing controls.

### 5.2.1 Markovian quantum dynamics

The map $M(t_2, t_1)$ acts on the space of density operators in a way that it will give the quantum state of a system at some time $t_2$ based on its state at $t_1$. That is, at $t_1$ it is in state $\rho_1$ and at $t_2$ it is in state $\rho_2 = M[\rho_1]$. For understanding how the system state may evolve in the duration $t_1$ to $t_2$, and hence model how systems may interact, it would be helpful to have a differential equation to describe it. A linear differential equation of form similar to Schrödinger's equation Eq. (1.4) would be ideal, that is something of the form

$$\frac{\mathrm{d}}{\mathrm{d}t}\rho(t) = L[\rho(t)]\,, \tag{5.2}$$

which would therefore have solutions of the form

$$M(t) = \mathrm{e}^{Lt}\,, \tag{5.3}$$

thus making $M$ a linear operator. In order for such an equation as Eq. (5.2) with solution Eq. (5.3) to hold the generator $L$ must be independent of the system state, either directly or indirectly. This implies that the generator of the dynamics has no memory the past, which in the context of an open system means the dynamics of the system are assumed not to affect the bath in any way that alters its effect on this system at some future time, which is often referred to as no *back-action*. This assumption relies on the interaction between the system and bath being sufficiently weak to ensure this, which is known as the Born approximation. As discussed above, this implies that the map itself is infinitely divisible, which when considering a differential equation like Eq. (5.2) implies

$$M = {M_j}^n \quad \forall\, n \in \mathbb{N}\,, \tag{5.4}$$

where $M_j$ is some other map that when applied $n$ times is equivalent to $M$. This divisibility requirement is known as the Markov approximation, because the operator generating forward evolution does not depend on previous time steps. Because one depends on the other, they are often referred to as the combined Born-Markov approximation.

For closed systems where $\rho_1$ and $\rho_2$ would be pure states, the form of $L$ is that from the Liouville-von Neumann equation Eq. (1.7), that is $L = -\mathrm{i}[H,\rho]$, so it is clear that when there are no dissipative or dephasing effects that $L$ must reduce to this. The map must be able to result in mixed states from pure states, and hence there must be additional components to the generator that allow this. It turns out that generators $L$ exist that satisfy the requirement that $M$ is a CPT map, so long as they are of the form

$$L[\rho] = -\mathrm{i}[H,\rho] + \sum_j \gamma_j \left[V_j \rho V_j^\dagger - \tfrac{1}{2}(V_j^\dagger V_j \rho + \rho V_j^\dagger V_j)\right] \tag{5.5}$$

where $H$ is a Hamiltonian, $\gamma_j > 0$, and the $V_j$ are referred to as Lindblad, jump or collapse operators. The dissipative effects are contained within the summation, and the $\gamma_j$ determine the strength of the interactions that result in the dissipation. The $L$ generator itself is referred to as the Lindbladian [190, 191, 192].

In this form the Lindbladian is constant in time, but it need not be so. So long as the time-dependence does not come from the system or bath state, but is independent, then the Hamiltonian and Lindblad operators may be time-dependent, that is $H \equiv H(t), \gamma_j \equiv \gamma_j(t), V_k \equiv V_k(t)$ in Eq. (5.5). When the Lindbladian is time-dependent, the map is described as *infinitesimally* divisible, that is

$$M = \prod_{j=1}^{n} M_j \quad \forall\, n \in \mathbb{N}\,, \tag{5.6}$$

where the $M_j$ are CPT maps, but no longer need be identical.

In the general case then, for the time-dependent Lindbladian generator $L(t)$, the state dynamics are described by the Gorini-Kossakowski-Sudarshan-Lindblad master equation

$$\frac{\mathrm{d}}{\mathrm{d}t}\rho(t) = L(t)[\rho(t)]\,. \tag{5.7}$$

For which a derivation can be found in Ref. [21]. To solve the Lindblad master equation Eq. (5.7) numerically it is convenient to take the *superoperator* form of the Lindbladian [193]

$$\mathcal{L} = \mathrm{i}\left(H^{\mathsf{T}} \otimes \mathbb{I} - \mathbb{I} \otimes H\right) + \sum_j \gamma_j \left[\bar{V} \otimes V - \tfrac{1}{2}\left(\mathbb{I} \otimes V^{\dagger}V + V^{\mathsf{T}}\bar{V} \otimes \mathbb{I}\right)\right] . \tag{5.8}$$

This acts on a column stacked vector form of the density matrix, that is

$$|\rho\rangle := \left(\rho_{1,1}, \rho_{2,1}, \dots, \rho_{d,1}, \dots, \rho_{1,d}, \dots, \rho_{d,d}\right)^{\mathsf{T}} , \tag{5.9}$$

such that

$$\mathcal{L}|\rho\rangle \equiv L[\rho] . \tag{5.10}$$

Here the action of $\mathcal{L}$ on $|\rho\rangle$ is matrix multiplication. $|\rho\rangle$ is a vector of length $d^2$, and hence $\mathcal{L}$ a $d^2 \times d^2$ matrix. This allows an equivalent differential

$$\frac{\mathrm{d}}{\mathrm{d}t}|\rho(t)\rangle = \mathcal{L}|\rho(t)\rangle . \tag{5.11}$$

The solution for which is

$$|\rho(t)\rangle = \hat{\mathcal{T}} \exp\left(\int_0^t d\tau \mathcal{L}(\tau)\right) |\rho(0)\rangle , \tag{5.12}$$

where $\hat{\mathcal{T}}$ denotes Dyson's time ordering operator. This defines another representation for the map $M$, as the $d^2 \times d^2$ matrix $\mathcal{M}$ for which $|\rho(t)\rangle = \mathcal{M}|\rho(0)\rangle$. This in turns implies the operator form of Eq. (5.11)

$$\frac{\mathrm{d}}{\mathrm{d}t}\mathcal{M}(t) = \mathcal{L}(t)\mathcal{M}(t) , \qquad \mathcal{M}(0) = \mathbb{I} . \tag{5.13}$$

The solution for which is

$$\mathcal{M}(t) = \hat{\mathcal{T}} \exp\left(\int_0^t d\tau \mathcal{L}(\tau)\right) . \tag{5.14}$$

It is clear that in the time-independent $\mathcal{L}$ case that the solution simplifies to $\mathcal{M}(t) = \mathrm{e}^{\mathcal{L}t}$.

The time-dependent version of Eq. (5.11) can be used to investigate the control of open systems, by substituting $H$ with the controlled Hamiltonian of Eq. (1.22). Examples of this can be found in

Refs. [171, 172]. The control fields are typically assumed neither to be modulated by, nor directly affect, the dissipative system-environment interactions. This assumption has been criticised when the dynamics are non-Markovian [194].

Control optimisation with piecewise constant control functions was described in Section 1.6. For the GRAPE algorithm, where the Lindbladian is constant in the timeslots, the propagators in the timeslots are maps of the form $e^{\mathcal{L}t}$. The combined Lindblad superoperator is

$$\mathcal{L}(t) = \mathcal{L}_0 + \sum_{j=1}^{N_c} u_j(t)\mathcal{L}_j \,, \tag{5.15}$$

where the $\mathcal{L}_j$ are the control Lindlad superoperators, which when these are assumed to act only on the system, are of the form $\mathcal{L}_j = \mathrm{i}\left(H_j^{\mathsf{T}} \otimes \mathbb{I} - \mathbb{I} \otimes H_j\right)$, where the $H_j$ are control Hamiltonians, makes them superoperators of the Liouville-von Neumann equation Eq. (1.7). The dissipative effects are wholly contained in the drift Lindbladian $\mathcal{L}_0$. The evolution in the $k^{\text{th}}$ time slice is given by

$$\mathcal{M}_k = \mathrm{e}^{\mathcal{L}_k \Delta t}. \tag{5.16}$$

The full evolution is given by

$$\mathcal{M}(T) = \mathcal{M}_{N_{\mathrm{t}}}\mathcal{M}_{N_{\mathrm{t}}-1} \cdots \mathcal{M}_k \cdots \mathcal{M}_2\mathcal{M}_1. \tag{5.17}$$

The cost function for the optimisation is the ‘trace difference’ infidelity defined in Eq. (1.48). For map operator optimisation the scale factor is $\lambda = 1/(2d^2)$. For density operator optimisation, the scale factor is $\lambda = 1/2$ (as in Eq. (1.52)). Exact gradients can be calculated for the propagators, and hence the infidelity gradients, using the Fréchet derivative method, as the Lindblad superoperator is applied as general linear matrix.

Not all processes are Markovian. Therefore in some cases a different description of the dynamics is needed. One that claims to be *exact*, in so far as the Born-Markov approximation is not required, is the hierarchical equations of motion (HEOM), which is described in the next section.

## 5.3 The hierarchical equations of motion

The canonical separation of the total Hamiltonian of system and environment is

$$H_{\mathrm{T}} = H_{\mathrm{S}} + H_{\mathrm{I}} + H_{\mathrm{B}}\,, \tag{5.18}$$

where $H_{\mathrm{S}}$ is the free Hamiltonian of system that solely would govern its dynamics if closed from the environment. $H_{\mathrm{B}}$ is the Hamiltonian of the bath and, most importantly for an open systems description, $H_{\mathrm{I}}$ is the Hamiltonian that describes how the system and bath interact.

There are many derivations of the HEOM, as mentioned in the introduction to this chapter. As the primary interest in this study is for a method that can be used in controlled open system evolution, for which the system Hamiltonian is time-dependent, then the derivation in Ref. [179] is the primary source in this section, as it explicitly states that it is valid for systems driven by external fields. The full derivation is not repeated here, but notation from the paper is used so that the reader may easily refer back to the source.

In the model for the HEOM the bath is a collection harmonic oscillators, which need not be of effectively infinite number. The only assumption is the interactions with the system satisfy Gaussian statistics. The form of Eq. (5.18) for the HEOM derivation is

$$H_{\mathrm{T}} = H(t) - \sum_{a} Q_a \hat{F}_a + h_{\mathrm{B}}\,. \tag{5.19}$$

The system Hamiltonian is explicitly time-dependent. The exact form of the bath Hamiltonian $h_{\mathrm{B}}$ is not important, because its effects on the system are entirely described by the interaction Hamiltonian, in which the $Q_a$ are system operators and $\hat{F}_a$ the bath interaction operators.

The canonical bath ensemble average, defined for some bath operator $\hat{O}$, is

$$\langle \hat{O} \rangle_{\mathrm{B}} \equiv \mathrm{Tr}_{\mathrm{B}}(\hat{O}\rho_{\mathrm{B}}^{\mathrm{eq}}) = \mathrm{Tr}_{\mathrm{B}}(\hat{O}\mathrm{e}^{-\beta h_{\mathrm{B}}})/\mathrm{Tr}_{\mathrm{B}} e^{-\beta h_{\mathrm{B}}}\,, \tag{5.20}$$

where $\beta \equiv 1/T$ is the inverse temperature and $\rho_{\mathrm{B}}^{\mathrm{eq}}$ being the thermal equilibrium state of the bath. The $\hat{F}_a$ are Gaussian stochastic processes, meaning that $\langle \hat{F}_a \rangle_{\mathrm{B}} = 0$. Based on these assumptions the effects of the bath operators are encapsulated in the correlation functions

$$C_{ab}(t-\tau) = \langle \hat{F}_a(t)\hat{F}_b(\tau)\rangle_{\mathrm{B}}\,. \tag{5.21}$$

As before, the primary interest in is the time evolution of the system density operator

$$\rho(t) = \mathrm{Tr}_{\,\mathrm{B}}\rho_{\mathrm{T}}(t) = \mathcal{U}(t,t_0)\rho(t_0)\,, \tag{5.22}$$

where $\mathcal{U}$ acting as a map on $\rho$ is the propagating superoperator. Again this is based on separable states at time $t_0$, that is $\rho_{\mathrm{T}}(t_0) = \rho(t_0) \otimes \rho_{\mathrm{B}}^{\mathrm{eq}}$

The propagator $\mathcal{U}$ is found using the path integral formalism [195], using an approach that the authors of Ref. [179] call influence generating functionals calculus-on-path-integral (IGF-COPI) which they detail in a previous paper [196]. For this $\{|\alpha\rangle\}$ is a basis set in the system subspace, with $\boldsymbol{\alpha} = (\alpha, \alpha')$ being a shorthand for some path in the state space. In this representation the evolution of the system density operator from Eq. (5.22) is given by

$$\rho(\boldsymbol{\alpha},t) \equiv \rho(\alpha,\alpha',t) = \int d\boldsymbol{\alpha}_0 \mathcal{U}(\boldsymbol{\alpha},t;\boldsymbol{\alpha}_0,t_0)\rho(\boldsymbol{\alpha}_0,t_0)\,. \tag{5.23}$$

The propagator is calculated using the path integral

$$\mathcal{U}(\boldsymbol{\alpha},t;\boldsymbol{\alpha}_0,t_0) = \int_{\boldsymbol{\alpha}_0[t_0]}^{\boldsymbol{\alpha}[t]} \mathcal{D}\boldsymbol{\alpha}\, \mathrm{e}^{\mathrm{i}S[\alpha]}\mathcal{F}[\boldsymbol{\alpha}]\mathrm{e}^{-\mathrm{i}S[\alpha']}. \tag{5.24}$$

The free dynamics of the reduced system are encapsulated in the classical action functional $S[\alpha]$. This is evaluated along the path $\alpha(\tau)$, which is constrained by the two *fixed* ending points $\alpha(t_0) = \alpha_0$ and $\alpha(t) = \alpha$. The interaction with the bath is through the Feynman-Vernon influence functional $\mathcal{F}$. If there were no interactions with the bath, then $\mathcal{F} = 1$, and the dynamics would reduce to the Liouville-von Neumann equation for the superoperator

$$\frac{\mathrm{d}}{\mathrm{d}t}\mathcal{U} = -\mathrm{i}[H(t),\mathcal{U}]\,, \tag{5.25}$$

that is, fully coherent dynamics.

The system-bath interactions are all incorporated into $\mathcal{F}$ through the correlation functions $C_{ab}$ and system operators $Q_a$. It turns out, with steps in detail given in Ref. [179], that Eq. (5.24) becomes a product of integrals for which the time derivative depends only on the fixed end points. This then gives a differential equation of the form

$$\frac{\mathrm{d}}{\mathrm{d}t}\mathcal{U} = -\mathrm{i}[H(t),\mathcal{U}] - [\mathcal{R}(t),\mathcal{U}]\,, \tag{5.26}$$

or equivalently

$$\frac{\mathrm{d}\rho}{\mathrm{d}t} = -\mathrm{i}[H(t),\rho] - [\mathcal{R}(t),\rho]\,. \tag{5.27}$$

The first half of the RHS being the only place where the system Hamiltonian appears means that so long as it is assumed that the external fields only affect the system, and not the interaction Hamiltonian, then controlled evolution in the presence of noise can be calculated using this model. The bath correlations and system-bath couplings determine the *dissipation functional* $\mathcal{R}(t)$. The operator form of Eq. (5.26) means it can used to evolve maps that can be expressed as operators.

An exact operator-level expression for $\mathcal{R}$ is not generally available. However Eq. (5.26) can be solved using a hierarchy of coupled differential equations, the form which depends on the parametrisation scheme for the bath correlation functions. In the Drude-Debeye model this is

$$C_{\boldsymbol{a}}(t>0) \equiv C_{aa'}(t) = \eta_{\boldsymbol{a}}\mathrm{e}^{-\gamma_{\boldsymbol{a}}t}\,, \tag{5.28}$$

where the frequencies $\gamma_{\boldsymbol{a}}$ are real and coefficients $\eta_{\boldsymbol{a}}$ are complex. For this model, as derived in Ref. [179], the set of hierarchical coupled differential equations are

$$\dot{\rho}_{\mathsf{n}} = -\Big(-\mathrm{i}\mathcal{L} + \sum_{\boldsymbol{a}} n_{\boldsymbol{a}}\gamma_{\boldsymbol{a}}\Big)\rho_{\mathsf{n}} + \sum_{\boldsymbol{a}}\Big(n_{\boldsymbol{a}}\mathcal{C}_{\boldsymbol{a}}\rho_{\mathsf{n}_{\boldsymbol{a}}^-} - \mathrm{i}\mathcal{Q}_a\rho_{\mathsf{n}_{\boldsymbol{a}}^+}\Big)\,. \tag{5.29}$$

The set of non-negative integers $\mathsf{n} = (n_a, n_b, \dots)$ determines the interaction between *ancillary density operators*. The number of elements in $\mathsf{n}$ is equal to the number of different types of bath mode pair interactions, that is the number of different correlation functions Eq. (5.28), as determined by the frequencies and coefficients. The integers $n_a, n_b, \dots$ are the excitation level of the bath mode. Each $\rho_{\mathsf{n}}$, as identified by its unique index, is an ancillary density operator, except

$\rho_{(0,0,\ldots,0)}$, which is the system density operator. The index set $\mathsf{n}_{\boldsymbol{a}}^{\pm} = (n_a \pm 1, n_b, \ldots)$ differs only from $\mathsf{n}$ in the specific $n_a$, which is changed to $n_a \pm 1$.

The operators in Eq. (5.29) are defined as

$$\mathcal{L}\hat{O} = [H(t), \hat{O}] \tag{5.30a}$$

$$\mathcal{C}_{\boldsymbol{a}}\hat{O} = -\mathrm{i}(\eta_{\boldsymbol{a}} Q_a \hat{O} - \eta_{\boldsymbol{a}}^{*} \hat{O} Q_a) \tag{5.30b}$$

$$\mathcal{Q}_{\boldsymbol{a}}\hat{O} = Q_a \hat{O} - \hat{O} Q_a \tag{5.30c}$$

Through the interactions between the system and ancillary density operators, the action of $H(t)$ will drive the strength of the correlations between bath modes, and hence in return these correlations may drive the system. So the memory of the system-bath interactions are stored in the ancillary density operators, which is how the HEOM allow for non-Markovian dynamics.

The model is only exact when taken to an infinite level of hierarchy, as the bath is modelled as oscillators, which have an infinite number of potential states, as explained in Chapter 3. Even allowing for a moderate number of excitation levels results in a great many ancillary elements, which is expanded upon with an example bath correlation model in the next section. It can be seen from Eq. (5.29) that the effect of the bath correlations on the dynamics are included based on the frequencies $\gamma_{\boldsymbol{a}}$ and coefficients $\eta_{\boldsymbol{a}}$ only. This means that any set of correlation functions that can be represented as exponentials with real decay constants and complex coefficients can be used with the HEOM. One such, that has physical relevance, is the Drude-Lorentz model for the spectral density.

## 5.4 Spectral density

The bath interacts with the system through the coupling of oscillator modes with system energies. Not all mode frequencies are equally likely, consequently some way of modelling their relative probability is required – this is the spectral density. The magnitude of the interaction is also dependent on the temperature of the bath and the strength of the coupling with the bath. For the bath interaction models considered in this chapter the temperature dependence is accounted for

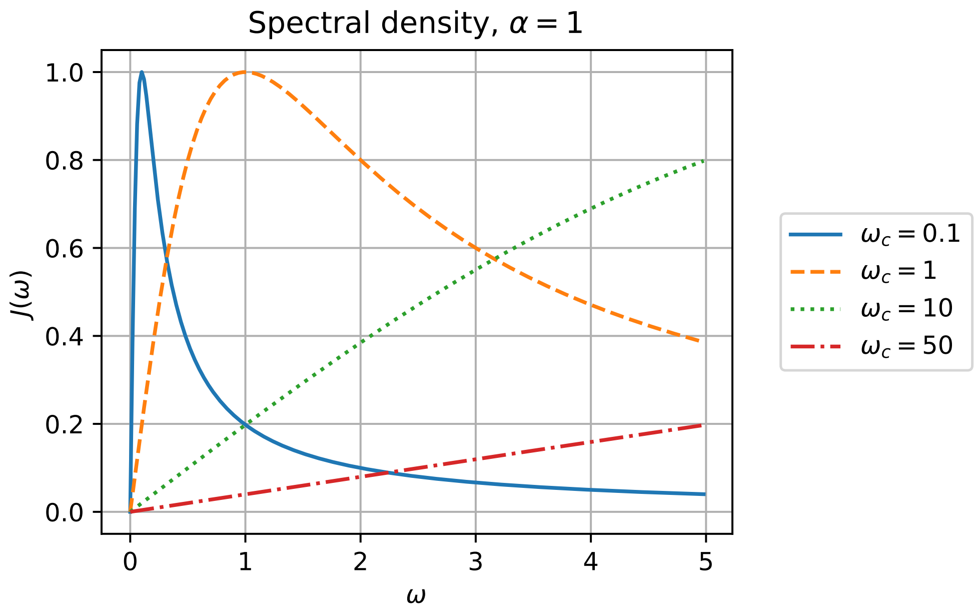


**Figure 5.1:** Plots of the Drude-Lorentz spectral density of Eq. (5.31), with $\alpha = 1, \omega_c = 0.1, 1, 10, 50$. Note that at $\omega = 1$: for $\omega_c = 0.1$ there is a steep gradient, $\omega_c = 1$ there is a resonance peak, whereas $J(\omega)$ is approximately linear for $\omega_c = 10$, and, relative to the others, $J(\omega)$ is approximately constant for $\omega_c = 50$.

separately. The coupling strength is considered independent of mode frequency and bath temperature, and so is a constant scaling factor for the interaction. One such model is the Drude-Lorentz model for the spectral density, which is often referred to as a model for an underdamped Brownian oscillator. The form used in this study is

$$J_{\mathrm{DL}}(\omega) = \omega \frac{2\omega_c \alpha}{\omega^2 + {\omega_c}^2} \,, \tag{5.31}$$

where $\omega_c$ is described as the cut-off frequency and sets the position of the maximum of the function, and hence is also referred to as the resonant frequency. The scaling factor coupling strength is $\alpha$. The Fourier transform of Eq. (5.31) can be decomposed as a sum of exponentials, as in Eq. (5.28). For these reasons it is commonly used in studies of non-Markovian dynamics, including one of the earliest HEOM papers [174], and more recent work [197, 181]. Plots of this function are shown in Fig. 5.1 for a range of $\omega$ that is relevant in the numerical investigation in the later sections.

