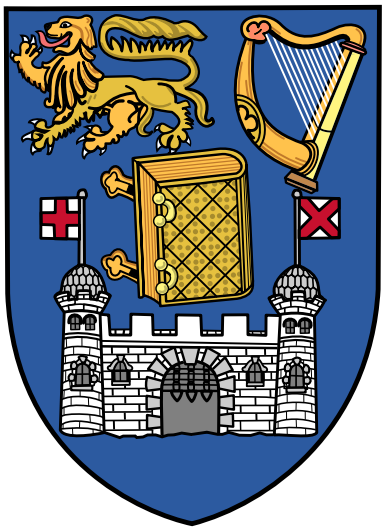


Optimisation of non-Markovian systems using tensor networks

Eoin Butler

A thesis submitted for the degree of

Doctor of Philosophy

at Trinity College Dublin

2024

Declaration

I declare that this thesis has not been submitted as an exercise for a degree at this or any other university and it is entirely my own work.

I agree to deposit this thesis in the University's open access institutional repository or allow the Library to do so on my behalf, subject to Irish Copyright Legislation and Trinity College Library conditions of use and acknowledgement.

I consent to the examiner retaining a copy of the thesis beyond the examining period, should they so wish (EU GDPR May 2018).

Signed: _____ Date: _____

Abstract

Structured environments that do not admit simple time-local descriptions display a wealth of features which are otherwise absent from unstructured environments. Here we use optimal control to learn how to perform quantum operations in systems under the influence of non-Markovian decoherence. Despite recent advancements, non-Markovian environments are difficult to simulate and complex to understand, which inhibits the ability to perform optimal control. We present an efficient method to find optimal controls for quantum systems coupled to non-Markovian environments, by using the process tensor to compute the gradient of an objective function.

We consider the case of a driven two level system coupled to a bosonic environment. We use optimal control to identify control protocols to perform state transfer on our system. We demonstrate how there is a trade-off between process duration and fidelity, and how longer processes can have higher fidelity by exploiting non-Markovian effects. We contrast this to the case of Hamiltonian dynamics where the quantum speed limit characterises the shortest time possible to perform state transfer. We describe how optimal control can be used to characterise the performance-speed trade-off. We propose a non-trivial mechanism for how the optimiser is suppressing decoherence while exploiting non-Markovian effects. We give an overview of how these results could be generalised to quantum gates.

Acknowledgements

Firstly, I would like to give big thanks to my supervisor Paul Eastham for his endless guidance and support over the past four years. His ability to always be on the correct side of an argument was as frustrating as it was beneficial to bringing me here today. I would also like to thank Gerald Fux for all of the discussions, contributions and get out of jail cards he has handed me. I would like to thank Brendon Lovett and Jonathan Keeling for their invaluable assistance with learning about process tensor and TEMPO methods.

The margin of the acknowledgements section is too small to write down all of the people both in and outside of physics that I have benefitted from. I'd like to thank my family and friends for the love and support they have given me through these past four years.

List of Publications

G. E. Fux, E. P. Butler, P. R. Eastham, B. W. Lovett, and J. Keeling, "Efficient Exploration of Hamiltonian Parameter Space for Optimal Control of Non-Markovian Open Quantum Systems", Physical Review Letters 126, 200401 (2021).

E. P. Butler, G. E. Fux, C. Ortega-Taberner, B. W. Lovett, J. Keeling, and P. R. Eastham, "Optimizing Performance of Quantum Operations with Non-Markovian Decoherence: The Tortoise or the Hare?", Physical Review Letters 132, 060401 (2024).

The process tensor algorithms as well as the gradient based optimization code are open source and contained in `OQuPy`, available on GitHub

Contents

1 Introduction **1**
- 1.1 Open Quantum Systems . . . 2
  - 1.1.1 Markovian methods . . . 5
- 1.2 Non-Markovian methods . . . 8
  - 1.2.1 Tensor networks in open quantum systems . . . 9
  - 1.2.2 Representations of dynamical maps . . . 11
  - 1.2.3 Feynman path integral . . . 16
  - 1.2.4 Quasi-adiabatic propagator path integral . . . 18
  - 1.2.5 Time-evolving Matrix-Product-Operators . . . 21
  - 1.2.6 The process tensor . . . 23
  - 1.2.7 Meaning of non-Markovianity . . . 27
  - 1.2.8 Measures of non-Markovianity . . . 27
- 1.3 Quantum Speed Limits . . . 30
  - 1.3.1 What is time in quantum mechanics . . . 31
  - 1.3.2 Finding the minimum time . . . 31
- 1.4 Physical Models . . . 35
  - 1.4.1 Specifying the environment . . . 35
  - 1.4.2 Semiconductor quantum dots . . . 35