The Fourier transform of the spectral density gives the sum of exponentials in the time domain

$$C(t) = \sum_{k=0}^{\infty} \eta_k \mathrm{e}^{-\gamma_k t} \,. \tag{5.32}$$

Each exponential term in the summation $C_k = \eta_k \mathrm{e}^{-\gamma_k t}$ can be taken as a two-time correlation function in Eq. (5.28). Note the spectral density transform into correlation functions is only exact for an infinite number of exponential terms, but assuming a truncation of the sum at $k = M$, it can be used in the HEOM model to solve the dynamics numerically. Each $C_k$ is referred to as a Matsubara term, and the corresponding Matsubara frequencies and prefactors are given by

$$\begin{cases} \gamma_k = 2\pi k/\beta & k \geq 1 \\ \gamma_0 = \omega_c & \\ \eta_k = \frac{4\alpha\omega_c\gamma_k}{\beta({\gamma_k}^2+\omega_c^2)} & k \geq 1 \\ \eta_0 = \alpha\omega_c[-\mathrm{i} + \cot(\beta\omega_c/2)] & \end{cases} \tag{5.33}$$

The influence of the temperature of the bath on the strength of interaction with the system enters here through the inverse temperature $\beta = 1/T$. It can be seen from the form of the formula for $\eta_k$ that $\eta_k \to 0$ as $k \to \infty$, and so some termination will provide an approximation. Also that $\eta_k \to 0$ as $\beta \to \infty$ ($k > 0$), whereas $\eta_0 \to \infty$ as $\beta \to 0$, meaning that the $k = 0$ term will dominate at higher temperatures. According to Ref. [174], for high temperature baths ($T \gtrsim 10$), the Matsubara terms for $k \geq 1$ will have negligible effect.

The QuTiP implementation of a solver using the HEOM coupled differential equations Eq. (5.29) is based on the Drude-Lorentz spectral density. The Matsubara parameters are calculated as per Eq. (5.33) based on the spectral density parameters $\alpha, \omega_c, T$, leading to the form of the HEOM implemented in QuTiP, which is described in Section 2.8.2.

The bath cutoff parameter $N_c$ and number of Matsubara terms $K = M + 1$ must be chosen. These must be selected carefully, as the number of ancillary elements in the hierarchy is $(N_c + K)!/(N_c!K!)$, so this quickly becomes infeasible when both $N_c \gg 1$ and $K \gg 1$. The following sections validate the HEOM implementation in QuTiP, and investigate the appropriate $N_c$ and $K$ parameter values for certain spectral density scenarios.

## 5.5 Comparison with pure-dephasing

One of the few exactly solvable models for open systems dynamics is the pure-dephasing of a qubit – a two-level system (TLS), such a spin-½ system in an external field or a two level atom – coupled to a cavity modes with an arbitrary spectral density. The Hamiltonian for the two level system is

$$H_s = \frac{\omega_q}{2}\sigma_z + \frac{\Delta}{2}\sigma_x \tag{5.34}$$

where $\omega_q$ is the energy of the TLS and $\Delta$ is the tunnelling term. The two energetic states of the system are denoted $|g\rangle$ and $|e\rangle$. The coupling to the bath is through $Q = \sigma_z$. In this investigation into the dephasing predicted by the HEOM model, to ensure only dephasing effects of the coupling to the bath, then $\omega_q = 1, \Delta = 0$ and the initial state is set at $\psi(0) = (|e\rangle + |g\rangle)/\sqrt(2)\,, \rho(0) = |\psi(0)\rangle\langle\psi(0)|$. The coherence (purity) of the state is given by the absolute value of the off diagonals of $\rho(t)$, that is $|\rho_{12}|$, with $\rho_{12} := \langle e|\rho|g\rangle$. For the initial pure state $\rho_{12}(0) = 0.5$, and in the maximally mixed state $\rho_{12} = 0$.

The exact model for this pure dephasing set up is

$$\rho_{12}(t) = \exp\left[-\int_0^\infty d\omega \left\{4J(\omega)\coth\frac{\beta\omega}{2}\left(\frac{1-\cos(\omega t)}{\omega^2}\right)\right\}\right]\rho_{12}(0)\,. \tag{5.35}$$

A derivation of this can be found in Ref. [22]. A derivation based on the model from Ref. [176] is also included in Appendix E. Eq. (5.35), with $J(\omega) = J_{\mathrm{DL}}(\omega)$ from Eq. (5.31), is solved here numerically to compare with the evolution computed using the HEOM solver in QuTiP.

As discussed in the previous sections, the HEOM model is only exact when taken to an infinite level of hierarchy, and the spectral density transform into correlation functions is only exact for an infinite number of Matsubara terms. Clearly this is impossible in numerical simulation, and so some truncation is necessary to use the HEOM in control optimisation. Numerical simulation is used to determine the Matsubara parameters for satisfactory convergence of HEOM with the exact model. It is expected that only one Matsubara term is required at high temperatures, which is $T \gtrsim 10$. When the interaction is stronger at $\omega_c \approx \omega_q$, and as the bath interactions become less Markovian, that is $\omega_c \ll \omega_q$, it is expected that a deeper hierarchy will be required.

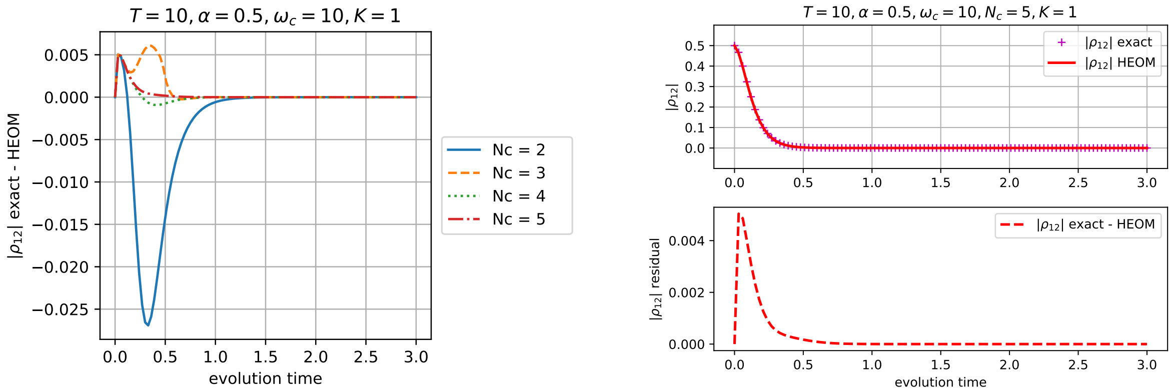


**Figure 5.2:** Convergence of the HEOM numerical simulation for temperature $T = 10$, coupling strength $\alpha = 0.5$, spectral density cut-off frequency $\omega_c = 10$. The coherence (purity) of the state is given by the absolute value of the off diagonals of $\rho(t)$, denoted $|\rho_{12}|$. The plot on the left shows the differences between the HEOM evolution and the exact model for increasing level bath cut-off $N_c$. The plots on the right show (upper) the evolution of the coherence, and (lower) the difference from the exact model, for the best attempted HEOM parameters $N_c = 5, K = 1$.

The investigation starts with the cut-off frequency that puts the resonance outside of the energetic range of the system. The plots in Fig. 5.2 indicate that for $\omega_c = 10, T = 10$ a bath cut-off of five and one Matsubara term are required for good convergence with the exact model. At the lower temperature of $T = 1$ used to produced the plots in Fig. 5.3 convergence is only around 90% for $K = 1$, but continues to improve with increasing numbers of Matsubara terms, with maximum error of around 2% for $K = 6$.

The results for cutoff frequency on resonance with the system energy, that is $\omega_c = 1$, for the bath temperature $T = 10$ are shown in Fig. 5.4. Due to the increased interaction with the bath (compared to the the $\omega_c = 10$ scenario) as $J(\omega)$ is much greater at the system energy, then a much greater bath cut-off is required, resulting in a deeper hierarchy. Here $N_c = 30, K = 1$ gives a maximum error of around 0.1%. With the same cutoff frequency, but lower temperature, (shown in Fig. 5.5) fewer hierarchy levels are required. At the lower temperature the average excitation count is lower, and hence the interaction is weaker. However, again more Matsubara terms are required at the lower temperature. The best HEOM parameters tested here $N_c = 10, K = 5$ give an error of around 0.2%, whereas the worst performing parameters attempted $N_c = 4, K = 1$ give a maximum error of approx 6%.

Finally, with the spectral density cut-off $\omega_c = 0.1$ and $T = 10$, at the very strong coupling of

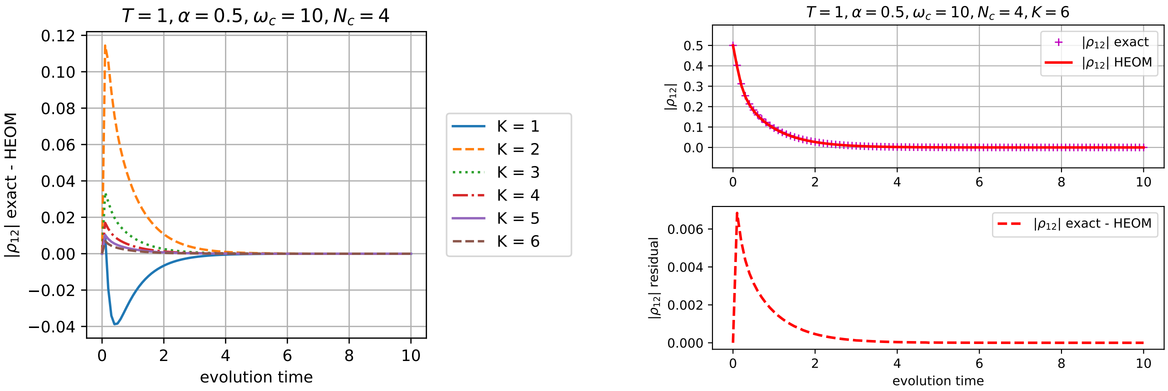


**Figure 5.3:** Convergence of the HEOM numerical simulation for $T = 1, \alpha = 0.5, \omega_c = 10$ (see Fig. 5.2 for description of parameters). The bath cut-off of $N_c = 4$ found for sufficient for $T = 10$ is used again here. The plot on the left shows the differences between the HEOM evolution and the exact model for increasing numbers of Matsubara terms $K$. The plots on the right show (upper) the evolution of the coherence, and (lower) the difference from the exact model, for the best attempted HEOM parameters $N_c = 4, K = 6$.

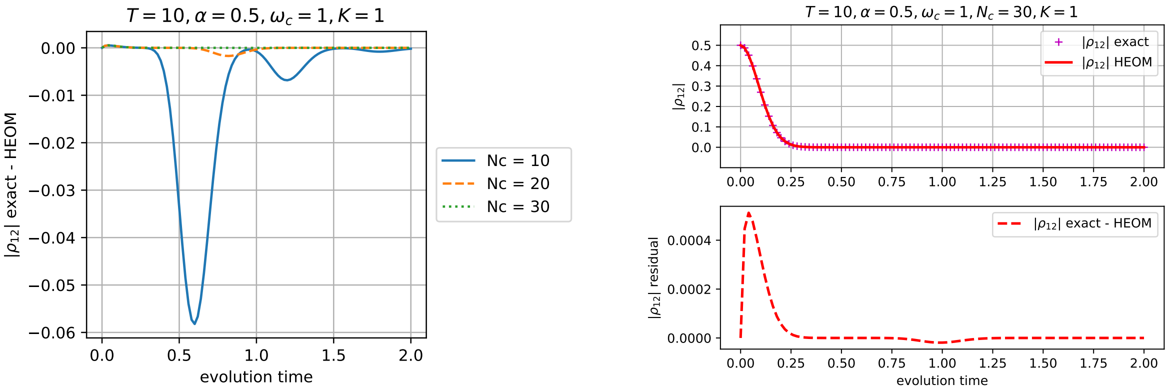


**Figure 5.4:** Convergence of the HEOM numerical simulation for $T = 10, \alpha = 0.5, \omega_c = 1$ (see Fig. 5.2 for description of parameters). The plot on the left shows the differences between the HEOM evolution and the exact model for increasing levels of bath cut-off $N_c$. The plots on the right show (upper) the evolution of the coherence, and (lower) the difference from the exact model, for the best attempted HEOM parameters $N_c = 30, K = 1$.

$\alpha = 0.5$, a very high bath cut-off of $N_c = 100$ is required to bring the maximum error to within 0.2%, as shown in Fig. 5.6. Very similar features are seen with the smaller coupling of $\alpha = 0.05$, however only $N_c = 40$ is required to achieve similar convergence. With these spectral density cut-off and coupling parameters a single Matsubara term was found to be sufficient at $T = 1$ also. The difference between the $T = 1$ snd $T = 10$ scenarios (Fig. 5.7) is that there is some divergence early in the HEOM evolution, similar to that seen in the other $\omega_c$ scenarios for $T = 1$.

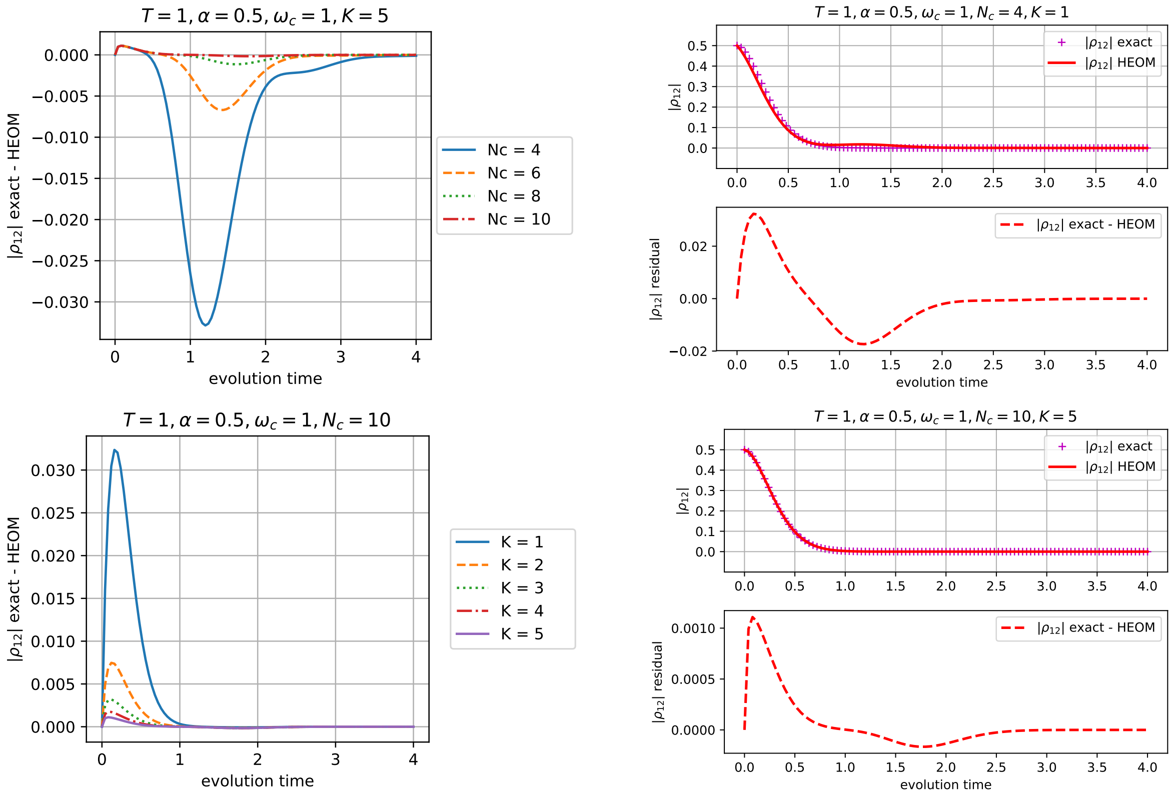


**Figure 5.5:** Convergence of the HEOM numerical simulation for $T = 1, \alpha = 0.5, \omega_c = 1$ (see Fig. 5.2 for description of parameters). The plots on the left show the differences between the HEOM evolution and the exact model; left (upper) for increasing levels of bath cutoff $N_c$, with $K = 5$, and (lower) for increasing number of Matsubara terms $K$, with $N_c = 10$. The plots on the right show the evolution of the coherence and its difference from the exact model. The upper two show the worst case tested $N_c = 4$, $K = 1$, and the lower two for the best HEOM parameters $N_c = 10$, $K = 5$.

## 5.6 Comparison with Lindblad master equation

For the combined dissipation and dephasing there is no exact model for validating the HEOM solver. However, it is possible to compare the evolution computed with the HEOM model and numerical solutions to the Lindblad master equation Eq. (5.7). The solutions should converge where the Born-Markov approximation can be made. Standard qubit-bath master equations assume that coupling to the bath is weak. The HEOM does not make this assumption, and so it would be better to have a Lindblad master equation that also holds in the ultrastrong coupling regime for comparison.

Using a model referred to as the *dressed picture* in Ref. [198], the authors derive a Lindbladian

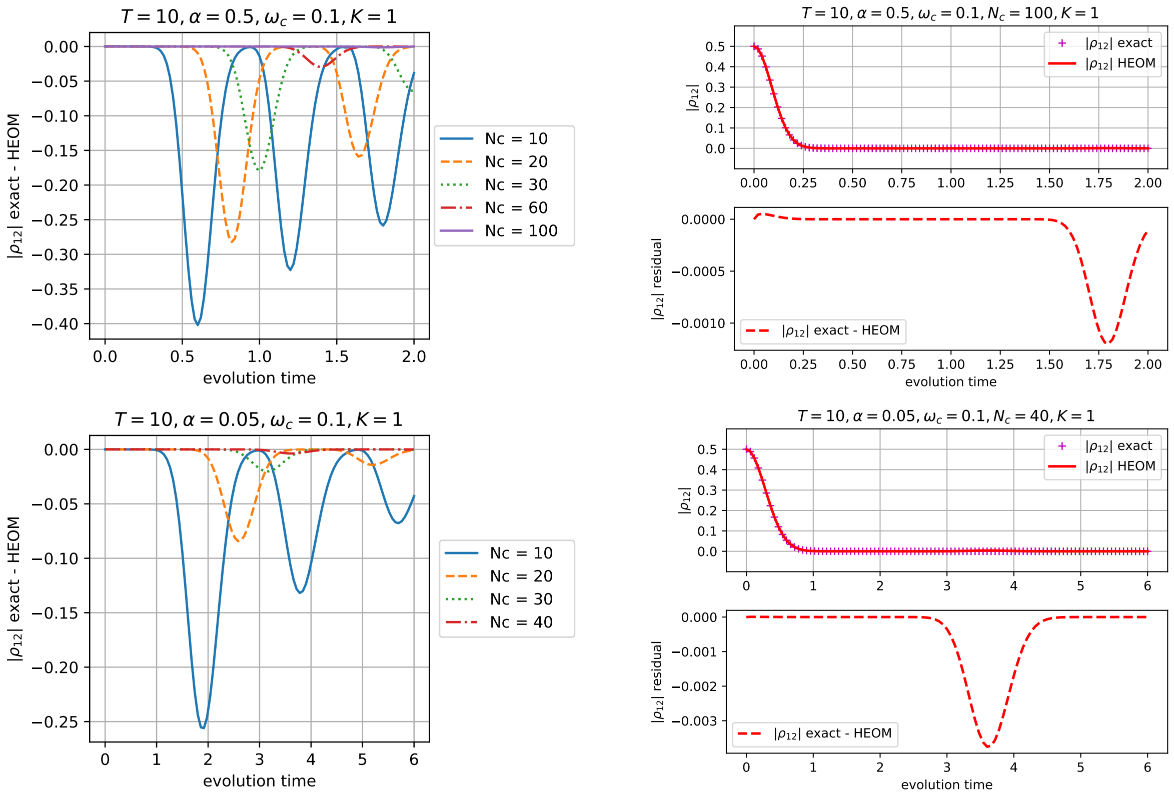


**Figure 5.6:** Convergence of the HEOM numerical simulation for $T = 10, \omega_c = 0.1$ (see Fig. 5.2 for description of parameters). The upper set of plots are for $\alpha = 0.5$ and the lower for $\alpha = 0.05$. The plots on the left show the differences between the HEOM evolution and the exact model for increasing levels of bath cut-off $N_c$, with $K = 1$. The plots on the right show the evolution of the coherence and its difference from the exact model with the best HEOM parameters tested for the corresponding $\alpha$.

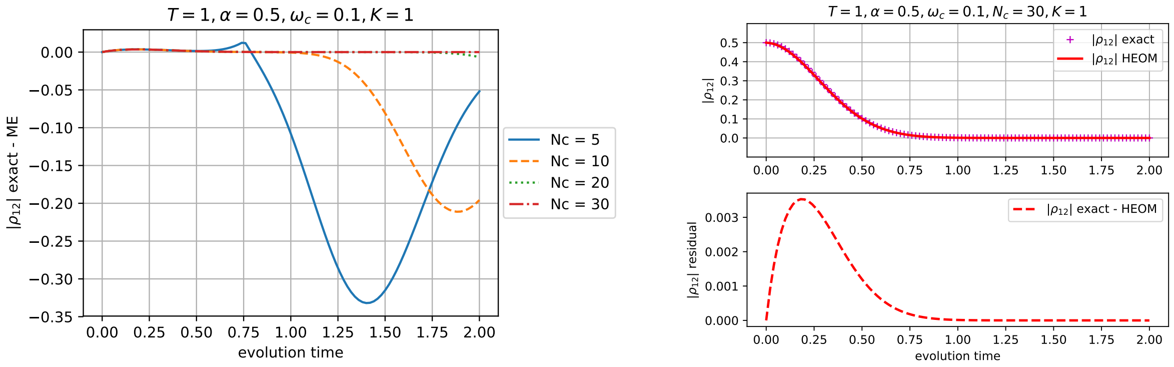


**Figure 5.7:** Convergence of the HEOM numerical simulation for $T = 1, \alpha = 0.5, \omega_c = 0.1$ (see Fig. 5.2 for description of parameters). The plots on the left show the differences between the HEOM evolution and the exact model for increasing levels of bath cut-off $N_c$, with $K = 1$. The plots on the right show the evolution of the coherence and its difference from the exact model with the best attempted HEOM parameters.

[Eq.(16) of 198], based on the transition probabilities of the system states, that takes account of qubit-resonator coupling to the bath in the ultrastrong regime. Based on the same method, a Lindbladian for a system with Hamiltonian $H_S$ coupled to a bath of harmonic oscillators that allows for strong coupling is described below. Firstly $H_S$ is taken in a diagonal basis to give the eigendecomposition $\{E_j, |j\rangle\}$, ordered by increasing energy, with the difference denoted $\Delta_{kj} = E_k - E_j$. The system is coupled to the bath by operator $Q$ with coupling strength $\alpha$. The bath is characterised by a spectral density $J(\omega)$ for oscillator mode frequencies $\omega$, in which $\alpha$ is a scaling factor. The bath temperature $T$, gives an average phononic boson count

$$N_{\mathsf{bs}}(\nu, T) = \frac{1}{e^{\nu\beta} - 1}\,, \tag{5.36}$$

where $\beta = 1/T$. Then, defining the dissipater operator as

$$D[V](\rho) = V\rho V^\dagger - \tfrac{1}{2}(V^\dagger V\rho + \rho V^\dagger V)\,, \tag{5.37}$$

the dissipative part of the Lindbladian is given by

$$L_{\mathrm{dr}}(\rho) = \sum_{j,k>j} \{\Gamma_\kappa^{jk} D[|j\rangle\langle k|](\rho) + \Gamma_\gamma^{kj} D[|k\rangle\langle j|](\rho)\} + \sum_j \Gamma_\phi^j D[|j\rangle\langle j|](\rho)\,, \tag{5.38}$$

where

$$\Gamma_\kappa^{jk} = 2|\langle j|Q|k\rangle|^2 \alpha J(\Delta_{k,j})(1 + N_{\mathsf{bs}}(\Delta_{k,j}, T)\,, \tag{5.39}$$

$$\Gamma_\gamma^{kj} = 2|\langle k|Q|j\rangle|^2 J(\Delta_{k,j}) N_{\mathsf{bs}}(\Delta_{k,j}, T)\,, \tag{5.40}$$

$$\Gamma_\phi^j = 4|\langle j|Q|j\rangle|^2 J_0 T\,. \tag{5.41}$$

The rate of transition to higher energy states is given by $\Gamma_\kappa^{jk}$, where $|\langle j|Q|k\rangle|^2$ gives the transition probability, which is then factored by the spectral density corresponding to that energy and the phonon count. Similarly, the relaxation rate $\Gamma_\gamma^{kj}$ to lower energies is proportional to that transition probability. The dephasing rate in the eigenstates is given by $\Gamma_\phi^j$, where $J_0 := \lim_{\omega\to 0} J(\omega)/\omega$.

The plots in Fig. 5.8 show comparisons between the dressed picture master equation (DME)

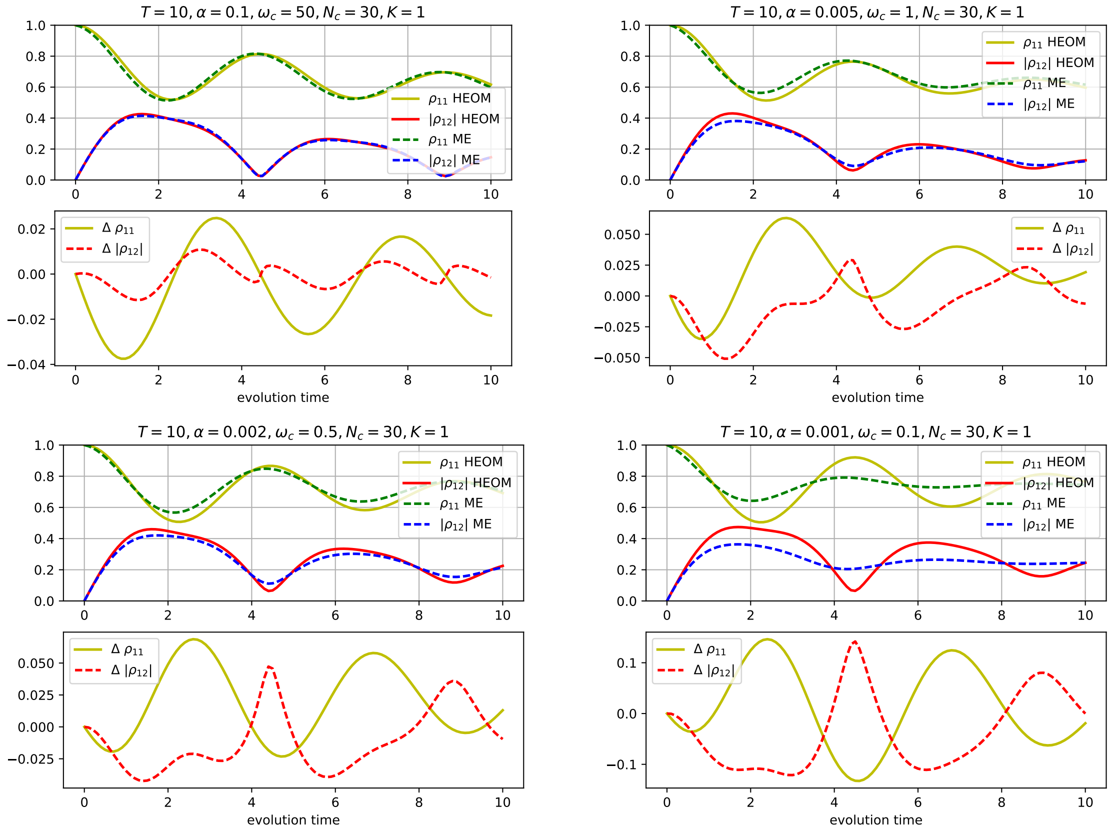


**Figure 5.8:** Comparison of the HEOM and dressed picture ME evolution of the two-level system with Hamiltonian from Eq. (5.34) with $\omega_q = 1, \Delta = 1$. The spectral density cut-off for the four scenarios decreases left to right and top to bottom, with values $\omega_c = 50, 1, 0.5, 0.1$. Coupling constant values are chosen as $\alpha = 0.1, 0.005, 0.002, 0.001$ respectively, such that the dissipation is similar. All scenarios have temperature $T = 10$. Two plots are shown for each scenario. The upper axes show plots for both models of the first element of the density matrix $\rho_{11}$ and the absolute value of the off diagonals $|\rho_{12}|$ during the evolution. The lower axes show plots of the difference between the models for the two measures.

and HEOM decoherence models for different spectral densities. The coupling constant $\alpha$ values are chosen such that the dissipation from $t = 0$ to $t = 10$ is similar in each spectral density cut-off $w_c$ scenario. The system evolves under its Hamiltonian Eq. (5.34) with $\omega_q = 1, \Delta = 1$ from its initial excited state $|e\rangle$. The probability of measuring the system in the state $|e\rangle$ is given by the first element of the density matrix and is hence denoted $\rho_{11}$. The absolute value of the off diagonals $|\rho_{12}|$ gives the maximum probability of measuring the system in an equal superposition state. In the scenario where $\omega_c \gg E$ ($E$ being the system energy levels) the Markovian approximation should hold, and the two noise models do appear to predict very similar evolution. When $w_c \ll E$

the bath memory time, via the correlation functions, is long compared with the system dynamics timescales, and hence the Markovian approximation is invalid. This is supported by the divergence between the predictions of the models for $w_c = 0.1$. The HEOM parameters used here are those that were found to converge well with the exact model in the previous section. Although the evidence in Fig. 5.8 is far from conclusive, as far as demonstrating the HEOM convergence in this more general setting, the two models do diverge where expected, potentially indicating that the Markovian approximation in the DME is not valid for $\omega_c \ll E$.

So, how reliable are noise models that make the Markovian approximation in predicting the effects of noise in some control experiment? The next section investigates a specific control objective to validate an optimisation performed using the Lindblad ME solver, in the Markovian and non-Markovian scenarios identified here.

## 5.7 Controlled evolution

Assuming a control optimisation searching for some control parameters that result in a desired evolution of the system, to achieve some target state or operation, where these are to be used in some physical experiment, if the model for the dissipation is inaccurate, then it would be doubtful that the control parameters, when applied to the actual system, would achieve the predicted result. This is investigated here by considering the HEOM as exact, and using it for a comparison with a driven system evolution where the decoherence is determined by the dressed master equation (DME) Lindbladian. That is, the controls are optimised in a simulation where the noise model is DME.

In the previous section the evolution of a two-level system (TLS) with the HEOM and DME were seen to coincide for a spectral density in the Markovian regime ($T = 10, \omega_c = 50, \alpha = 0.1$), but diverge for the non-Markovian ($T = 10, \omega_c = 0.1, \alpha = 0.001$). So the control parameters would be expected to achieve the same target in both models in the Markovian regime, but what will happen in the non-Markovian regime is difficult to predict. This is investigated using a TLS

Hamiltonian with controls

$$H_s(t) = \left(\frac{\omega_q}{2} + u_z(t)\right)\sigma_z + \left(\frac{\Delta}{2} + u_x(t)\right)\sigma_x\,, \tag{5.42}$$

where $u_z(t)$ and $u_x(t)$ are control functions. In this case they are piecewise constant in five equal durations, making ten control parameters in all. The control fields are assumed not to influence the bath. The collapse operators for the decoherence model are based on the energy splitting of $H_s$, as per Eq. (5.6), and so these are calculated for each timeslot. The system drift in this case is set by $\omega_q = 1, \Delta = 1$.

The control objective is to transfer from the initial pure state $|e\rangle$ to the pure superposition state $(|e\rangle + |g\rangle)/\sqrt{2}$. The cost function to optimise is the squared trace difference described in Section 1.6.1, Eq. (1.52), defined here again as

$$\mathcal{E}_{\mathrm{TD}^2}\left(\rho(t_f), \rho_{\mathrm{target}}\right) := \tfrac{1}{2}\,\mathrm{Tr}[(\rho(t_f) - \rho_{\mathrm{target}})^\dagger(\rho(t_f) - \rho_{\mathrm{target}})]\,. \tag{5.43}$$

where $\rho(t_f)$ is the density matrix at the end of the evolution, and $\rho_{\mathrm{target}}$ is the density matrix for the target. The control modules in QuTiP are used optimise the parameters of the control pulses to minimise the cost function. The method for computing the evolution of the open system dynamics is described in Section 2.8. The control amplitudes are bounded at $[-0.5, 0.5]$.

The dissipation scenarios are taken from the previous section, using the two extremes. The bath parameters, temperature $T = 10$, spectral density cutoff frequency $\omega_c = 50$ and coupling strength $\alpha = 0.1$, are expected to allow the Markovian assumption, whilst a bath characterised by $T = 10, \omega_c = 0.1, \alpha = 0.001$ potentially induces non-Markovian evolution. The plots in Fig. 5.9 show the optimised control pulses and resulting system evolution. The exact pure dephasing for both these scenarios predicts that, if in superposition state $(|e\rangle + |g\rangle)/\sqrt{2}$ for the whole evolution, the system would be almost completely dephased, that is in the maximally mixed state, with $\mathcal{E}_{\mathrm{TD}^2} \approx 0.25$. So the optimisation consequently finds parameters that favour keeping the system state close to its initial state (which is not subject to dephasing) until near the end of the evolution, and then allowing it to move rapidly to the target state near the end of the allowed time. This is particularly noticeable in the Markovian case.

The optimised control parameters found under the DME noise model are used to drive the system under the HEOM noise model. The system evolution using both models are plotted together in Fig. 5.9. As expected, in the Markovian scenario, the evolution matches very closely, and the final cost is almost identical, $\mathcal{E}_{\mathrm{TD}^2} = 0.011$ (DME) and $\mathcal{E}_{\mathrm{TD}^2} = 0.012$ (HEOM). More surprisingly, despite some deviation of state during the evolution, in the non-Markovian scenario, the final state is very similar for both noise models, with HEOM achieving a lower cost $\mathcal{E}_{\mathrm{TD}^2} = 7.4 \times 10^{-4}$ and DME $\mathcal{E}_{\mathrm{TD}^2} = 1.6 \times 10^{-3}$.

## 5.8 Dynamical decoupling optimisation

Dynamical decoupling (DD) is known to be a method for preserving the purity of quantum states [199]. It has been shown that the method only works for systems that are actually coupled to environments [200], and therefore cannot be simulated based on dynamics modelled using the Born-Markov approximation [189]. The HEOM are based on *actual* couplings to environmental oscillators, and therefore can be used to model DD. Consequently optimisation of theoretical DD schemes can be performed using the HEOM to compute the dynamics. An example of this is given here, based on the DD scheme presented in [201].

For simplicity, the rotating frame of the system is considered by setting $\omega_q = 0$ in Eq. (5.34), making the system Hamiltonian in this controlled system

$$H_s(t) = u_{\mathsf{DD}}(t)\sigma_x \,, \tag{5.44}$$

where $u_{\mathsf{DD}}(t)$ is the control function. In the DD case the control is restricted to a number of short ‘bursts’, known in the NMR community as ‘bang-bang’ pulses. Specifically, in this case, these are the so-called $\pi$ pulses. Working in the frame rotating in $z$ at $\omega_q/2$, these are defined by $2V\tau_{\mathsf{p}} = \pm\pi$, where $V$ is the amplitude and $\tau_{\mathsf{p}}$ is the duration of the individual pulses. This pulse induces a $\pi/2$ rotation about the Bloch sphere $x$-axis in the rotating frame. They are called $\pi$ pulses because they equate to $\theta = \pi$ in Eq. (1.11). Typically, in most theoretical work, it is assumed that this can be applied with a large enough amplitude $V$ such that $\tau_{\mathsf{p}} \to 0$, making the operation effectively instantaneous. In numerical simulation $\tau_{\mathsf{p}}$ must be chosen to be sufficient such that the pulse is

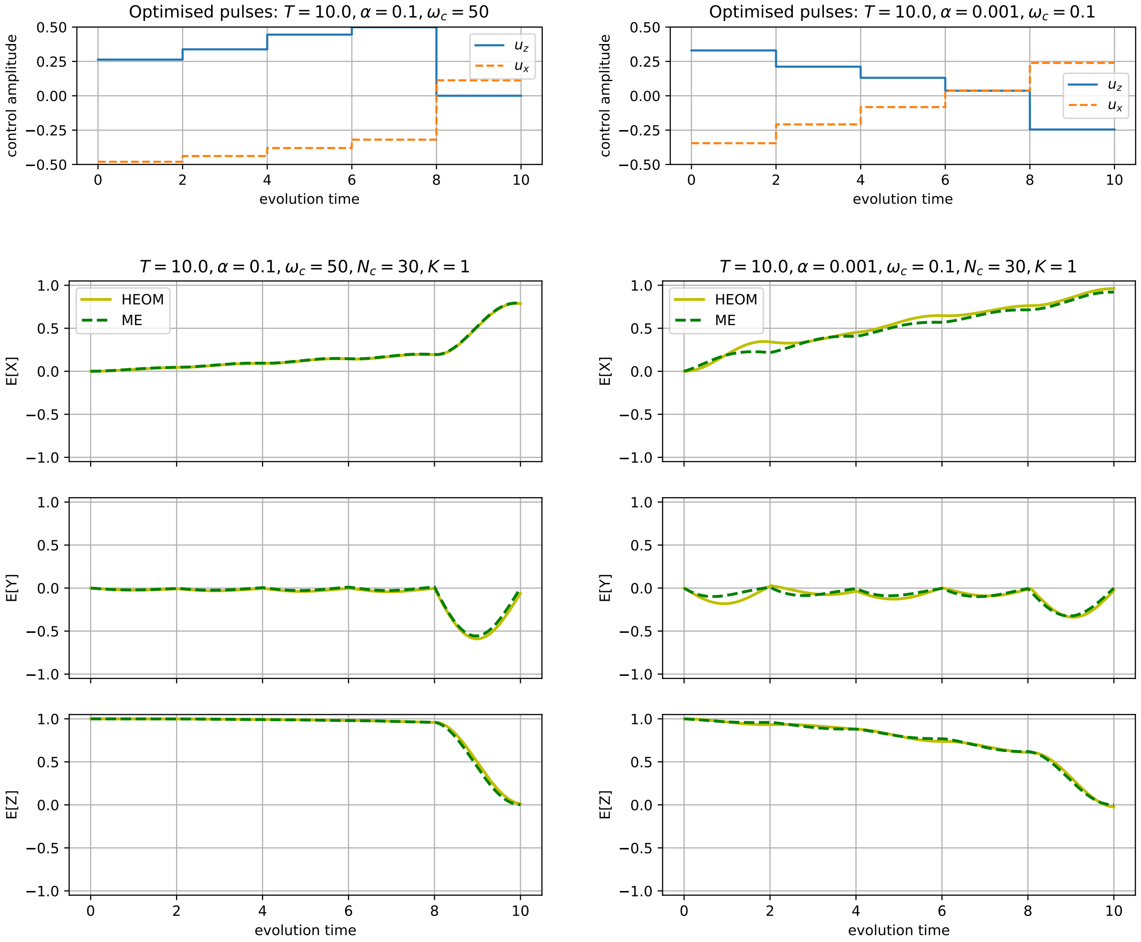


**Figure 5.9:** Controlled evolution of the two level system with Hamiltonian Eq. (5.42), comparing HEOM and dressed picture ME dissipation models in the Markovian and non-Markovian scenario. The system is initialised in the excited state $|e\rangle$. The target for the optimisation is $(|e\rangle + |g\rangle)/\sqrt{2}$. The plots on the left are resulting from the spectral density parameters, $T = 10, \omega_c = 50, \alpha = 0.1$, which is referred to as the Markovian scenario. The plots on the right are from simulations with $T = 10, \omega_c = 0.1, \alpha = 0.001$, which is the non-Markovian scenario. The upper plots show the piecewise constant control amplitudes found by numerical optimisation, which were bounded $[-0.5, 0.5]$. The lower plots show the expectation values throughout the evolution in the canonical basis.

accurately included in the numerical integration.

The objective of the control is to preserve the purity of the state, so the infidelity measure is based on purity. Therefore the cost for the optimisation is based on

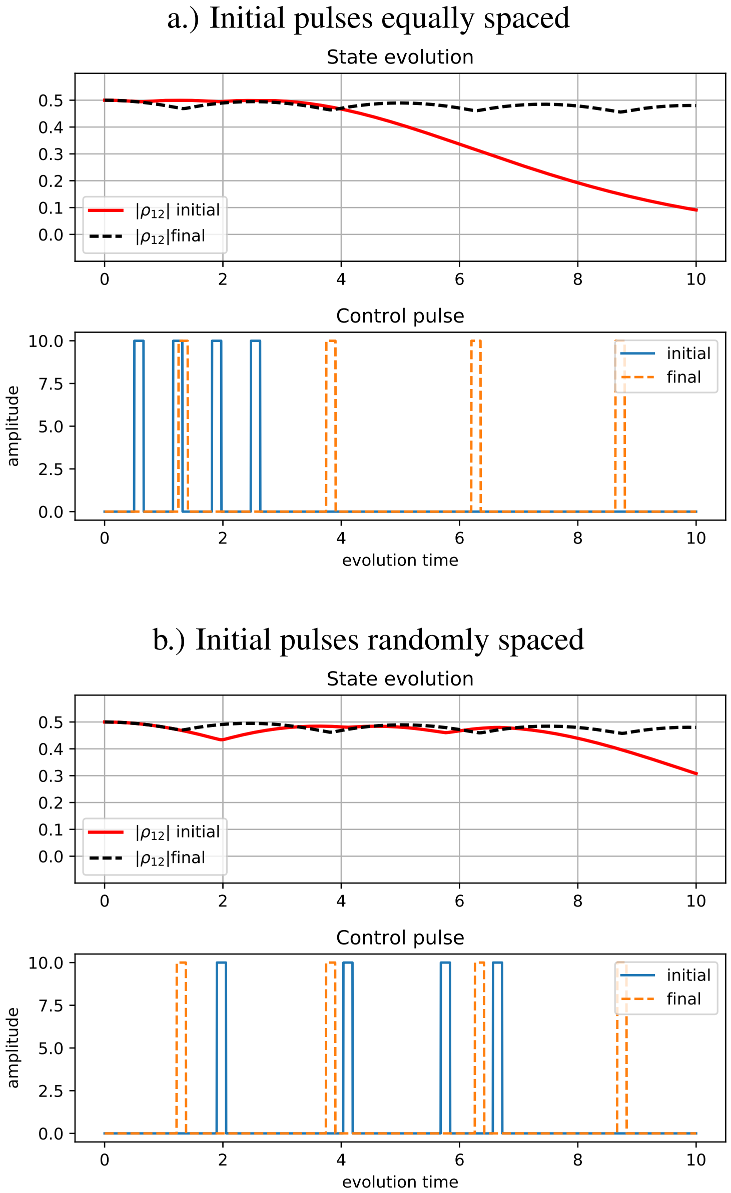
$$\mathcal{E}_{\text{pure}}\left(\rho(t_f)\right) = \mathcal{E}_{\text{TD}^2}\left(\rho(t_f), \rho(t_f)^2\right) , \tag{5.45}$$

with the trace difference infidelity for density matrices defined in Eq. (5.43). For a pure state $\rho = \rho^2$, and hence $\mathcal{E}_{\text{pure}} = 0$. As was seen in previous sections, with the system-bath interaction operator $Q = \sigma_z$, dephasing is observed when the system is in a superposition state in the canonical basis. Therefore, the system is initialised in the state $\psi(0) = (|e\rangle + |g\rangle)/\sqrt(2)$, with density matrix $\rho(0) = |\psi(0)\rangle\langle\psi(0)|$. Hence, as the Hamiltonian Eq. (5.44) has only $\sigma_x$ rotations, then the only dynamics of the state, other than global phase, can be dephasing.