2 Gradients using process tensors **37**
- 2.1 Introduction to optimal control . . . 37
  - 2.1.1 Extremising an objective function . . . 37
- 2.2 Physical Model . . . 38
  - 2.2.1 Parametrising the system Hamiltonian . . . 38
  - 2.2.2 State transfer . . . 39
- 2.3 Adjoint Method in PT-MPO . . . 39
  - 2.3.1 General objective function F . . . 42
- 2.4 Optimisation Results . . . 43

3 Optimising performance of open quantum systems **47**
- 3.1 Background . . . 47
- 3.2 Performance vs Fidelity . . . 48
  - 3.2.1 Hamiltonian dynamics . . . 49
  - 3.2.2 Optimising the performance of state transfer . . . 51
  - 3.2.3 Relationship between optimal control and the QSL . . . 53
- 3.3 Understanding non-Markovian Behaviour . . . 55
  - 3.3.1 Measures of non-Markovianity . . . 56
  - 3.3.2 Solving the Independent Boson Model . . . 56

3.3.3 Reversal of the polaron formation 61
3.4 Conclusions 63

4 Optimal control of SLM pulse shaping **65**
4.1 Pulse shaping using an SLM 66
4.1.1 Physical model 67
4.2 Results 68
4.3 State transfer 68
4.4 Optimisation of an ensemble 69
4.5 Relationship with previous work and conclusions 70

5 Optimal control of quantum gates **73**
5.1 Introduction 73
5.1.1 Fidelity of quantum gates 73
5.1.2 Transforms between representations of quantum channels 74
5.1.3 Back-propagating gates 76
5.2 Results 76

6 Summary **79**

Appendices **81**

A Second order Trotterisation **83**
A.1 Additional considerations for second order Trotterisation 83
A.1.1 Process Tensor 83
A.1.2 Gradient 85

Chapter 1

Introduction

Optimal control has been successfully applied to a broad variety of topics, ranging from rocket science to biological systems, and to human behaviour. During the days of the machine learning boon, it keeps its place as a fundamental algorithm for finding solutions to problems that cannot be done analytically. Humans use optimal control intuitively on a daily basis, from deciding what path to take to travel to our destination, to refining ingredients for a cooking recipe. Intuitively, all that is required to preform optimal control is a sense of feedback. In the case of finding directions, it is generally the time taken between departure and arrival, and in the cooking example, discerning taste is all that is required to create a feedback loop to improve the outcome over time. This thesis covers the application of optimal control to open quantum systems, seeking to minimise the loss of information from a quantum system to its environment.

The possible applications exploiting quantum coherence have been apparent since the nineteen–eighties [1, 2], and significant experimental realisations have been achieved recently [3–6]. The loss of coherence — decoherence — arises from the fact it is never possible to isolate a quantum system from its surroundings. Since the environment contains infinite degrees of freedom, quickly we realise that it is not possible — nor desirable — to treat the environment exactly. These environments can be simple, or highly structured, in which case identifying solutions to minimise decoherence is a highly non-trivial task. Buoyed by recent advancements in simulating the dynamics of strongly coupled and structured environments [7, 8], here we apply optimal control to find solutions to perform quantum operations while minimising decoherence.

Simple, Markovian, time-local equations of motion of our system can be obtained for unstructured environments that are weakly coupled to a system. This means that information and coherence is irreversibly lost to the environment. In this thesis we perform optimal control on non-Markovian [9, 10] systems, attempting to exploit the fact that information that is emitted to the environment can now be recovered. Now we find that obtaining future trajectory of the system requires knowing its history, and the difficulty in doing so scales exponentially with memory time. As such, examples of optimisation of non-Markovian systems are relatively scarce [11–19]. We apply an efficient method for obtaining the gradient of a cost or objective function [20, 21] to the process tensor [8, 22–24] to allow for gradient based optimisation of high dimensional parameter space.