The system is initially evolved with a 'guess' DD sequence, using the HEOM solver in QuTiP with time-dependent system Hamiltonian, as described in Subsection 2.8.2. The sequence is then optimised using the L-BFGS-B algorithm. The variables for the optimisation are the intervals between pulses. The results for optimisations using four DD pulses are shown Fig. 5.10. It can be seen from the plots of the evolution of $|\rho_{12}(t)|$ show how, after the application of the DD pulse, the coherence is returned to the system from the bath for a time period. The system then dephases again until the next DD pulse flips the system into the state where coherence starts to be returned. The optimal sequence is found to be that with equal spacing between the DD pulses, which would seem to arise from the symmetry of the dephasing / re-phasing periods of the evolution. As the evolution needs to symmetrical about the middle of the evolution time, to ensure the maximum coherence coincidences with the end of the evolution time, so then the DD sequence is symmetrical about the same. The same optimal sequence was found whichever initial sequence was chosen. Fig. 5.10 shows the results for two optimisations – one starting from equally separated, and the other from randomly separated, initial sequences. Further detail on how the optimisations were performed are given in Section 2.8.2.

## 5.9 Conclusion

Some justification has been given for the validity of the HEOM in driven system dynamics. The dissipation functional was shown to be independent of the system Hamiltonian [179], which would seem to imply that the bath is influenced by the state of system only, which is evolving in time whether the system is driven or otherwise. Hence the HEOM can be used to calculate the system propagation in control simulations, so long as the controls are assumed to only effect the system


a.) Initial pulses equally spaced

b.) Initial pulses randomly spaced

**Figure 5.10:** Results of optimisation of dynamical decoupling (DD) sequences, with controlled system Hamiltonian Eq. (5.44). The system is initialised in the superposition state $(|e\rangle + |g\rangle)/\sqrt{2}$ and is evolved using the HEOM. The coupling to the bath is through $Q = \sigma_z$. Two initial pulse scheme scenarios are shown: (a) 4 equally spaced $\pi$ pulses; (b) 4 randomly spaced $\pi$ pulses. The DD sequences are optimised to minimise the infidelity $\mathcal{E}_{\text{pure}}$. There are two sets of axes for each scenario, the upper showing the time-evolution of the system, and the lower the initial and optimised sequences. The red solid line shows the absolute value of off-diagonals of $\rho(t)$ over the evolution $|\rho_{12}(t)|$, based on the initial DD sequence (solid blue line). The black dashed line shows $|\rho_{12}(t)|$ for the evolution with the optimised DD sequence (dashed orange line). The decoherence arises through the HEOM with spectral density parameters $T = 10, \omega_c = 0.1, \alpha = 0.001$, and bath correlation parameters $N_c = 30, K = 1$.

directly. The derivation assumes that the system bath interaction operator is not time-dependent, so any time-dependence in the bath resulting from the driving fields could only be included in the parameters of the spectral density, such as the coupling strength or cut-off frequency, which may be worthy of investigation.

Open quantum system dynamics can be computed far more efficiently by solving the Lindblad master equation than by solving the hierarchical equations of motion. However, there are physical systems where the bath correlation times are known to exceed the limit that allows the Born-Markov approximation, and there are many others where it is suspected that this may also be so. The HEOM solver in QuTiP is sufficiently efficient to allow for computation of non-Markovian dynamics on a standard desktop computer in a few seconds for small quantum systems. This means that it is also suitable for control optimisation simulations for small systems.

The validation of the HEOM solver against the pure dephasing model demonstrates that truncations of the hierarchy can be made that provide sufficient precision, and which allow for efficient computation. This was shown for a range of spectral density parameters and temperatures. In particular, for high temperature scenarios, a single Matsubara term was found to be sufficient, which was predicted in Ref. [174]. There are known issues at low temperatures for this Drude-Lorentz spectral density parametrisation used in the HEOM. These are addressed with other parametrisations [179], which could be validated through the same method. It should be noted that this validation was only against one specific model. There is scope for further validation against other models, exact and approximate. This validation against the pure dephasing model was just to find HEOM parameters appropriate for the spectral density used in the other tests, rather than any validation of the exactitude of the HEOM.

The comparison with the Lindblad master equation dynamics showed convergence of the two solutions in the Markovian limit, but divergence when the bath frequency distribution potentially leads to correlation times that exceed the system dynamics timescales, which was as expected. The master equation used relies on the secular assumption, that is, high frequencies and degenerate energy levels a not considered. This may mean that it is inaccurate for pure dephasing and high frequency bath modes, which may account for some of the divergence encountered. There is potential to develop a non-secular ME solver, which may show closer convergence, and could be validated against the pure-dephasing model.

The similarity of outcome for pure state to pure state optimised controls is encouraging for the large body of work that assumes the Markovian model. In the driven example, the system evolution by solved master equation and HEOM are much closer than in the undriven case. This is likely due to the controls maintaining the system state where it is less subject to dephasing, and hence the difference between the models is not so evident. The dynamical decoupling optimisation demonstrated that it is possible to use the HEOM for investigating control objectives where modelling the actual couplings to environmental systems is essential. This is the first numerical optimisation of a dynamical decoupling scheme using a realistic bath model, as validated in this study and others. A genetic algorithm has been used for optimising dynamical decoupling [202]. However, in this work they use coupling to four spins as the model for the bath. The success of this initial test suggests potential for extension of the many results in the theory of dynamical decoupling, such as the randomised scheme in Ref. [203], or in Ref. [204], where they discuss the potential for DD while maintaining the positive effects of some other driving field.

Small systems only have been investigated here due to the extra expense of computing the dynamics with the HEOM. To extend this study into many-body systems it would be most beneficial to implement some method of parallelising the computation. The equations have been described as highly suited to parallel process solving [205], and these and other have reported performance advantages of using graphical processing units (GPUs) to solve the hierarchy [183, 206]. Therefore it should be fairly straightforward to implement in QuTiP, which would allow for investigation into scaling relations in control of many-body systems with open system evolution computed through the HEOM, thus allowing for non-Markovianity.

# Chapter 6

# Conclusion

The reachable set of symplectic operations on Gaussian systems with unstable quadratic Hamiltonians was proven not to contain any passive operations (except the identity) for single-mode systems. The numerics indicated a distinct reachable region. The shape of this region, at least for the most unstable system (defined by c=0 in Eq. (3.39)), is much simpler than that of the proven unreachable region. This hints that it should be possible to derive an analytical description of it. Attempts to find a similar reachable region with the two-mode unstable using numerics were less successful. However, The assumption about the inevitability of squeezing was challenged by some of the two-mode numerical results, and some analytical explanation for this may be a promising direction for further study, as reputedly the question of whether passives are inaccessible without neutral generators has remained open for some time for $n$-mode systems.

The numerics consistently found that systems with neutral generators were controllable; in the single-mode, two-mode and coupled chain simulations. However, the difficulty of reaching target symplectic operations, in all the systems studied, suggests that the control parameter space contains traps that do not exist for composite qubit system optimisations. Referring back to the discussion in Section 1.6.7, this raises the question of whether the existence of non-recurrent generators constitutes a constraint on the optimisation. The control parameter space was proven to be smooth and trap-free [14]. The experiences of this study, and anecdotal evidence from other researchers, runs contrary to this proof. Comparative analysis of these landscapes is also an interesting potential direction for further study.

The GRAPE type algorithms using analytical gradients in, combination with the L-BFGS-B multi-variable optimisation algorithm, provided fast convergence on optimisation targets. The low number of cost function evaluations, when compared with simplex or approximate gradient based methods, have made it possible to use control optimisation to study scalability with system size. As discussed throughout this thesis, the computational expense of a cost function scales with the Hilbert space size or worse, and so this is the limiting factor on the upper bound of system size that can be studied with simulated control optimisation methods. This is much excitement currently about machine-learning based algorithms in many applications. The use of machine learning tools in quantum optimal control is not new, with genetic algorithms in use in 1995 [207]. With this fresh excitement, there is renewed interest in machine learning applied applied to quantum control optimisation. The deep reinforced learning method used by AlphaGo [63], has now been used in quantum optimal control [67, 68]. Machine learning has also be shown to work *in situ* to optimise Bose-Einstein condensate production [69], where they refer to it as ‘online optimization’. These tools have been shown to be effective in time optimal control (finding control pulses that have shorter total evolution times for achieving gate synthesis) and increasing robustness to noise [68]. For specific, difficult to achieve, control objectives, these tools may be best the method. However, there is an overhead in developing the ’learning’, which would have made them infeasible in this study, without having access to significantly more computing resource. If control parameter landscapes have been proven trap-free, which as discussed in Section 1.6.7 is true for the majority of quantum systems, then gradient-based algorithms should out-perform deep-learning based ones.

The *in-situ* optimisation of gates in quantum simulators was shown to scale only polynomially with the number of qubits, making it a viable scheme for developing the set of controls necessary for executing quantum algorithms on such devices. This set of controls only needs to be found once for a given device, assuming that the Hamiltonian, although not fully characterised, does not alter between computations. There is also potential for *in-situ* optimisation of error correction protocols. The collaboration team of this study have also worked on optimising controls that protect logical qubits from environmental noise, including tuneable interactions that allow the system to discover decoherence-free subspaces. Work continues on demonstrating the scalability of the approach, which is challenging to simulate classically due to the doubled space of CPT maps.

The next step is to demonstrate the viability of the *in-situ* optimisation scheme on some small device. A collaboration with the University of California, Berkeley has been arranged for this purpose, where the *in-situ* scheme will be used optimise controls for two-qubit gates and error correction protocols in cavity-coupled transmon qubits. Initial simulation results, using the estimated parameters of the system, have shown promise for good experimental outcomes. The *in-situ* scheme could be applicable to any control target on a quantum simulator, and so could also be used to enhance fidelities of any outcome, so long as the fidelity can be measured efficiently. In fact the classical-quantum hybrid optimisation approach, with the classical computer performing an open-loop optimisation of experimental controls, is applicable to a wide range of quantum experiments. This is the focus of the thesis author's immediate future research, as part of a joint European QuantERA project called 'Theory-blind Quantum Control' or TheBlinQC. In applications where analytical gradients are not accessible, such as in the *in-situ* case, or just difficult to calculate, such as simulating dynamical decoupling with the hierarchical equations of motion, then the additional overhead of calculating approximate gradients makes it less apparent which method will perform best. Collaborators at Imperial College have found promising initial results using Bayesian algorithms for determining trajectories through the control parameter space. These have a lower overhead than training a neural-network in deep learning methods. These will be studied by the thesis author for their suitability to *in-situ* control optimisation.

The hierarchical equations of motion (HEOM) were shown to be a viable method for computing quantum dynamics in the context of control optimisation. An optimisation of a dynamical decoupling (DD) scheme was performed that would be impossible using Lindbladian dynamics. This demonstrates clear potential for other applications of dynamical decoupling. In this study, gate synthesis in many-body Heisenberg interacting systems was found to be exponentially (with system size) more difficult to achieve than with Ising interacting systems. The potential for negating unwanted system interactions was proposed in Ref. [201]. It should be possible to derive potential DD schemes for effectively making Heisenberg systems behave as Ising systems, and optimise these with the tools developed for this study. The study of scaling of the HEOM with system-size remains to be completed. The hierarchy superoperator is already very large before tensoring with the system operator space. As, for qubit systems, the system operator space scales with $2^n$, then this compounds the issue. Additional development is required to paralellise solving of the HEOM, which should be straightforward using OpenMP, as used elsewhere in QuTiP. So, it remains to be seen what is the limit of system-size for control optimisation while using the HEOM dynamics.

In summary, numerical quantum control optimisation has been used to test a variety objectives in different system models, using software tools developed for this study. It has be found to be a valuable method in investigating controllability through characterisation of reachable sets, and also minimum times required to perform operations. The question of scalability in control of many-body systems for practical application in quantum information processing has been addressed through investigating minimum gates times in both discrete and continuous variable systems. In both cases, in specific examples, the scaling was demonstrated to be sub-exponential over the system size range tested, which is limited by the processing time and memory required for simulating many-body quantum systems on a classical computer. There are many directions for potential future study, outlined above, using numerical control optimisation, which this study has demonstrated to be a valid and useful method.

# Appendix A

# Proof of the uncontrollability condition

It should be noted that this proof was first made by Wu, Li, Zhang and Tarn [12] working with the group $\mathrm{SU}(1,1)$. The translation to the symplectic group $\mathrm{Sp}_{2,\mathbb{R}}$, which is isomorphic to $\mathrm{SU}(1,1)$, was primarily completed by Uther Shackerley-Bennett for [128]. It is appended to this thesis for completeness, and adapted slightly for the purpose. The relevant lemmata and definitions are repeated here for the convenience of the reader.

The controlled dynamics operator equation for a single mode with one control operator is

$$\dot{S}(t) = (A + u(t)B)\, S(t), \quad S(0) = \mathbb{I}_2, \tag{A.1}$$

where the control function $u(t)$ is a real valued function defined for all positive times $t$ and $A, B \in \mathfrak{sp}_{2,\mathbb{R}}$.

These operators specify a basis of $\mathfrak{sp}_{2,\mathbb{R}}$:

$$K_x = \frac{1}{2}\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \quad K_y = \frac{1}{2}\begin{pmatrix} -1 & 0 \\ 0 & 1 \end{pmatrix}, \quad K_z = \frac{1}{2}\begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix}, \tag{A.2}$$

with these commutation relations

$$[K_x, K_y] = -K_z, \quad [K_y, K_z] = K_x, \quad [K_z, K_x] = K_y. \tag{A.3}$$

Given $A$ and $B$ from Eq. (A.1), the subset $\Xi$ of $\mathfrak{sp}_{2,\mathbb{R}}$ with elements of the form $A + vB$, $v \in \mathbb{R}$, is called the set of accessible dynamical generators of the system

**Lemma A.1.** *If $\Xi$ only contains hyperbolic elements then Eq. (A.1) is similar, via a symplectic transformation, to*

$$\dot{S}(t) = (-K_x + bK_z + u(t)K_y)S(t), \quad S(0) = \mathbb{I}_2, \tag{A.4}$$

*where constant $b \in \mathbb{R}$, $|b| < 1$.*

The set of accessible controls of Eq. (A.4) is denoted by $\widetilde{\Xi}$, with elements of the form $-K_x + bK_z + vK_y$, $v \in \mathbb{R}$, and its reachable set by $\widetilde{\mathcal{R}}$.

A general $2 \times 2$ real matrix can be written as

$$X = \begin{pmatrix} x_1 + x_3 & x_2 + x_4 \\ x_4 - x_2 & x_1 - x_3 \end{pmatrix}, \quad x_i \in \mathbb{R}. \tag{A.5}$$

**Lemma A.2.** *If $X \in \widetilde{\mathcal{R}}$ then the function*

$$f(X) := (x_1 - x_4)^2 - (x_2 - x_3)^2 \tag{A.6}$$

*satisfies*

$$\begin{aligned} f(X) &\geq 1, \\ \dot{f}(X) &\geq 0, \\ \dot{f}(X = \mathbb{I}) &\geq 1, \end{aligned} \tag{A.7}$$

*for any choice of $u(t)$ in Eq. (A.4).*

In order to prove Lemmata A.1 and A.2, we need a few preliminary statements, which are also taken directly from [12].

**Lemma A.3.** *The equation*

$$\mathrm{Tr}[[M, N]^2] = \mathrm{Tr}[MN]^2 - 2\,\mathrm{Tr}[N^2]\,\mathrm{Tr}[M^2] \tag{A.8}$$

*holds for $M, N \in \mathfrak{sp}_{2,\mathbb{R}}$.*

**Proof** First we expand the elements in the basis defined in Eq. (A.2):

$$M = m_1 K_x + m_2 K_y + m_3 K_z, \tag{A.9}$$

$$N = n_1 K_x + n_2 K_y + n_3 K_z. \tag{A.10}$$

We use this expansion to express the value of the following terms:

$$\mathrm{Tr}[M^2] = \frac{1}{2}(m_1^2 + m_2^2 - m_3^2), \tag{A.11}$$

$$\mathrm{Tr}[N^2] = \frac{1}{2}(n_1^2 + n_2^2 - n_3^2), \tag{A.12}$$

$$\mathrm{Tr}[MN] = \frac{1}{2}(m_1 n_1 + m_2 n_2 - m_3 n_3), \tag{A.13}$$

$$\begin{aligned}\mathrm{Tr}[[M,N]^2] = &\frac{1}{2}((m_2 n_3 - m_3 n_2)^2 \\ &+ (m_3 n_1 - m_1 n_3)^2 \\ &- (m_1 n_2 - m_2 n_1)^2).\end{aligned} \tag{A.14}$$

Then we combine them to prove the statement:

$$\mathrm{Tr}[[M,N]^2] = \mathrm{Tr}[MN]^2 - 2\,\mathrm{Tr}[N^2]\,\mathrm{Tr}[M^2]. \tag{A.15}$$

□

**Lemma A.4.** *If* $\mathrm{Tr}[[A,B]^2] = 0$ *in Eq. (A.1) then the system does not obey the Lie algebra rank criterion.*

**Proof** From Eqs. (A.9) and (A.10) it can be concluded that $M$, $N$ and $[M,N]$ are linearly dependent if and only if

$$\mathrm{Det}\begin{pmatrix} m_1 & n_1 & m_2 n_3 - m_3 n_2 \\ m_2 & n_2 & m_3 n_1 - m_1 n_3 \\ m_3 & n_3 & -(m_1 n_2 - m_2 n_1) \end{pmatrix} = 0, \tag{A.16}$$

or equivalently

$$(m_2 n_3 - m_3 n_2)^2 + (m_3 n_1 - m_1 n_3)^2 - (m_1 n_2 - m_2 n_1)^2 = 0. \tag{A.17}$$

From Eq. (A.14) we see that this is equivalent to $\mathrm{Tr}[[M,N]^2]=0$. If $A$, $B$ and $[A,B]$ are linearly dependent then the span of $A$ and $B$ does not generate $\mathfrak{sp}_{2,\mathbb{R}}$. □

**Lemma A.5.** *Consider hyperbolic $M\in\mathfrak{sp}_{2,\mathbb{R}}$. There exists $P\in\mathrm{Sp}_{2,\mathbb{R}}$ such that $PMP^{-1}=\sqrt{2\,\mathrm{Tr}[M^2]}K_y$.*

**Proof** $M$ is hyperbolic and so $\mathrm{Tr}[M^2]>0$. We seek a matrix $P_1=e^{\alpha K_z}\in\mathrm{Sp}_{2,\mathbb{R}}$ which satisfies

$$P_1MP_1^{-1}=\sqrt{m_1^2+m_2^2}K_y+m_3K_z. \tag{A.18}$$

using the decomposition of Eq. (A.9). Let $\alpha$ be the angle satisfying

$$\sin[\alpha]=\frac{m_1}{\sqrt{m_1^2+m_2^2}},\quad \cos[\alpha]=\frac{m_2}{\sqrt{m_1^2+m_2^2}}. \tag{A.19}$$

According to the formula

$$e^MNe^{-M}=N+[M,N]+\frac{1}{2!}[M,[M,N]]+\dots, \tag{A.20}$$

one can immediately obtain that

$$\begin{aligned} e^{\alpha K_z}Me^{-\alpha K_z}&=m_1e^{\alpha K_z}K_xe^{-\alpha K_z}+m_2e^{\alpha K_z}K_ye^{-\alpha K_z}+m_3K_z\\ &=(m_1\cos[\alpha]-m_2\sin[\alpha])K_x+(m_1\sin[\alpha]+m_2\cos[\alpha])K_y+m_3K_z\\ &=\sqrt{m_1^2+m_2^2}K_y+m_3K_z. \end{aligned} \tag{A.21}$$

Next we show that there is a matrix $P_2=e^{\beta K_x}\in\mathrm{Sp}_{2,\mathbb{R}}$ which can convert $\sqrt{m_1^2+m_2^2}K_y+m_3K_z$ into $\sqrt{2\,\mathrm{Tr}[M^2]}K_y$. Using Eq. (A.11) we know that $m_1^2+m_2^2-m_3^2>0$ and so we can choose $\beta$ such that

$$\sinh[\beta]=\frac{m_3}{\sqrt{m_1^2+m_2^2-m_3^2}},\quad \cosh[\beta]=\frac{\sqrt{m_1^2+m_2^2}}{\sqrt{m_1^2+m_2^2-m_3^2}}. \tag{A.22}$$

Make use of Eq. (A.20) again and obtain

$$\begin{aligned} e^{\beta K_x}(\sqrt{m_1^2+m_2^2}K_y+m_3K_z)e^{-\beta K_x} &= \sqrt{m_1^2+m_2^2}e^{\beta K_x}K_ye^{-\beta K_x}+m_3e^{\beta K_x}K_ze^{-\beta K_x} \\ &= (\sqrt{m_1^2+m_2^2}\cosh[\beta]-m_3\sinh[\beta])K_y \\ &\quad +(m_3\cosh[\beta]-\sqrt{m_1^2+m_2^2}\sinh[\beta])K_z \\ &= \sqrt{m_1^2+m_2^2-m_3^2}K_y \\ &= \sqrt{2\operatorname{Tr}[M^2]}K_y. \end{aligned} \tag{A.23}$$

Consequently the $\mathrm{Sp}_{2,\mathbb{R}}$ matrix $e^{\beta K_x}e^{\alpha K_z}$ will convert $M$ into $\sqrt{2\operatorname{Tr}[M^2]}K_y$ when $M$ is hyperbolic. □

Using the two previous lemmas we may now proceed to a proof of Lemma A.1 that was stated in the main text. First we restate it.

**Lemma A.1.** *If $\Xi$ only contains hyperbolic elements then Eq. (A.1) is similar, via a symplectic transformation, to*

$$\dot{S}(t)=(-K_x+bK_z+u(t)K_y)S(t), \quad S(0)=\mathbb{I}_2, \tag{A.24}$$

*where $b$ is some real constant with modulus strictly less than one.*

**Proof** If Eq. (A.1) only has hyperbolic controls then the following inequality holds:

$$\operatorname{Tr}[(A+vB)^2]=\operatorname{Tr}[B^2]v^2+2\operatorname{Tr}[AB]v+\operatorname{Tr}[A^2]>0, \tag{A.25}$$

for all $v\in\mathbb{R}$. For this inequality to hold for all $v$ it is immediately clear that $\operatorname{Tr}[A^2]>0$. We can see that $\operatorname{Tr}[B^2]>0$ because (a) if it were less than zero then there exists $v$ for which the inequality does not hold and (b) if it were equal to zero then $\operatorname{Tr}[AB]$ must equal zero; by Lemma A.3 this implies that $\operatorname{Tr}([A,B]^2)=0$ which implies that the system does not obey the Lie algebra rank criterion by Lemma A.4 which would contradict our assumption.

With the knowledge that $B$ is hyperbolic, Lemma A.5 states that there exists a symplectic similarity transformation to transform Eq. (A.1) into:

$$\dot{S}(t) = (A' + u(t)K_y)S(t), \quad S(0) = \mathbb{I}_2, \tag{A.26}$$

where $A'$ is some unspecified element of $\mathfrak{sp}_{2,\mathbb{R}}$. Expand $A'$ in the symplectic basis of Eq. (A.2):

$$A' = b_x K_x + b_y K_y + b_z K_z. \tag{A.27}$$

By redefining $u(t)$ we can transform the system such that $b_y$ equals zero. We know that $A'$ is hyperbolic because this property is invariant under similarity transformation, therefore we know that $|b_x| > |b_z|$ from Eq. (A.11). The role of time in Eq. (A.26) allows us to rescale such that the coefficient of $K_x$ has modulus one leaving us with system

$$\dot{S}(t) = (\epsilon K_x + bK_z + u(t)K_y)S(t), \quad S(0) = \mathbb{I}_2, \tag{A.28}$$

where $|b| < 1$ and $\epsilon = \pm 1$. If $\epsilon = -1$ then we leave the system as it is and the proof is finished. If $\epsilon = 1$ then enacting a similarity transformation under the symplectic matrix $\Omega$ is equivalent to time reversal and sends each of the basis matrices to their negative. Thus we have shown that Eq. (A.1) is symplectically similar to Eq. (A.24). Note that we did not talk about effects on the initial value of $X$ because this is *set* to be $\mathbb{I}_2$. □

We are now able to proceed to a proof of Lemma A.2 stated above. First we restate it.

**Lemma A.2.** *If $X \in \widetilde{\mathcal{R}}$ then the function*

$$f(X) := (x_1 - x_4)^2 - (x_2 - x_3)^2 \tag{A.29}$$

*satisfies*

$$f(X) \geq 1, \tag{A.30}$$

$$\frac{\mathrm{d}}{\mathrm{d}t} f(X) \geq 0, \tag{A.31}$$

$$\frac{\mathrm{d}}{\mathrm{d}t} f(X = \mathbb{I}) \geq 1, \tag{A.32}$$

*for any choice of* $u(t)$ *in Eq. (A.1).*

**Proof** Eqs. (A.24) and (A.5) provide the set of equations

$$\dot{x}_1 = \frac{1}{2}(ax_2 - x_4 - ux_3), \tag{A.33}$$

$$\dot{x}_2 = \frac{1}{2}(-ax_1 + x_3 - ux_4), \tag{A.34}$$

$$\dot{x}_3 = \frac{1}{2}(-ax_4 + x_2 - ux_1), \tag{A.35}$$

$$\dot{x}_4 = \frac{1}{2}(ax_3 - x_1 - ux_2). \tag{A.36}$$

Subtracting Eqs. (A.33) and (A.36) then followed by a succeeding multiplication by $2(x_1 - x_4)$ provides

$$\begin{aligned}\frac{\mathrm{d}}{\mathrm{d}t}(x_1 - x_4)^2 = & \, a(x_1 - x_4)(x_2 - x_3) \\ & + (x_1 - x_4)^2 \\ & + u(x_1 - x_4)(x_2 - x_3).\end{aligned} \tag{A.37}$$

Similarly, we have

$$\begin{aligned}\frac{\mathrm{d}}{\mathrm{d}t}(x_2 - x_3)^2 = & - a(x_1 - x_4)(x_2 - x_3) \\ & - (x_2 - x_3)^2 \\ & + u(x_1 - x_4)(x_2 - x_3).\end{aligned} \tag{A.38}$$

Then subtracting Eqs. (A.37) and (A.38)

$$\begin{aligned}& \frac{\mathrm{d}}{\mathrm{d}t}\left((x_1 - x_4)^2 - (x_2 - x_3)^2\right) \\ & = \; 2a(x_1 - x_4)(x_2 - x_3) \\ & \quad\quad + \left((x_1 - x_4)^2 + (x_2 - x_3)^2\right) \\ & = \; (1 - |a|)\left((x_1 - x_4)^2 + (x_2 - x_3)^2\right) \\ & \quad\quad + |a|\left((x_1 - x_4) - sign(a)(x_2 - x_3)\right)^2 \\ & \geq 0\,.\end{aligned} \tag{A.39}$$

Thus, the function $f$ is nondecreasing for every trajectory of the system. Since the initial value of $f$ is 1 it can be concluded that the reachable states of Eq. (A.24) should satisfy the restriction that

$f \geq 1$. Furthermore the initial value of the rate of change is equal to one if we set $x_1 = 1$ and $x_2 = x_3 = x_4 = 0$. This implies that the function is increasing from the beginning. □

# Appendix B

# Symplectic singular value decomposition

## B.1 Uniqueness of the singular value decomposition

Alexander Pitchford and Uther Shackerley-Bennett independently found bounds on the ranges $\theta$ and $\phi$ to ensure the uniqueness of the decomposition. The proof and its presentation in this form was completed by Uther Shackerley-Bennett for the study reported in [128].

To prevent any ambiguity we require that the singular value decomposition be unique. This is not true in general and therefore we need to restrict the range of allowed angles so that it is properly defined. In short, we want

$$S = R_\theta Z R_\phi = R_\alpha Z' R_\beta \tag{B.1}$$

to imply that $\alpha = \theta$, $\beta = \phi$ and $Z' = Z$. The first thing to notice is that the singular values of $S$ are unique and so we would only ever get either $Z' = Z$ or $Z' = Z^{-1}$. The latter case corresponds to the situation where $z < 1$ which may be ignored provided the range of the angles is properly limited allowing $Z^{-1} = R_{-\pi/2} Z R_{\pi/2}$. Thus we need only consider two cases, $z = 1$ and $z > 1$. In the conclusion we use these cases to show that we have a freedom in how to represent the singular value decomposition.

### B.1.1 $Z \neq \mathbb{I}$

Let's first look at the former case, $Z' = Z$, where $Z \neq \mathbb{I}$. Assume a non-unique decomposition:

$$R_\theta Z R_\phi = R_\alpha Z R_\beta, \tag{B.2}$$

or equivalently

$$R_{\theta-\alpha} Z = Z R_{\beta-\phi}, \tag{B.3}$$

and explicitly

$$\begin{pmatrix} \frac{1}{z}\cos[\theta-\alpha] & -z\sin[\theta-\alpha] \\ \frac{1}{z}\sin[\theta-\alpha] & z\cos[\theta-\alpha] \end{pmatrix} = \begin{pmatrix} \frac{1}{z}\cos[\beta-\phi] & -\frac{1}{z}\sin[\beta-\phi] \\ z\sin[\beta-\phi] & z\cos[\beta-\phi] \end{pmatrix}. \tag{B.4}$$

This implies the set of conditions

$$\begin{aligned} \frac{1}{z}\sin[\theta-\alpha] &= z\sin[\beta-\phi], \\ z\sin[\theta-\alpha] &= \frac{1}{z}\sin[\beta-\phi], \\ \cos[\theta-\alpha] &= \cos[\beta-\phi], \end{aligned} \tag{B.5}$$

which only hold when

$$\begin{aligned} \sin[\theta-\alpha] &= 0, \\ \sin[\beta-\phi] &= 0, \\ \cos[\theta-\alpha] &= \cos[\beta-\phi]. \end{aligned} \tag{B.6}$$

These only hold when

$$\alpha = \theta + n\pi \quad \text{and} \quad \beta = \phi + m\pi \tag{B.7}$$

for $n, m \in \mathbb{Z}$ either both odd or both even.

To avoid Eq. (B.7) being satisfied for $m, n \neq 0$ we limit $\phi$ to vary in a range less than $\pi$ so that $\beta = \phi$. This sets $m = 0$ and so to satisfy Eq. (B.7) without letting $\alpha = \theta$ the nearest option would be to let $\alpha = \theta \pm 2\pi$. The maximum range for the angles governing $\mathrm{SO}(2)$ is $2\pi$ and so this is the bound that will apply to $\theta$. For uniqueness, therefore, we set the ranges of $\theta$ and $\phi$ to:

$$-\pi + \theta_0 \leq \theta < \pi + \theta_0, \quad -\frac{\pi}{2} + \phi_0 \leq \phi < \frac{\pi}{2} + \phi_0, \tag{B.8}$$

where $\theta_0$, $\phi_0$ fix the centre of the ranges.

### B.1.2 $Z = \mathbb{I}$

In this case we consider $Z = \mathbb{I}$. We look for times when

$$R_\theta R_\phi = R_\alpha R_\beta \tag{B.9}$$

is satisfied.

These are cases when

$$\theta + \phi = \alpha + \beta + 2n\pi, \tag{B.10}$$

for $n \in \mathbb{Z}$.

This holds true for a whole range of angles. We can arbitrarily set $\phi = \phi_0$ to let $\theta$ label the elements of $\mathrm{SO}(2)$.

### B.1.3 Angle limit

Now we have choices on how to set the angles such that the decomposition is unique. We choose

$$-\pi + \theta_0 \leq \theta < \pi + \theta_0, \quad -\frac{\pi}{2} + \phi_0 \leq \phi < \frac{\pi}{2} + \phi_0 \tag{B.11}$$

to make the singular value decomposition unique when $Z \neq \mathbb{I}$. $\theta_0$ and $\phi_0$ are some constants that we are free to set. Note that we have made a further arbitrary choice in exactly where to make the bounds tight. For $Z = \mathbb{I}$ we must totally restrict one of the angles and leave the other free; we choose so set $\phi = \phi_0$.

## B.2 Singular value decomposition coordinates for $f$

Alexander Pitchford and Uther Shackerley-Bennett independently derived the coordinate transformation. The presentation in this form was completed by Uther Shackerley-Bennett for [128].

In this section we represent $\cos[\theta]$ as $c\theta$ and $\sin[\theta]$ as $s\theta$ for brevity. We begin with two expressions for $X \in \mathrm{Sp}_{2,\mathbb{R}}$:

$$X = \begin{pmatrix} x_1 + x_3 & x_2 + x_4 \\ x_4 - x_2 & x_1 - x_3 \end{pmatrix}, \tag{B.12}$$

and

$$X = \begin{pmatrix} \frac{c\theta c\phi}{z} - zs\theta s\phi & -\frac{c\theta s\phi}{z} - zs\theta c\phi \\ \frac{s\theta c\phi}{z} + zc\theta s\phi & -\frac{s\theta s\phi}{z} + zc\theta c\phi \end{pmatrix}. \tag{B.13}$$

Equating the two expression and solving for $x_i$ we find that

$$2x_1 = \frac{1}{z}(c\theta c\phi - s\theta s\phi) + z(c\theta c\phi - s\theta s\phi), \tag{B.14}$$

$$2x_2 = -\frac{1}{z}(s\theta c\phi + c\theta s\phi) - z(s\theta c\phi + c\theta s\phi), \tag{B.15}$$

$$2x_3 = \frac{1}{z}(c\theta c\phi + s\theta s\phi) - z(s\theta s\phi + c\theta c\phi), \tag{B.16}$$

$$2x_4 = \frac{1}{z}(s\theta c\phi - c\theta s\phi) + z(c\theta s\phi - s\theta c\phi), \tag{B.17}$$

and so

$$2(x_1 - x_4) = \frac{1}{z}(c\theta c\phi - s\theta s\phi - s\theta c\phi + c\theta s\phi) + z(c\theta c\phi - s\theta s\phi - c\theta s\phi + s\theta c\phi), \tag{B.18}$$

$$2(x_2 - x_3) = -\frac{1}{z}(s\theta c\phi + c\theta s\phi + c\theta c\phi + s\theta s\phi) - z(s\theta c\phi + c\theta s\phi - s\theta s\phi - c\theta c\phi), \tag{B.19}$$

or more simply

$$\begin{aligned} 2(x_1 - x_4) &= \frac{1}{z}(c\theta - s\theta)(c\phi + s\phi) + z(c\theta + s\theta)(c\phi - s\phi), \\ 2(x_2 - x_3) &= -\frac{1}{z}(c\theta + s\theta)(c\phi + s\phi) - z(c\theta - s\theta)(-c\phi + s\phi), \end{aligned} \tag{B.20}$$

which leads to

$$\begin{aligned}(x_1-x_4)^2 &= \frac{1}{4}\bigg(\frac{1}{z^2}(c\theta-s\theta)^2(c\phi+s\phi)^2+z^2(c\theta+s\theta)^2(c\phi-s\phi)^2\\ &\qquad +2(c\theta+s\theta)(c\theta-s\theta)(c\phi+s\phi)(c\phi-s\phi)\bigg),\\ (x_2-x_3)^2 &= \frac{1}{4}\bigg(\frac{1}{z^2}(c\theta+s\theta)^2(c\phi+s\phi)^2+z^2(c\theta-s\theta)^2(c\phi-s\phi)^2\\ &\qquad -2(c\theta+s\theta)(c\theta-s\theta)(c\phi+s\phi)(c\phi-s\phi)\bigg).\end{aligned} \tag{B.21}$$

Subtracting the two

$$\begin{aligned}(x_1-x_4)^2-(x_2-x_3)^2 = \frac{1}{4}\bigg(&\frac{1}{z^2}(c\phi+s\phi)^2\big((c\theta-s\theta)^2-(c\theta+s\theta)^2\big)+\\ &z^2(c\phi-s\phi)^2\big((c\theta+s\theta)^2-(c\theta-s\theta)^2\big)+\\ &4(c\theta^2-s\theta^2)(c\phi^2-s\phi^2)\bigg),\end{aligned} \tag{B.22}$$

to

$$(x_1-x_4)^2-(x_2-x_3)^2 = \frac{1}{4}\bigg(\frac{1}{z^2}(1+s2\phi)(-s2\theta)+z^2(1-s2\phi)(s2\theta)+4c2\theta c2\phi\bigg), \tag{B.23}$$

to

$$(x_1-x_4)^2-(x_2-x_3)^2 = c2\theta c2\phi - s2\theta\left(\frac{1}{2}\left(z^2+\frac{1}{z^2}\right)s2\phi-\frac{1}{2}\left(z^2-\frac{1}{z^2}\right)\right). \tag{B.24}$$

which is our new expression for $f$ in terms of $\theta$, $\phi$ and $z$.

# Appendix C

# Proof reachable set excludes passives

Uther Shackerley-Bennett derived the proofs in this appendix as part of the study reported in [128]. They are included in this thesis for completeness. The presentation is adapted for the purpose.

The controlled dynamics operator equation for a single mode with one control operator is

$$\dot{S}(t) = (A + u(t)B)\, S(t), \quad S(0) = \mathbb{I}_2, \tag{C.1}$$

where the control function $u(t)$ is a real valued function defined for all positive times $t$ and $A, B \in \mathfrak{sp}_{2,\mathbb{R}}$.

These operators specify a basis of $\mathfrak{sp}_{2,\mathbb{R}}$:

$$K_x = \frac{1}{2}\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \quad K_y = \frac{1}{2}\begin{pmatrix} -1 & 0 \\ 0 & 1 \end{pmatrix}, \quad K_z = \frac{1}{2}\begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix}, \tag{C.2}$$

with these commutation relations

$$[K_x, K_y] = -K_z, \quad [K_y, K_z] = K_x, \quad [K_z, K_x] = K_y. \tag{C.3}$$

In this basis the controlled dynamics equation for hyperbolic systems is:

$$\dot{S}(t) = (-K_x + bK_z + u(t)K_y)S(t), \quad S(0) = \mathbb{I}_2, \tag{C.4}$$

where constant $b \in \mathbb{R}, |b| < 1$. The reachable set of Eq. (C.4) is denoted by $\widetilde{\mathcal{R}}$.

The following is a restatement of Lemma A.2 (proof in Appendix A) in the singular value decomposition coordinates, using the transformation derived in Appendix B.2.

**Corollary C.1.** *If $X \in \widetilde{\mathcal{R}}$ then the function*

$$g(X) := \cos[2\theta]\cos[2\phi] - \lambda(z,\phi)\sin[2\theta]\,, \tag{C.5}$$

*where*

$$\lambda(z,\phi) := \frac{1}{2}\left(z^2 + \frac{1}{z^2}\right)\sin[2\phi] - \frac{1}{2}\left(z^2 - \frac{1}{z^2}\right), \tag{C.6}$$

*satisfies*

$$g(X) \geq 1, \tag{C.7}$$

$$\dot{g}(X) \geq 0, \tag{C.8}$$

$$\dot{g}(X = \mathbb{I}) \geq 1. \tag{C.9}$$

*for any choice of $u(t)$ in Eq. (C.4).*

To gain some analytical insight into unstable systems we use Corollary C.1 to provide some bounds on $\widetilde{\mathcal{R}}$.

**Lemma C.1.** *The existence of solutions for $g(X) > d$, where $d \geq 1$, implies that*

$$z > \sqrt{\frac{d+1}{2}}. \tag{C.10}$$

**Proof** First we prove that $g(X) > 1$ implies that $\sin[2\theta] \geq 0$. Define

$$\delta := \lambda(z,\phi) - \sin[2\phi], \tag{C.11}$$

which allows one to rewrite $g(X)$ as

$$g(X) \equiv \cos[2(\theta - \phi)] - \delta\sin[2\theta]. \tag{C.12}$$

If $\sin[2\theta] < 0$ then $g(X) > 1$ only has solutions if $\delta > 0$. This is only true if $\lambda(z, \phi) > \sin[2\phi]$. Hence

$$\frac{1}{2}\left(z^2 + \frac{1}{z^2}\right)\sin[2\phi] - \frac{1}{2}\left(z^2 - \frac{1}{z^2}\right) > \sin[2\phi], \tag{C.13}$$

equally

$$\left(z^2 + \frac{1}{z^2} - 2\right)\sin[2\phi] > z^2 - \frac{1}{z^2}. \tag{C.14}$$

$z^2 + 1/z^2 - 2$ is positive for all values of $z$ and $\sin[2\phi] \leq 1$. Therefore this has only solutions if

$$z^2 + \frac{1}{z^2} - 2 > z^2 - \frac{1}{z^2}, \tag{C.15}$$

implying $z < 1$ which we have ruled out by convention. Proving that $\sin[2\theta] \geq 0$.

Now we look for existence of solutions to the inequality $g(X) > d$. Because $\sin[2\theta] \geq 0$ these exist if and only if there exist solutions to

$$\delta < -(d-1). \tag{C.16}$$

This translates to

$$\frac{1}{2}\left(z^2 + \frac{1}{z^2}\right)\sin[2\phi] - \frac{1}{2}\left(z^2 - \frac{1}{z^2}\right) < \sin[2\phi] - (d-1), \tag{C.17}$$

equally

$$(z^2 + \frac{1}{z^2} - 2)\sin[2\phi] < z^2 - \frac{1}{z^2} - (d-1). \tag{C.18}$$

$z^2 + \frac{1}{z^2} - 2$ is positive for all values of $z$ and $\sin[2\phi] \leq 1$. Therefore this has solutions if and only if

$$z^2 + \frac{1}{z^2} - 2 < z^2 - \frac{1}{z^2} - (d-1), \tag{C.19}$$

which only has solutions for

$$z^2 > \frac{d+1}{2}, \tag{C.20}$$

proving the statement. □

Lemma C.1 links lower bounds on $g(X)$ to lower bounds on $z$ which will be used to further understand the boundary of the reachable set when considered in $(z, \theta, \phi)$ space.

**Lemma C.2.** *There does not exist* $X \in \widetilde{\mathcal{R}}$ *such that*

$$X = SR_\theta S^{-1}, \tag{C.21}$$

*where* $S \in \mathrm{Sp}_{2,\mathbb{R}}$, $R_\theta \in \mathrm{SO}(2)$.

**Proof** Assume there exists $X \in \widetilde{\mathcal{R}}$ that satisfies the above condition. We know that

$$X^m \in \widetilde{\mathcal{R}} \quad \forall m \in \mathbb{N} \tag{C.22}$$

because the reachable set of Eq. (C.4) has a semigroup structure. Note that

$$\begin{aligned} \|X^m - \mathbb{I}\| &= \|S(R_\theta^m - \mathbb{I})S^{-1}\| \\ &\leq \|S\|\|S^{-1}\|\|R_\theta^m - \mathbb{I}\|, \end{aligned} \tag{C.23}$$

where we use the Euclidean norm

$$\|X\| := \sqrt{\mathrm{Tr}[X^\mathsf{T} X]}. \tag{C.24}$$

$S$ is time-independent and so $||S||||S^{-1}||$ is constant. $R_\theta$ is quasi-periodic and so there must exist some $m$ such that

$$\|R_\theta^m - \mathbb{I}\| < \varepsilon, \quad \forall \varepsilon > 0 \tag{C.25}$$

and so there exists $m$ such that

$$\|X^m - \mathbb{I}\| < \varepsilon, \quad \forall \varepsilon > 0. \tag{C.26}$$

From Eq. (3.36) we know that the value of $g(X)$ must be non-decreasing along any trajectory of the system and from Eq. (3.37) we know that its rate of change at identity is $1$. As a result, for some finite evolution time of Eq. (C.4) all subsequent trajectories must contain elements that have a lower bound on their value of $g(X)$ that is greater than $1$. By Lemma C.1 this implies a lower bound on the value $z$ along a given trajectory of the control system given some minimal evolution time. $X^m$ is a possible trajectory of the system for all $m$ and we can find $m$ such that the $z$ value of $X^m$ is arbitrarily close to $1$ violating the lower bound. Therefore such an $X$ cannot be an element of $\widetilde{\mathcal{R}}$. □

**Theorem C.1.** *If Eq. (C.1) is restricted to hyperbolic dynamical generators then its reachable set does not contain any elements of* $\mathrm{SO}(2)$ *except for* $\mathbb{I}$.

**Proof** The reachable set $\mathcal{R}$ of Eq. (C.1) is symplectically similar to $\widetilde{\mathcal{R}}$. Lemma C.2 states that $\widetilde{\mathcal{R}}$ does not contain any element that is symplectically similar to an element of $\mathrm{SO}(2) \setminus \mathbb{I}$. Thus $\mathcal{R}$ does not contain any element of $\mathrm{SO}(2) \setminus \mathbb{I}$. □

# Appendix D

# Bound of the local estimator fidelity

This derivation of this local estimator to the fidelity, and the proof that it bounds the gate fidelity, based on methods from [169], was completed by Benjamin Dive and is presented in [145] and in more detail in [172]. It is included here for completeness, with notation adapted for this thesis.

The local estimator to the gate fidelity $\mathcal{F}_{\text{LE}}$ is a lower bound on the true gate fidelity, as will be shown below. Firstly the gate fidelity in terms of Choi states given in Eq. (4.2) is defined again here for a CPT map $M$ and unitary operator for comparison $U$, which in this context is the target operation.

$$\mathcal{F}(M,U) = \langle\psi|\rho_M|\psi\rangle \, , \tag{D.1}$$

where $|\psi\rangle = (U \otimes \mathbb{I})|\Phi\rangle$ and $\rho_M$ is the Choi state of $M$, and $|\Phi\rangle$ is maximally entangled state between the original Hilbert space and a copy of it. These are more explicitly defined in Subsection 4.1.1.

The local estimator is only applicable to target unitaries with a tensor product structure such that $U = \bigotimes_i U_i$. The Choi state of $U$ inherits the tensor product structure, that is $|\psi\rangle = \bigotimes_i |\psi_i\rangle$. The first step in finding a bound for the gate fidelity is to define the projectors $h_i = \mathbb{I}_i - |\psi_i\rangle\langle\psi_i|$ for each $U_i$. These projectors have a very simple spectrum with a single $0$ eigenvalue with corresponding eigenket $|\psi_i\rangle$, and a degenerate orthogonal space with eigenvalue $1$. These projectors are summed together to form a Hamiltonian $H = \sum_i h_i \otimes \mathbb{I}_{\bar{i}}$ such that each $h_i$ acts on its own part of the Hilbert space and is identity on the rest (the bar over the $i$ indicates all subsystems except $i$). This has a single $E_0 = 0$ eigenvalue with eigenstate $|E_0\rangle = |\psi\rangle$, while all its other eigenvalues are positive integers. Expanding this Hamiltonian in its eigenbasis $\{E_k, |E_k\rangle\}$ gives

$$\begin{aligned} \operatorname{Tr} H\rho &= \sum_{k\geq 0} E_k \langle E_k|\rho|E_k\rangle \\ &\geq \sum_{k>0} \langle E_k|\rho|E_k\rangle \,, \end{aligned} \tag{D.2}$$

as $E_0 = 0$ and all the other energies are one or greater. Using this and properties of the density operator provides the identity

$$\operatorname{Tr} \rho = \sum_{k\geq 0} \langle E_k|\rho|E_k\rangle = 1 \,, \tag{D.3}$$

which allows the inequality of Eq. (D.2) to be expressed as

$$\begin{aligned} \operatorname{Tr} H\rho &\geq 1 - \langle E_0|\rho|E_0\rangle \\ &\geq 1 - \langle \psi|\rho|\psi\rangle \,. \end{aligned} \tag{D.4}$$

Using the definition of the gate fidelity from Eq. (D.1) with $\rho$ taken as the Choi state of $M$ gives a bound

$$F(M, U) \geq 1 - \operatorname{Tr} H\rho \,. \tag{D.5}$$

The expectation value of the Hamiltonian acting on the Choi state, based on the definition of the Choi state, can be evaluated as follows

$$\operatorname{Tr} H\rho = \sum_i \operatorname{Tr} \left(\mathbb{I}_i - |\psi_i\rangle\langle\psi_i|\right) \otimes \mathbb{I}_{\bar{i}} \ \rho \tag{D.6}$$

$$= \sum_i \mathrm{Tr}_i[(\mathbb{I}_i - |\psi_i\rangle\langle\psi_i|)\rho_i] \tag{D.7}$$

$$= \sum_i (1 - \langle\psi_i|\rho_i|\psi_i\rangle) \,, \tag{D.8}$$

where $\rho_i = \mathrm{Tr}_{\bar{i}}[\rho]$ is the Choi state of the map

$$M_i(\,\cdot\,) \equiv M(\,\cdot \bigotimes_{j\neq i} \tfrac{1}{d_j}\mathbb{I}_j), \tag{D.9}$$

which is the map $M$ acting on subsystem $i$ with the other subsystems in the maximally mixed state.

This results in

$$F(M,U) \geq 1 - \sum_i (1 - F(M_i, U_i))\,, \tag{D.10}$$

which describes the bound on the gate fidelity, the right hand side being called the *local estimator* $\mathcal{F}_{\mathrm{LE}}$.

# Appendix E

# Derivation of pure dephasing

This derivation is included here because it goes slightly beyond the one in [22] in terms of its application. It is also based on the model from one of the landmark HEOM papers [176]. It was completed by Neill Lambert. The notation has been adapted for this thesis.

## E.1 Pure dephasing – exact

Starting from the expression for the propagator of the reduced density matrix

$$\rho(t) = \mathcal{U}(t)\rho(0)\,, \tag{E.1}$$

which in this case, adapted from Eq.(2.7) of [176],

$$\mathcal{U}(t) = \exp\left[-\frac{1}{\hbar^2}\int_0^t ds \int_0^s dy \sigma_z^{\times}\left\{S(s-y)\sigma_z^{\times} - i\frac{\hbar}{2}\chi(s-y)\sigma_z^{o}\right\}\right]\,, \tag{E.2}$$

is in the interaction picture. Here there is a single batch and the system operator which couples to the bath $\sigma_z$ is not time dependent. The hyperoperator notation $\emptyset^{\times}$ indicates the commutator, *e.g.* $\emptyset^{\times}\rho \equiv [O, \rho]$. Similarly $\emptyset^{o}$ indicates the anticommutator.

Because the system operators in the above are time independent in the interaction picture, there is no time ordering difficulty, and the integral can be evaluated directly. Using

$$\begin{aligned} S(t) = & \quad \frac{\hbar}{\pi}\int_0^\infty d\omega J(\omega)\coth\frac{\beta\hbar\omega}{2}\cos\omega t, \\ \chi(t) = & \qquad\qquad \frac{2}{\pi}\int_0^\infty d\omega J(\omega)\sin\omega t\,. \end{aligned} \tag{E.3}$$

Inserting these in Eq. (E.1), both time integrals can be evaluated, leaving

$$\mathcal{U}(t) = \exp\left[-\frac{1}{\hbar^2}\int_0^\infty d\omega\sigma_z^\times\left\{\frac{\hbar}{\pi}J(\omega)\coth\frac{\beta\hbar\omega}{2}\left(\frac{1-\cos(\omega t)}{\omega^2}\right)\sigma_z^\times - i\frac{\hbar}{\pi}J(\omega)\left(\frac{\sin(\omega t)-\omega t}{\omega^2}\right)\sigma_z^o\right\}\right] \tag{E.4}$$

As $\sigma_z^\times\sigma_z^o = 0$ and

$$\begin{aligned} \sigma_z^\times\sigma_z^\times &= 2(\mathbb{I} - \sigma_z\cdot\sigma_z) \\ &= 4(|e\rangle\langle e|\cdot|g\rangle\langle g| + |g\rangle\langle g|\cdot|e\rangle\langle e|) \end{aligned} \tag{E.5}$$

This gives us the time-dependence of an off-diagonal matrix element as

$$\langle e|\rho(t)|g\rangle = \exp\left[-\frac{1}{\pi\hbar}\int_0^\infty d\omega\left\{4J(\omega)\coth\frac{\beta\hbar\omega}{2}\left(\frac{1-\cos(\omega t)}{\omega^2}\right)\right\}\right]\langle e|\rho(0)|g\rangle\,. \tag{E.6}$$

Note that the $1/\pi$ factor may be included within the spectral density parameters in some cases.

# Bibliography

[1] Steven Conolly, Dwight Nishimura, and Albert Macovski. “Optimal Control Solutions to the Magnetic Resonance Selective Excitation Problem”. In: *IEEE Transactions on Medical Imaging* 5.2 (June 1986), pp. 106–115. DOI: 10.1109/TMI.1986.4307754.

[2] Jintong Mao, T.H Mareci, K.N Scott, and E.R Andrew. “Selective inversion radiofrequency pulses by optimal control”. In: *Journal of Magnetic Resonance (1969)* 70.2 (Nov. 1986), pp. 310–318. DOI: 10.1016/0022-2364(86)90016-8.

[3] David J. Tannor and Stuart A. Rice. “Control of selectivity of chemical reaction via control of wave packet evolution”. In: *The Journal of Chemical Physics* 83.10 (Nov. 1985), pp. 5013–5018. DOI: 10.1063/1.449767.

[4] Paul Brumer and Moshe Shapiro. “Control of unimolecular reactions using coherent light”. In: *Chemical Physics Letters* 126.6 (May 1986), pp. 541–546. DOI: 10.1016/S0009-2614(86)80171-3.

[5] J. Kelly et al. “Optimal quantum control using randomized benchmarking”. In: *Physical Review Letters* 112.24 (June 2014), p. 240504. DOI: 10.1103/PhysRevLett.112.240504. arXiv: 1403.0035.

[6] J. Kelly et al. “Scalable in situ qubit calibration during repetitive error detection”. In: *Physical Review A* 94.3 (2016), pp. 1–8. DOI: 10.1103/PhysRevA.94.032321. arXiv: 1603.03082.

[7] Paul A. Benioff. “Quantum mechanical Hamiltonian models of discrete processes that erase their own histories: Application to Turing machines”. In: *International Journal of Theoretical Physics* 21.3-4 (Apr. 1982), pp. 177–201. DOI: 10.1007/BF01857725.

[8] Richard P. Feynman. “Quantum Mechanical Computers”. In: *Optics News* 11.2 (Feb. 1985), p. 11. DOI: 10.1364/ON.11.2.000011.

[9] Seth Lloyd and Samuel L. Braunstein. “Quantum computation over continuous variables”. In: *Physical Review Letters* 82.8 (Feb. 1999), pp. 1784–1787. DOI: 10.1103/PhysRevLett.82.1784. arXiv: 9810082 [quant-ph].

[10] V. Jurdjevic and H. J. Sussmann. “Control systems on Lie groups.” In: *Journal of Differential Equations* 12.2 (1972), pp. 313–329. DOI: 10.1016/0022-0396(72)90035-6.

[11] Re Bing Wu, Tzyh Jong Tarn, and Chun Wen Li. “Smooth controllability of infinite-dimensional quantum-mechanical systems”. In: *Physical Review A - Atomic, Molecular, and Optical Physics* 73.1 (2006). DOI: 10.1103/PhysRevA.73.012719. arXiv: 0505063 [quant-ph].

[12] J.W. Wu, C.W. Li, Jing Zhang, and T.J. Tarn. “Controllability of Quantum Systems on the Lie Group SU (1, 1)”. In: *Arxiv preprint arXiv:0708.3147* (2007), pp. 1–12. arXiv: 0708.3147.

[13] Marco Genoni, Alessio Serafini, M Kim, and Daniel Burgarth. “Dynamical Recurrence and the Quantum Control of Coupled Oscillators”. In: *Physical Review Letters* 108.15 (2012), pp. 1–5. DOI: 10.1103/PhysRevLett.108.150501. arXiv: 1110.5584.

[14] Rebing Wu, Raj Chakrabarti, and Herschel Rabitz. “Optimal Control Theory for Continuous Variable Quantum Gates”. In: *Physical Review A* 77.5 (2007), p. 39.

[15] Maximilian Schlosshauer. “Decoherence, the Measurement Problem, and Interpretations of Quatum Mechanics”. In: *Reviews of Modern Physics* 76.4 (2004), pp. 1267–1305.

[16] J. R. Johansson, P. D. Nation, and Franco Nori. “QuTiP: An open-source Python framework for the dynamics of open quantum systems”. In: *Computer Physics Communications* 183.8 (2012), pp. 1760–1772. DOI: 10.1016/j.cpc.2012.02.021. arXiv: 1110.0573.

[17] J. R. Johansson, P. D. Nation, and Franco Nori. “QuTiP 2: A Python framework for the dynamics of open quantum systems”. In: *Computer Physics Communications* 184.4 (2013), pp. 1234–1240. DOI: 10.1016/j.cpc.2012.11.019. arXiv: 1211.6518.

[18] Angelo Bassi, Kinjalk Lochan, Seema Satin, Tejinder P. Singh, and Hendrik Ulbricht. “Models of wave-function collapse, underlying theories, and experimental tests”. In: *Reviews of Modern Physics* 85.2 (2013), pp. 471–527. DOI: 10.1103/RevModPhys.85.471. arXiv: 1204.4325.

[19] Domenico D’Alessandro. *Introduction to Quantum Control and Dynamics*. Vol. 13. Chapman & Hall/CRC, 2007. DOI: 10.1201/9781584888833.

[20] Michael A Nielsen and Isaac L Chuang. *Quantum Computation and Quantum Information*. Cambridge Series on Information and the Natural Sciences. Cambridge University Press, 2000.

[21] Ángel Rivas and Susana F Huelga. *Open Quantum Systems: An Introduction (SpringerBriefs in Physics)*. Springer, 2011.

[22] Heinz-Peter Breuer and Francesco Petruccione. *The Theory of Open Quantum Systems*. Oxford University Press, 2007.

[23] David Elliott. *Bilinear Control Systems*. Springer, 2009. DOI: 10.1023/b101451.

[24] David Deutsch. “Quantum computation”. In: *Physics World* 5.6 (1992), p. 57.

[25] Richard P. Feynman. “Simulating physics with computers”. In: *International Journal of Theoretical Physics* 21.6-7 (1982), pp. 467–488. DOI: 10.1007/BF02650179. arXiv: 9508027 [quant-ph].

[26] Simon Singh. *Code Book, The: The Secret History of Codes and Code-breaking*. Fourth Estate, 1999.

[27] Charles H. Bennett and Gilles Brassard. “Quantum cryptography: Public key distribution and coin tossing”. In: *Theoretical Computer Science* 560.P1 (2014), pp. 7–11. DOI: 10.1016/j.tcs.2014.05.025.

[28] Artur K. Ekert. “Quantum cryptography based on Bellâs theorem”. In: *Physical Review Letters* 67.6 (1991), pp. 661–663. DOI: 10.1103/PhysRevLett.67.661. arXiv: 0911.4171v2.

[29] Richard J. Hughes, Jane E. Nordholt, Derek Derkacs, and Charles G. Peterson. “Practical free-space quantum key distribution over 10 km in daylight and at night”. In: *New Journal of Physics* 4 (2002). DOI: 10.1088/1367-2630/4/1/343. arXiv: 0206092 [quant-ph].

[30] Sheng Kai Liao et al. “Satellite-Relayed Intercontinental Quantum Network”. In: *Physical Review Letters* 120.3 (2018), p. 30501. DOI: 10.1103/PhysRevLett.120.030501. arXiv: 1801.04418.

[31] H. J. Kimble. “The quantum internet”. In: *Nature* 453.7198 (2008), pp. 1023–1030. DOI: 10.1038/nature07127. arXiv: 0806.4195.

[32] Eleni Diamanti, Hoi-Kwong Lo, Bing Qi, and Zhiliang Yuan. “Practical challenges in quantum key distribution”. In: *Nature Publishing Group* May (2016), pp. 1–12. DOI: 10.1038/npjqi.2016.25. arXiv: 1606.05853.

[33] Kwok Ho Wan, Oscar Dahlsten, Hlér Kristjánsson, Robert Gardner, and M. S. Kim. “Quantum generalisation of feedforward neural networks”. In: *npj Quantum Information* 3.1 (Dec. 2017), p. 36. DOI: 10.1038/s41534-017-0032-4.

[34] Yiqiu Ma, Haixing Miao, Belinda Heyun Pang, Matthew Evans, Chunnong Zhao, Jan Harms, Roman Schnabel, and Yanbei Chen. “Proposal for gravitational-wave detection beyond the standard quantum limit through EPR entanglement”. In: *Nature Physics* 13.8 (May 2017), pp. 776–780. DOI: 10.1038/nphys4118.

[35] Elena Boto et al. “Moving magnetoencephalography towards real-world applications with a wearable system”. In: *Nature* 555.7698 (2018), pp. 657–661. DOI: 10.1038/nature26147. arXiv: NIHMS150003.

[36] Michael A. Taylor and Warwick P. Bowen. “Quantum metrology and its application in biology”. In: *Physics Reports* 615 (Feb. 2016), pp. 1–59. DOI: 10.1016/J.PHYSREP.2015.12.002.

[37] M. Lapert, G. Ferrini, and D. Sugny. “Optimal control of quantum superpositions in a bosonic Josephson junction”. In: *Physical Review A - Atomic, Molecular, and Optical Physics* 85.2 (Feb. 2012), p. 023611. DOI: 10.1103/PhysRevA.85.023611. arXiv: 1106.2316.

[38] Sergei N Yurchenko, Wesley Bond, Maire N Gorman, Lorenzo Lodi, Laura K McKemmish, William Nunn, Rohan Shah, and Jonathan Tennyson. “ExoMol molecular line lists XXVI: spectra of SH and NS”. In: *Monthly Notices of the Royal Astronomical Society* 478.1 (Apr. 2018), pp. 270–282. DOI: 10.1093/mnras/sty939.

[39] I. M. Georgescu, S. Ashhab, and Franco Nori. “Quantum simulation”. In: *Reviews of Modern Physics* 86.1 (2014), pp. 153–185. DOI: 10.1103/RevModPhys.86.153. arXiv: 1308.6253.

[40] J. Ignacio Cirac and Peter Zoller. “Goals and opportunities in quantum simulation”. In: *Nature Physics* 8.4 (2012), pp. 264–266. DOI: 10.1038/nphys2275.

[41] T. H. Johnson, S. R. Clark, and D. Jaksch. “What is a quantum simulator?” In: *EPJ Quantum Technology* 1.10 (2014), pp. 1–12. DOI: 10.1186/epjqt10. arXiv: 1405.2831.

[42] B. P. Lanyon et al. “Universal digital quantum simulation with trapped ions”. In: *Science* 334.6052 (2011), pp. 57–61. DOI: 10.1126/science.1208001. arXiv: 1109.1512.

[43] R. Blatt and C. F. Roos. “Quantum simulations with trapped ions”. In: *Nature Physics* 8.4 (Apr. 2012), pp. 277–284. DOI: 10.1038/nphys2252.

[44] Joseph W. Britton, Brian C. Sawyer, Adam C. Keith, C.-C. Joseph Wang, James K. Freericks, Hermann Uys, Michael J. Biercuk, and John J. Bollinger. “Engineered two-dimensional Ising interactions in a trapped-ion quantum simulator with hundreds of spins”. In: *Nature* 484.7395 (Apr. 2012), pp. 489–492. DOI: 10.1038/nature10981.

[45] Justin G Bohnet, Brian C Sawyer, Joseph W Britton, Michael L Wall, Ana Maria Rey, Michael Foss-Feig, and John J Bollinger. “Quantum spin dynamics and entanglement generation with hundreds of trapped ions.” In: *Science* 352.6291 (June 2016), pp. 1297–301. DOI: 10.1126/science.aad9958.

[46] Henning Labuhn, Daniel Barredo, Sylvain Ravets, Sylvain de Léséleuc, Tommaso Macrì, Thierry Lahaye, and Antoine Browaeys. “Tunable two-dimensional arrays of single Rydberg atoms for realizing quantum Ising models”. In: *Nature* 534.7609 (June 2016), pp. 667–670. DOI: 10.1038/nature18274.

[47] Peter Williston Shor. “Algorithms for quantum computation: discrete logarithms and factoring”. In: *Proc. 35th Symp. on Foundations of Computer Science (Santa Fe, NM, 20-22 November) ed S Goldwasser*. Los Alamitos, CA: IEEE Computer Society Press, 1994, pp. 124–34.

[48] Peter W. Shor. “Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer”. In: *SIAM Journal on Computing* 26.5 (Oct. 1997), pp. 1484–1509. DOI: 10.1137/S0097539795293172.

[49] David P. DiVincenzo. “Two-bit gates are universal for quantum computation”. In: *Physical Review A* 51.2 (Feb. 1995), pp. 1015–1022. DOI: 10.1103/PhysRevA.51.1015. arXiv: 9407022 [cond-mat].

[50] L M Vandersypen, Matthias Steffen, Gregory Breyta, C S Yannoni, M H Sherwood, and I L Chuang. “Experimental realization of Shor’s quantum factoring algorithm using nuclear magnetic resonance.” In: *Nature* 414 (2001), pp. 883–7. DOI: 10.1038/414883a.

[51] Aram W. Harrow and Ashley Montanaro. “Quantum computational supremacy”. In: *Nature* 549.7671 (Sept. 2017), pp. 203–209. DOI: 10.1038/nature23458.

[52] C Neill et al. “A blueprint for demonstrating quantum supremacy with superconducting qubits.” In: *Science (New York, N.Y.)* 360.6385 (Apr. 2018), pp. 195–199. DOI: 10.1126/science.aao4309.

[53] A. P. Lund, Michael J. Bremner, and T. C. Ralph. “Quantum sampling problems, BosonSampling and quantum supremacy”. In: *npj Quantum Information* 3.1 (Dec. 2017), p. 15. DOI: 10.1038/s41534-017-0018-2.

[54] Sergio Boixo, Sergei V. Isakov, Vadim N. Smelyanskiy, Ryan Babbush, Nan Ding, Zhang Jiang, Michael J. Bremner, John M. Martinis, and Hartmut Neven. "Characterizing quantum supremacy in near-term devices". In: *Nature Physics* 14.6 (June 2018), pp. 1–6. DOI: 10.1038/s41567-018-0124-x. arXiv: 1608.00263.

[55] Emanuel Knill. "Q & A Quantum computing". In: *Nature Physics* 463.January (2010).

[56] M. Veldhorst et al. "A two-qubit logic gate in silicon". In: *Nature* 526.7573 (Oct. 2015), pp. 410–414. DOI: 10.1038/nature15263.

[57] S. Schaal, S. Barraud, J. J. L. Morton, and M. F. Gonzalez-Zalba. "Conditional Dispersive Readout of a CMOS Single-Electron Memory Cell". In: *Physical Review Applied* 9.5 (May 2018), p. 054016. DOI: 10.1103/PhysRevApplied.9.054016.

[58] Andrew A. Houck, Hakan E. Türeci, and Jens Koch. "On-chip quantum simulation with superconducting circuits". In: *Nature Physics* 8.4 (Apr. 2012), pp. 292–299. DOI: 10.1038/nphys2251. arXiv: 1203.5363.

[59] R. Barends et al. "Superconducting quantum circuits at the surface code threshold for fault tolerance". In: *Nature* 508.7497 (Apr. 2014), pp. 500–503. DOI: 10.1038/nature13171. arXiv: 1402.4848.

[60] Jeremy L. O'Brien. "Optical quantum computing". In: *Science* 318.5856 (Dec. 2007), pp. 1567–1570. DOI: 10.1126/science.1142892. arXiv: 0803.1554.

[61] Jan Benhelm, Gerhard Kirchmair, Christian F. Roos, and Rainer Blatt. "Towards fault-tolerant quantum computing with trapped ions". In: *Nature Physics* 4.6 (2008), pp. 463–466. DOI: 10.1038/nphys961. arXiv: 0803.2798.

[62] Gil Kalai. "The Quantum Computer Puzzle". In: *AMS Notices* 63.5 (2014), pp. 508–516.

[63] David Silver et al. "Mastering the game of Go with deep neural networks and tree search". In: *Nature* 529.7587 (Jan. 2016), pp. 484–489. DOI: 10.1038/nature16961.

[64] Alexey A Melnikov, Hendrik Poulsen Nautrup, Mario Krenn, Vedran Dunjko, Markus Tiersch, Anton Zeilinger, and Hans J Briegel. "Active learning machine learns to create new quantum experiments." In: *Proceedings of the National Academy of Sciences of the United States of America* 115.6 (Feb. 2018), pp. 1221–1226. DOI: 10.1073/pnas.1714936115.

[65] P. Baireuther, T. E. O'Brien, B. Tarasinski, and C. W. J. Beenakker. "Machine-learning-assisted correction of correlated qubit errors in a topological code". In: *Quantum* 2 (May 2017), p. 48. DOI: 10.22331/q-2018-01-29-48. arXiv: 1705.07855.

[66] Pantita Palittapongarnpim, Peter Wittek, Ehsan Zahedinejad, Shakib Vedaie, and Barry C. Sanders. "Learning in quantum control: High-dimensional global optimization for noisy quantum dynamics". In: *Neurocomputing* 268 (Dec. 2017), pp. 116–126. DOI: 10.1016/J.NEUCOM.2016.12.087. arXiv: 1607.03428.

[67] Marin Bukov, Alexandre G. R. Day, Dries Sels, Phillip Weinberg, Anatoli Polkovnikov, and Pankaj Mehta. "Reinforcement Learning in Different Phases of Quantum Control". In: (May 2017), pp. 1–20. arXiv: 1705.00565.

[68] Murphy Yuezhen Niu, Sergio Boixo, Vadim Smelyanskiy, and Hartmut Neven. "Universal Quantum Control through Deep Reinforcement Learning". In: (2018), pp. 1–21. arXiv: 1803.01857.

[69] P. B. Wigley et al. “Fast machine-learning online optimization of ultra-cold-atom experiments”. In: *Scientific Reports* 6.1 (Sept. 2016), p. 25890. DOI: 10.1038/srep25890.

[70] Jacob Biamonte, Peter Wittek, Nicola Pancotti, Patrick Rebentrost, Nathan Wiebe, and Seth Lloyd. “Quantum machine learning”. In: *Nature* 549.7671 (2017), pp. 195–202. DOI: 10.1038/nature23474. arXiv: 1611.09347.

[71] Aram W. Harrow, Avinatan Hassidim, and Seth Lloyd. “Quantum Algorithm for Linear Systems of Equations”. In: *Physical Review Letters* 103.15 (Oct. 2009), p. 150502. DOI: 10.1103/PhysRevLett.103.150502.

[72] Patrick Rebentrost, Maria Schuld, Leonard Wossnig, Francesco Petruccione, and Seth Lloyd. “Quantum gradient descent and Newton’s method for constrained polynomial optimization”. In: (Dec. 2016). arXiv: 1612.01789.

[73] D. Sugny, C. Kontz, and H. R. Jauslin. “Time-optimal control of a two-level dissipative quantum system”. In: *Physical Review A - Atomic, Molecular, and Optical Physics* 76.2 (Aug. 2007), p. 023419. DOI: 10.1103/PhysRevA.76.023419. arXiv: 0708.3794.

[74] M. Lapert, E. Assémat, S. J. Glaser, and D. Sugny. “Understanding the global structure of two-level quantum systems with relaxation: Vector fields organized through the magic plane and the steady-state ellipsoid”. In: *Physical Review A - Atomic, Molecular, and Optical Physics* 88.3 (Sept. 2013), p. 033407. DOI: 10.1103/PhysRevA.88.033407.

[75] Alexei Gilchrist, Nathan K. Langford, and Michael A. Nielsen. “Distance measures to compare real and ideal quantum processes”. In: *Physical Review A* 71.6 (June 2005), p. 062310. DOI: 10.1103/PhysRevA.71.062310.

[76] Frederik F. Floether, Pierre De Fouquieres, and Sophie G. Schirmer. “Robust quantum gates for open systems via optimal control: Markovian versus non-Markovian dynamics”. In: *New Journal of Physics* 14.7 (2012), p. 073023. DOI: 10.1088/1367-2630/14/7/073023. arXiv: arXiv:1107.4358v2.

[77] Nelson Leung, Mohamed Abdelhafez, Jens Koch, and David I. Schuster. “Speedup for quantum optimal control from automatic differentiation based on graphics processing units”. In: *Physical Review A* 95.4 (2017), pp. 1–14. DOI: 10.1103/PhysRevA.95.042318. arXiv: 1612.04929.

[78] S Machnes, U Sander, S J Glaser, P De Fouquieres, A Gruslys, S Schirmer, and T Schulte-Herbrueggen. “Comparing, Optimising and Benchmarking Quantum Control Algorithms in a Unifying Programming Framework”. In: *Physical Review A* 84.2 (2010), pp. 1–23. arXiv: 1011.4874.

[79] D. L. Goodwin and Ilya Kuprov. “Auxiliary matrix formalism for interaction representation transformations, optimal control, and spin relaxation theories”. In: *Journal of Chemical Physics* 143.8 (Aug. 2015), pp. 1–7. DOI: 10.1063/1.4928978. arXiv: 1506.00628.

[80] Jorge Nocedal. “Updating quasi-Newton matrices with limited storage”. In: *Mathematics of Computation* 35.151 (Sept. 1980), pp. 773–773. DOI: 10.1090/S0025-5718-1980-0572855-7.

[81] Richard H. Byrd, Peihuang Lu, Jorge Nocedal, and Ciyou Zhu. “A Limited Memory Algorithm for Bound Constrained Optimization”. In: *SIAM Journal on Scientific Computing* 16 (1995), pp. 1190–1208. DOI: 10.1137/0916069.

[82] Shai Machnes, Elie Assémat, David Tannor, and Frank K. Wilhelm. “Tunable, Flexible, and Efficient Optimization of Control Pulses for Practical Qubits”. In: *Physical Review Letters* 120.15 (July 2018), p. 150401. DOI: 10.1103/PhysRevLett.120.150401. arXiv: arXiv:1507.04261v2.

[83] Navin Khaneja, Timo Reiss, Cindie Kehlet, Thomas Schulte-Herbruggen, Steffen J. Glaser, Thomas Schulte-Herbrüggen, and Steffen J. Glaser. “Optimal control of coupled spin dynamics: Design of NMR pulse sequences by gradient ascent algorithms”. In: *Journal of Magnetic Resonance* 172.2 (Mar. 2005), pp. 296–305. DOI: 10.1016/j.jmr.2004.11.004.

[84] Nicholas J Higham. *Functions of Matrices: Theory and Computation*. Society for Industrial & Applied Mathematics,U.S., 2008.

[85] H. J. Hogben, M. Krzystyniak, G. T.P. Charnock, P. J. Hore, and Ilya Kuprov. “Spinach - A software library for simulation of spin dynamics in large spin systems”. In: *Journal of Magnetic Resonance* 208.2 (2011), pp. 179–194. DOI: 10.1016/j.jmr.2010.11.008. arXiv: NIHMS150003.

[86] Travis E Oliphant. *A guide to NumPy*. USA: Trelgol Publishing, 2006.

[87] Eric Jones, Travis Oliphant, Pearu Peterson, et al. *SciPy: Open source scientific tools for Python*. URL: http://www.scipy.org/.

[88] Vadim Krotov. *Global methods in optimal control theory*. 1st ed. Monographs and textbooks in pure and applied mathematics 195. M. Dekker, 1996.

[89] Michael H. Goerz, Daniel Basilewitsch, Fernando Gago-Encinas, Matthias G. Krauss, Karl P. Horn, Daniel M. Reich, and Christiane P. Koch. “Krotov: A Python implementation of Krotov’s method for quantum optimal control”. In: (Feb. 2019). arXiv: 1902.11284.

[90] P Doria, T Calarco, and S Montangero. “Optimal Control Technique for Many-Body Quantum Dynamics”. In: *Physical Review Letters* 106.19 (2011), pp. 1–4. DOI: 10.1103/PhysRevLett.106.190501.

[91] Tommaso Caneva, Tommaso Calarco, and Simone Montangero. “Chopped random-basis quantum optimization”. In: *Physical Review A - Atomic, Molecular, and Optical Physics* 84.2 (2011). DOI: 10.1103/PhysRevA.84.022326. arXiv: 1103.0855.

[92] J. A. Nelder and R. Mead. “A Simplex Method for Function Minimization”. In: *The Computer Journal* 7.4 (Jan. 1965), pp. 308–313. DOI: 10.1093/comjnl/7.4.308.

[93] 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”. In: *Physical Review A* 92.6 (2015), pp. 1–7. DOI: 10.1103/PhysRevA.92.062343. arXiv: 1506.04601.

[94] H A Rabitz. “Quantum Optimally Controlled Transition Landscapes”. In: *Science* 303.5666 (Mar. 2004), pp. 1998–2001. DOI: 10.1126/science.1093649.

[95] H. Rabitz, T. S. Ho, M. Hsieh, R. Kosut, and M. Demiralp. “Topology of optimally controlled quantum mechanical transition probability landscapes”. In: *Physical Review A - Atomic, Molecular, and Optical Physics* 74.1 (July 2006), p. 012721. DOI: 10.1103/PhysRevA.74.012721.

[96] Raj Chakrabarti and Herschel Rabitz. “Quantum control landscapes”. In: *International Reviews in Physical Chemistry* 26.4 (Oct. 2007), pp. 671–735. DOI: 10.1080/01442350701633300. arXiv: 0710.0684.

[97] Benjamin Russell, Herschel Rabitz, and Re Bing Wu. “Control landscapes are almost always trap free: A geometric assessment”. In: *Journal of Physics A: Mathematical and Theoretical* 50.20 (May 2017), p. 205302. DOI: 10.1088/1751-8121/aa6b77. arXiv: 1608.06198.

[98] Alexander N. Pechen and David J. Tannor. “Are there traps in quantum control landscapes?” In: *Physical Review Letters* 106.12 (Mar. 2011), p. 120402. DOI: 10.1103/PhysRevLett.106.120402. arXiv: 1508.05434.

[99] Christian Arenz, Giulia Gualdi, and Daniel Burgarth. “Control of open quantum systems: Case study of the central spin model”. In: *New Journal of Physics* 16 (2014), pp. 1–19. DOI: 10.1088/1367-2630/16/6/065023. arXiv: arXiv:1312.0160v1.

[100] G van Rossum et al. *Python programming language*. URL: http://www.python.org.

[101] Wes McKinney. “Data Structures for Statistical Computing in Python”. In: *Proceedings of the 9th Python in Science Conference* 1697900.Scipy (2010), pp. 51–56.

[102] *SymPy: A Python library for symbolic mathematics*. URL: https://www.sympy.org.

[103] John D. Hunter. “Matplotlib: A 2D graphics environment”. In: *Computing in Science and Engineering* 9.3 (2007), pp. 99–104. DOI: 10.1109/MCSE.2007.55. arXiv: 0402594v3 [arXiv:cond-mat].

[104] *Wolfram Mathematica*. URL: http://www.wolfram.com/mathematica/.

[105] *The Jupyter Notebook: Sharing documents that contain live code, equations, visualizations and narrative text*. URL: http://jupyter.org/.

[106] *Linux Mint: Elegant, easy to use, up to date and comfortable GNU/Linux desktop distribution*. URL: https://www.linuxmint.com/.

[107] *Anaconda: Python data science platform*. URL: https://www.anaconda.com/.

[108] *Intel Math Kernal Library (MKL): Math library for Intel based systems*. URL: https://software.intel.com/en-us/mkl.

[109] Sze M Tan et al. “A computational toolbox for quantum and atomic optics”. In: *Journal of Optics B: Quantum and Semiclassical Optics* 1.4 (1999), pp. 424–432. DOI: 10.1088/1464-4266/1/4/312.

[110] *Mathworks Matlab*. URL: https://www.mathworks.com/products/matlab.html.

[111] *Github development platform*. URL: https://github.com/.

[112] *QuTiP website*. URL: http://qutip.org/.

[113] *QuTiP Github repository*. URL: https://github.com/qutip.

[114] *QuTiP documentation*. URL: http://qutip.org/documentation.html.

[115] *OpenMP parallel processing library*. URL: https://www.openmp.org/.

[116] *Supercomputing Wales: The national supercomputing research facility for Wales*. URL: https://www.supercomputing.wales/.

[117] Edwin Barnes, Christian Arenz, Alexander Pitchford, and Sophia E. Economou. “Fast microwave-driven three-qubit gates for cavity-coupled superconducting qubits”. In: *Physical Review B* 96.2 (July 2017), p. 024504. DOI: 10.1103/PhysRevB.96.024504. arXiv: 1612.09384.

[118] Jukka Kiukas, Kazuya Yuasa, and Daniel Burgarth. “Remote parameter estimation in a quantum spin chain enhanced by local control”. In: *Physical Review A* 95.5 (Jan. 2017), pp. 1–6. DOI: 10.1103/PhysRevA.95.052132. arXiv: 1701.07399.

[119] Ciyou Zhu, Richard H. Byrd, Peihuang Lu, and Jorge Nocedal. “Algorithm 778: L-BFGS-B: Fortran subroutines for large-scale bound-constrained optimization”. In: *ACM Transactions on Mathematical Software* 23.4 (1997), pp. 550–560. DOI: 10.1145/279232.279236.

[120] *QuTiP tutorials*. URL: http://qutip.org/tutorials.html.

[121] *QuTiP control examples*. URL: https://github.com/ajgpitch/qtrl-tutorial.

[122] Alexander Pitchford and Benjamin Dive. *In Situ Upgrade code repository*. URL: https://github.com/ajgpitch/quantum-in%7B%5C_%7Dsitu%7B%5C_%7Dopt.

[123] *Slurm workload manager: cluster management and job scheduling system*. URL: https://www.schedmd.com/.

[124] O Tange. “GNU Parallel - The Command-Line Power Tool”. In: *;login: The USENIX Magazine* 36.1 (Feb. 2011), pp. 42–47. DOI: http://dx.doi.org/10.5281/zenodo.16303.

[125] Neill Lambert. *Excitation-number-restricted states: Jaynes-Cummings chain*. URL: https://github.com/qutip/qutip-notebooks/blob/master/examples/excitation-number-restricted-states-jc-chain.ipynb.

[126] Stefan Behnel, Robert Bradshaw, Craig Citro, Lisandro Dalcin, Dag Sverre Seljebotn, and Kurt Smith. “Cython: The best of both worlds”. In: *Computing in Science and Engineering* 13.2 (2011), pp. 31–39. DOI: 10.1109/MCSE.2010.118.

[127] *Github repositiory ajgpitch/qutip: Alexander Pitchford’s fork of qutip*. URL: https://github.com/ajgpitch/qutip.

[128] Uther Shackerley-Bennett, Alexander Pitchford, Marco G Genoni, Alessio Serafini, and Daniel K Burgarth. “The reachable set of single-mode quadratic Hamiltonians”. In: *Journal of Physics A: Mathematical and Theoretical* 50.15 (Apr. 2017), p. 155203. DOI: 10.1088/1751-8121/aa6243. arXiv: 1605.09688.

[129] Stephen D. Bartlett, Barry C. Sanders, Samuel L. Braunstein, and Kae Nemoto. “Efficient Classical Simulation of Continuous Variable Quantum Information Processes”. In: *Physical Review Letters* 88.9 (2002), p. 4. DOI: 10.1103/PhysRevLett.88.097904. arXiv: 0109047 [quant-ph].

[130] Alessandro Ferraro, Stefano Olivares, and Matteo G A Paris. “Gaussian states in continuous variable quantum information”. In: *Evolution* (2005), p. 100.

[131] S. Olivares. “Quantum optics in the phase space”. In: *The European Physical Journal Special Topics* 203.1 (Apr. 2012), pp. 3–24. DOI: 10.1140/epjst/e2012-01532-4.

[132] D. Leibfried, R. Blatt, C. Monroe, and D. Wineland. “Quantum dynamics of single trapped ions”. In: *Reviews of Modern Physics* 75.1 (Mar. 2003), p. 281. DOI: 10.1103/RevModPhys.75.281.

[133] Markus Aspelmeyer, Tobias J. Kippenberg, and Florian Marquardt. *Cavity optomechanics: Nano- and micromechanical resonators interacting with light*. Springer Berlin Heidelberg, 2014, pp. 1–357. DOI: 10.1007/978-3-642-55312-7. arXiv: 1303.0733.

[134] V V Dodonov and V I Man’ko. *Theory of Nonclassical States of Light*. Ed. by V I Man’ko V.V. Dodonov. Springer Berlin Heidelberg, 2003. DOI: manko2002.

[135] Christian Weedbrook, Stefano Pirandola, Raúl García-Patrón, Nicolas J. Cerf, Timothy C. Ralph, Jeffrey H. Shapiro, and Seth Lloyd. “Gaussian quantum information”. In: *Reviews of Modern Physics* 84.2 (May 2012), pp. 621–669. DOI: 10.1103/RevModPhys.84.621.

[136] Maurice A de Gosson. *Symplectic Methods in Harmonic Analysis and in Mathematical Physics*. Birkhäuser, 2011.

[137] Helmut Hofer and Eduard Zehnder. *Symplectic Invariants and Hamiltonian Dynamics*. Birkhäuser, 2011.

[138] Zbigniew W. Gortel and Bukasz A. Turski. “Classical dynamics for a class of SU(1,1) Hamiltonians”. In: *Physical Review A* 43.7 (1991), pp. 3221–3226. DOI: 10.1103/PhysRevA.43.3221.

[139] S. K. Bose. “Dynamical algebra of spin waves in localised-spin models”. In: *Journal of Physics A: General Physics* 18.6 (1985), pp. 903–922. DOI: 10.1088/0305-4470/18/6/014.

[140] Arvind, B Dutta, N Mukunda, and R Simon. “The real symplectic groups in quantum mechanics and optics”. In: *Pramana* 45.6 (Dec. 1995), pp. 471–497. DOI: 10.1007/BF02848172. arXiv: 9509002 [quant-ph].

[141] Samuel L. Braunstein. “Squeezing as an irreducible resource”. In: *Physical Review A - Atomic, Molecular, and Optical Physics* 71.5 (May 2005), p. 055801. DOI: 10.1103/PhysRevA.71.055801. arXiv: 9904002 [quant-ph].

[142] Rachida El Assoudi-Baikari. “Semigroups of Simple Lie Groups and Controllability”. In: *Journal of Dynamical and Control Systems* 20.1 (Jan. 2014), pp. 91–104. DOI: 10.1007/s10883-013-9201-8.

[143] Michael Murphy, Simone Montangero, Vittorio Giovannetti, and Tommaso Calarco. “Communication at the quantum speed limit along a spin chain”. In: *Physical Review A - Atomic, Molecular, and Optical Physics* 82.2 (2010). DOI: 10.1103/PhysRevA.82.022318. arXiv: 1004.3445.

[144] H Pino, J Prat-Camps, K Sinha, B Prasanna Venkatesh, and O Romero-Isart. "On-chip quantum interference of a superconducting microsphere". In: *Quantum Science and Technology* 3.2 (Apr. 2018), p. 025001. DOI: 10.1088/2058-9565/aa9d15.

[145] Benjamin Dive, Alexander Pitchford, Florian Mintert, and Daniel Burgarth. "In situ upgrade of quantum simulators to universal computers". In: *Quantum* 2 (Aug. 2018), p. 80. DOI: 10.22331/q-2018-08-08-80. arXiv: 1701.01723.

[146] Michael Johanning, Andrés F. Varón, and Christof Wunderlich. "Quantum simulations with cold trapped ions". In: *Journal of Physics B: Atomic, Molecular and Optical Physics* 42.15 (Aug. 2009), p. 154009. DOI: 10.1088/0953-4075/42/15/154009. arXiv: 0905.0118.

[147] Immanuel Bloch, Jean Dalibard, and Sylvain Nascimbène. "Quantum simulations with ultracold quantum gases". In: *Nature Physics* 8.4 (Apr. 2012), pp. 267–276. DOI: 10.1038/nphys2259.

[148] Xin hua Peng, Dieter Suter, Xin hua Peng, and Dieter Suter. "Spin qubits for quantum simulations". In: *Frontiers of Physics in China* 5.1 (Mar. 2010), pp. 1–25. DOI: 10.1007/s11467-009-0067-x.

[149] Isabela A. Silva et al. "Observation of time-invariant coherence in a nuclear magnetic resonance quantum simulator". In: *Physical Review Letters* 117.16 (Oct. 2016), p. 160402. DOI: 10.1103/PhysRevLett.117.160402. arXiv: 1511.01971.

[150] Jianming Cai, Alex Retzker, Fedor Jelezko, and Martin B. Plenio. "A large-scale quantum simulator on a diamond surface at room temperature". In: *Nature Physics* 9.3 (Mar. 2013), pp. 168–173. DOI: 10.1038/nphys2519.

[151] P. J.J. O'Malley et al. "Scalable quantum simulation of molecular energies". In: *Physical Review X* 6.3 (July 2016), p. 031007. DOI: 10.1103/PhysRevX.6.031007. arXiv: 1512.06860.

[152] Peter L McMahon et al. "A fully programmable 100-spin coherent Ising machine with all-to-all connections." In: *Science* 354.6312 (Nov. 2016), pp. 614–617. DOI: 10.1126/science.aah5178.

[153] Richard S. Judson and Herschel Rabitz. "Teaching lasers to control molecules". In: *Physical Review Letters* 68.10 (1992), pp. 1500–1503. DOI: 10.1103/PhysRevLett.68.1500.

[154] M Q Phan and H Rabitz. "Learning control of quantum-mechanical systems by laboratory identification of effective input-output maps". In: *Chemical Physics* 217.2-3 (May 1997), p. 389. DOI: 10.1016/S0301-0104(97)00068-2.

[155] Minh Q. Phan and Herschel Rabitz. "A self-guided algorithm for learning control of quantum-mechanical systems". In: *Journal of Chemical Physics* 110.1 (1999), pp. 34–41. DOI: 10.1063/1.478081.

[156] T.C. Weinacht, R. Bartels, S. Backus, P.H. Bucksbaum, B. Pearson, J.M. Geremia, H. Rabitz, H.C. Kapteyn, and M.M. Murnane. "Coherent learning control of vibrational motion in room temperature molecular gases". In: *Chemical Physics Letters* 344.3-4 (Aug. 2001), pp. 333–338. DOI: 10.1016/S0009-2614(01)00788-6.

[157] Wusheng Zhu and Herschel Rabitz. “Closed loop learning control to suppress the effects of quantum decoherence”. In: *The Journal of Chemical Physics* 118.15 (Apr. 2003), pp. 6751–6757. DOI: 10.1063/1.1559484.

[158] David Cardoza, Carlos Trallero-Herrero, Florian Langhojer, Herschel Rabitz, and Thomas Weinacht. “Transformations to diagonal bases in closed-loop quantum learning control experiments”. In: *The Journal of Chemical Physics* 122.12 (Mar. 2005), p. 124306. DOI: 10.1063/1.1867334.

[159] Chunlin Chen, Daoyi Dong, Ruixing Long, Ian R. Petersen, and Herschel A. Rabitz. “Sampling-based learning control of inhomogeneous quantum ensembles”. In: *Physical Review A - Atomic, Molecular, and Optical Physics* 89.2 (Feb. 2014), p. 023402. DOI: 10.1103/PhysRevA.89.023402. arXiv: 1308.1454.

[160] D. J. Egger and F. K. Wilhelm. “Adaptive Hybrid Optimal Quantum Control for Imprecisely Characterized Systems”. In: *Physical Review Letters* 112.24 (June 2014), p. 240503. DOI: 10.1103/PhysRevLett.112.240503.

[161] D. J. Egger and F. K. Wilhelm. “Optimized controlled-Z gates for two superconducting qubits coupled through a resonator”. In: *Superconductor Science and Technology* 27.1 (Jan. 2014), p. 014001. DOI: 10.1088/0953-2048/27/1/014001. arXiv: 1306.6894.

[162] Easwar Magesan, J. M. Gambetta, and Joseph Emerson. “Scalable and robust randomized benchmarking of quantum processes”. In: *Physical Review Letters* 106.18 (2011), pp. 8–11. DOI: 10.1103/PhysRevLett.106.180504. arXiv: 1009.3639.

[163] A. D. Córcoles, Jay M. Gambetta, Jerry M. Chow, John A. Smolin, Matthew Ware, Joel Strand, B. L.T. Plourde, and M. Steffen. “Process verification of two-qubit quantum gates by randomized benchmarking”. In: *Physical Review A - Atomic, Molecular, and Optical Physics* 87.3 (2013), pp. 1–4. DOI: 10.1103/PhysRevA.87.030301. arXiv: 1210.7011.

[164] Jun Li, Xiaodong Yang, Xinhua Peng, and Chang-Pu Sun. “Hybrid Quantum-Classical Approach to Quantum Optimal Control”. In: *Physical Review Letters* 118.15 (Apr. 2017), p. 150503. DOI: 10.1103/PhysRevLett.118.150503.

[165] Seth Lloyd. “Almost Any Quantum Logic Gate is Universal”. In: *Physical Review Letters* 75.2 (July 1995), pp. 346–349. DOI: 10.1103/PhysRevLett.75.346.

[166] Jennifer L. Dodd, Michael A. Nielsen, Michael J. Bremner, and Robert T. Thew. “Universal quantum computation and simulation using any entangling Hamiltonian and local unitaries”. In: *Physical Review A* 65.4 (Apr. 2002), p. 040301. DOI: 10.1103/PhysRevA.65.040301.

[167] Daniel Burgarth, Sougato Bose, Christoph Bruder, and Vittorio Giovannetti. “Local controllability of quantum networks”. In: *Physical Review A* 79.6 (June 2009), p. 060305. DOI: 10.1103/PhysRevA.79.060305.

[168] Marcus P. da Silva, Olivier Landon-Cardinal, and David Poulin. “Practical Characterization of Quantum Devices without Tomography”. In: *Physical Review Letters* 107.21 (Nov. 2011), p. 210404. DOI: 10.1103/PhysRevLett.107.210404.

[169] Marcus Cramer, Martin B. Plenio, Steven T. Flammia, Rolando Somma, David Gross, Stephen D. Bartlett, Olivier Landon-Cardinal, David Poulin, and Yi-Kai Liu. "Efficient quantum state tomography". In: *Nature Communications* 1.9 (Dec. 2010), p. 149. DOI: 10.1038/ncomms1147.

[170] Christopher Ferrie and Osama Moussa. "Robust and efficient in situ quantum control". In: *Physical Review A - Atomic, Molecular, and Optical Physics* 91.5 (May 2015), p. 052306. DOI: 10.1103/PhysRevA.91.052306. arXiv: 1409.3172.

[171] Christian Arenz. "Control of open quantum systems". PhD thesis. Aberystwyth University, 2016. URL: http://hdl.handle.net/2160/43312.

[172] Benjamin Dive. "Controlling Open Quantum Systems". PhD thesis. Imperial College London, 2017. URL: http://hdl.handle.net/10044/1/56626.

[173] Yoshitaka Tanimura and Ryogo Kubo. "Time Evolution of a Quantum System in Contact with a Nearly Gaussian-Markoffian Noise Bath". In: *Journal of the Physical Society of Japan* 58.1 (1989), pp. 101–114. DOI: 10.1143/JPSJ.58.101.

[174] Yoshitaka Tanimura. "Nonperturbative expansion method for a quantum system coupled to a harmonic-oscillator bath". In: *Physical Review A* 41.12 (1990), pp. 6676–6687. DOI: 10.1103/PhysRevA.41.6676.

[175] Akihito Ishizaki and Yoshitaka Tanimura. "Quantum dynamics of system strongly coupled to low-temperature colored noise bath: Reduced hierarchy equations approach". In: *Journal of the Physical Society of Japan* 74.12 (2005), pp. 3131–3134. DOI: 10.1143/JPSJ.74.3131.

[176] Akihito Ishizaki and Graham R. Fleming. "Unified treatment of quantum coherent and incoherent hopping dynamics in electronic energy transfer: Reduced hierarchy equation approach". In: *Journal of Chemical Physics* 130.23 (2009), pp. 1–10. DOI: 10.1063/1.3155372.

[177] Gregory S. Engel, Tessa R. Calhoun, Elizabeth L. Read, Tae Kyu Ahn, Tomáš Mančal, Yuan Chung Cheng, Robert E. Blankenship, and Graham R. Fleming. "Evidence for wavelike energy transfer through quantum coherence in photosynthetic systems". In: *Nature* 446.7137 (2007), pp. 782–786. DOI: 10.1038/nature05678. arXiv: 1001.5108.

[178] Tony Zhu, Jordan M. Snaider, Long Yuan, and Libai Huang. "Ultrafast Dynamic Microscopy of Carrier and Exciton Transport". In: *Annual Review of Physical Chemistry* 70.1 (Apr. 2019), annurev–physchem–042018–052605. DOI: 10.1146/annurev-physchem-042018-052605.

[179] Rui Xue Xu and Yijing Yan. "Dynamics of quantum dissipation systems interacting with bosonic canonical bath: Hierarchical equations of motion approach". In: *Physical Review E - Statistical, Nonlinear, and Soft Matter Physics* 75.3 (2007), pp. 1–11. DOI: 10.1103/PhysRevE.75.031107. arXiv: 0612649 [cond-mat].

[180] Etienne Mangaud, Raijumon Puthumpally-Joseph, Dominique Sugny, Christoph Meier, Osman Atabek, and Michèle Desouter-Lecomte. "Non-Markovianity in the optimal control of an open quantum system described by hierarchical equations of motion". In: (Nov. 2017). arXiv: 1711.05162.

[181] Jake Iles-Smith, Neill Lambert, and Ahsan Nazir. “Environmental dynamics, correlations, and the emergence of noncanonical equilibrium states in open quantum systems”. In: *Physical Review A - Atomic, Molecular, and Optical Physics* 90.3 (2014), pp. 1–9. DOI: 10.1103/PhysRevA.90.032114. arXiv: 1311.0016.

[182] Jake Iles-Smith, Arend G. Dijkstra, Neill Lambert, and Ahsan Nazir. “Energy transfer in structured and unstructured environments: Master equations beyond the Born-Markov approximations”. In: *Journal of Chemical Physics* 144.4 (2016), pp. 62–64. DOI: 10.1063/1.4940218. arXiv: 1511.05181.

[183] Johan Strümpfer and Klaus Schulten. “Open quantum dynamics calculations with the hierarchy equations of motion on parallel computers”. In: *Journal of Chemical Theory and Computation* 8.8 (2012), pp. 2808–2816. DOI: 10.1021/ct3003833. arXiv: NIHMS150003.

[184] Jeffrey J Hunter. *Mathematical Techniques of Applied Probability. Volume 1: Discrete Time Models; Basic Theory (Operations Research and Industrial Engineering)*. Elsevier Inc, Academic Press, 1983.

[185] Bassano Vacchini, Andrea Smirne, Elsi Mari Laine, Jyrki Piilo, and Heinz Peter Breuer. “Markovianity and non-Markovianity in quantum and classical systems”. In: *New Journal of Physics* 13.9 (Sept. 2011), p. 093004. DOI: 10.1088/1367-2630/13/9/093004. arXiv: 1106.0138.

[186] Heinz Peter Breuer, Elsi Mari Laine, and Jyrki Piilo. “Measure for the Degree of Non-Markovian Behavior of Quantum Processes in Open Systems”. In: *Physical Review Letters* 103.21 (2009), pp. 1–4. DOI: 10.1103/PhysRevLett.103.210401. arXiv: 0908.0238.

[187] Francesco Buscemi and Nilanjana Datta. “Equivalence between divisibility and monotonic decrease of information in classical and quantum stochastic processes”. In: *Physical Review A* 93.1 (2016), pp. 1–7. DOI: 10.1103/PhysRevA.93.012101. arXiv: 1408.7062.

[188] Ángel Rivas, Susana F. Huelga, and Martin B. Plenio. “Quantum non-Markovianity: Characterization, quantification and detection”. In: *Reports on Progress in Physics* 77.9 (2014). DOI: 10.1088/0034-4885/77/9/094001. arXiv: 1405.0303.

[189] Christian Arenz, Robin Hillier, Martin Fraas, and Daniel Burgarth. “Distinguishing decoherence from alternative quantum theories by dynamical decoupling”. In: *Physical Review A - Atomic, Molecular, and Optical Physics* 92.2 (Aug. 2015), p. 022102. DOI: 10.1103/PhysRevA.92.022102. arXiv: 1405.7644.

[190] A. Kossakowski. “On quantum statistical mechanics of non-Hamiltonian systems”. In: *Reports on Mathematical Physics* 3.4 (Dec. 1972), pp. 247–274. DOI: 10.1016/0034-4877(72)90010-9.

[191] Vittorio Gorini, Andrzej Kossakowski, and E. C. G. Sudarshan. “Completely positive dynamical semigroups of N-level systems”. In: *Journal of Mathematical Physics* 17.5 (Aug. 1976), p. 821. DOI: 10.1063/1.522979.

[192] G. Lindblad. “On the generators of quantum dynamical semigroups”. In: *Communications in Mathematical Physics* 48.2 (June 1976), pp. 119–130. DOI: 10.1007/BF01608499.

[193] R Alicki and K Lendi. *Quantum dynamical semigroups and applications*. Berlin New York: Springer-Verlag, 2007.

[194] Carole Addis, Elsi Mari Laine, Clemens Gneiting, and Sabrina Maniscalco. “Problem of coherent control in non-Markovian open quantum systems”. In: *Physical Review A* 94.5 (2016), pp. 1–7. DOI: 10.1103/PhysRevA.94.052117. arXiv: 1604.07998.

[195] R.P Feynman and F.L Vernon. “The theory of a general quantum system interacting with a linear dissipative system”. In: *Annals of Physics* 24 (Oct. 1963), pp. 118–173. DOI: 10.1016/0003-4916(63)90068-X.

[196] Rui Xue Xu, Ping Cui, Xin Qi Li, Yan Mo, and Yijing Yan. “Exact quantum master equation via the calculus on path integrals”. In: *Journal of Chemical Physics* 122.4 (2005). DOI: 10.1063/1.1850899. arXiv: 0409007 [quant-ph].

[197] Arend G. Dijkstra and Yoshitaka Tanimura. “Non-Markovian entanglement dynamics in the presence of system-bath coherence”. In: *Physical Review Letters* 104.25 (2010), pp. 1–4. DOI: 10.1103/PhysRevLett.104.250401. arXiv: 1004.1450.

[198] Félix Beaudoin, Jay M. Gambetta, and A. Blais. “Dissipation and ultrastrong coupling in circuit QED”. In: *Physical Review A - Atomic, Molecular, and Optical Physics* 84.4 (2011). DOI: 10.1103/PhysRevA.84.043832. arXiv: 1107.3990.

[199] Daniel A Lidar and Todd A Brun. *Quantum Error Correction*. Cambridge University Press, 2013.

[200] Robin Hillier, Christian Arenz, and Daniel Burgarth. “A continuous-time diffusion limit theorem for dynamical decoupling and intrinsic decoherence”. In: *Journal of Physics A: Mathematical and Theoretical* 48.15 (2015). DOI: 10.1088/1751-8113/48/15/155301. arXiv: 1405.7666.

[201] Lorenza Viola and Seth Lloyd. “Dynamical suppression of decoherence in two-state quantum systems”. In: *Physical Review A* 58.4 (Oct. 1998), pp. 2733–2744. DOI: 10.1103/PhysRevA.58.2733.

[202] Gregory Quiroz and Daniel A. Lidar. “Optimized dynamical decoupling via genetic algorithms”. In: *Physical Review A* 88.5 (Nov. 2013), p. 052306. DOI: 10.1103/PhysRevA.88.052306.

[203] Lorenza Viola and Emanuel Knill. “Random Decoupling Schemes for Quantum Dynamical Control and Error Suppression”. In: *Physical Review Letters* 94.6 (Feb. 2005), p. 060502. DOI: 10.1103/PhysRevLett.94.060502.

[204] Zhe Sun, Longwen Zhou, Gaoyang Xiao, Dario Poletti, and Jiangbin Gong. “Finite-time Landau-Zener processes and counterdiabatic driving in open systems: Beyond Born, Markov, and rotating-wave approximations”. In: *Physical Review A* 93.1 (2016), pp. 1–10. DOI: 10.1103/PhysRevA.93.012121. arXiv: 1511.07695.

[205] Christoph Kreisbeck, Tobias Kramer, Mirta Rodríguez, and Birgit Hein. “High-performance solution of hierarchical equations of motion for studying energy transfer in light-harvesting complexes”. In: *Journal of Chemical Theory and Computation* 7.7 (2011), pp. 2166–2174. DOI: 10.1021/ct200126d. arXiv: 1012.4382.

[206] Masashi Tsuchimoto and Yoshitaka Tanimura. “Spins Dynamics in a Dissipative Environment: Hierarchal Equations of Motion Approach Using a Graphics Processing Unit (GPU)”. In: *Journal of Chemical Theory and Computation* 11.8 (2015), pp. 3859–3865. DOI: 10.1021/acs.jctc.5b00488.

[207] Bjarne Amstrup, Gábor J. Tóth, Gábor Szabó, Herschel Rabitz, and András Lörincz. “Genetic algorithm with migration on topology conserving maps for optimal control of quantum systems”. In: *Journal of Physical Chemistry* 99.14 (1995), pp. 5206–5213. DOI: 10.1021/j100014a048.