We perform optimisation of state transfer on a model that resembles a quantum dot in a semiconductor substrate. We ask how fast can we perform state transfer in the physically

important condition, and we see how this is intimately related to the idea of a Quantum Speed Limit (QSL). We describe the relationship between optimal control and the QSL, and compare it with that in a system in the absence of a bath. We argue that there is no such idea of a fastest time possible to perform state transfer in an open system, rather that one should consider a trade-off between the process duration and the achieved fidelity. We show how we can use optimal control to find protocols which we believe exploit the fact that information can now flow from the environment back into the system. By comparison with an analytically solvable model [25], we propose a mechanism for which the optimiser is preventing the build-up of correlations between the system and the environment. We show some preliminary results about extending this method to efficient optimisation of quantum gates.

The introduction is structured as follows: Firstly we give a brief overview of the standard approach to simulating open quantum systems. In order to gain a better understanding of the meaning of non-Markovianity, we sketch how one can use approximations to derive a time-local Markovian equation of motion. We give an overview of tensor network methods and outline a graphical representation of high-rank tensors. Using this notation we describe the canonical representations of the dynamical map that maps an initial state to a final state under the influence of an environment. In order to obtain such dynamical maps we give a brief overview of the Feynman path integral. We then introduce the main algorithm for simulating non-Markovian systems, starting by building up a representation of the history of the system in section 1.2.4, the compression algorithm in section 1.2.5, and finally introducing the process tensor (Sec. 1.2.6). We wrap up section 1.2 with a brief discussion of the meaning of non-Markovianity. The remainder of the introduction is devoted to introducing the QSL, followed by an overview of the experimental considerations of a quantum dot, showing the origin for our model.

In the next chapter we will show how we can obtain an efficient representation of the gradient of the process tensor. There we detail the considerations for optimisation, both in the specific case we consider, and describe how to deal with more general objective functions not considered in this thesis.

1.1 Open Quantum Systems

Simply put, it is not possible to have a quantum system, such as a two level system, completely isolated from any external influence from other factors, be it other quantum systems, or more generally, an environment. While in principle it allows one to perform actions such as qubit reset [26], it makes rudimentary computational tasks difficult to impossible.

Suppose one has a two level system, The analogy that is generally used is that of a spin-1/2 particle which has two eigenvalues $\pm\frac{\hbar}{2}$, that correspond to the spin up, $|\uparrow\rangle$, and down, $|\downarrow\rangle$ normalised states. Suppose one is performing a computation on a *pure* state, i.e. a state of the form $|\psi\rangle = c_1|\uparrow\rangle + c_2|\downarrow\rangle$ and one wishes to perform a unitary rotation of all states around, say the S_x axis of the Bloch sphere. In the case where there is no interaction with an environment, the state just evolves as per the Schrödinger equation under a (possibly time dependent) Hamiltonian $H(t)$,

$$i\frac{d}{dt}|\psi(t)\rangle = H(t)|\psi(t)\rangle, \tag{1.1}$$

where $\hbar = 1$ unless explicitly stated otherwise for this thesis. The solution of this equation

can be expressed by introducing the unitary time evolution operator $U(t, t_i)$, which takes some state $|\psi(t_i)\rangle$, and evolves it to the state at time t as

$$|\psi(t)\rangle = U(t, t_i)|\psi(t_i)\rangle. \tag{1.2}$$

If the Hamiltonian is time-independent, then the time evolution operator is

$$U(t, t_i) = \exp[-iH(t - t_i)], \tag{1.3}$$

whereas for a time-dependent Hamiltonian

$$U(t, t_i) = \underset{\leftarrow}{T} \exp\left[-i \int_{t_i}^{t} d\tau H(\tau)\right], \tag{1.4}$$

where $\underset{\leftarrow}{T}$ is the time-ordering operator which orders time-dependent operators such that their time arguments increase from right to left.

Suppose one wishes to perform a rotation of angle ϑ around the S_x axis of the Bloch sphere. To do this, we can set the Hamiltonian to be time-independent and $H = S_x$, so

$$U(t, 0)_{R_{x,\vartheta}} = \exp[-iS_x\vartheta] = \begin{pmatrix} e^{-\frac{i\vartheta}{2}} + e^{\frac{i\vartheta}{2}} & e^{-\frac{i\vartheta}{2}} - e^{\frac{i\vartheta}{2}} \\ e^{-\frac{i\vartheta}{2}} - e^{\frac{i\vartheta}{2}} & e^{-\frac{i\vartheta}{2}} + e^{\frac{i\vartheta}{2}}, \end{pmatrix} \tag{1.5}$$

$$= \cos\left[\frac{\vartheta t}{2}\right] \mathbb{1} + i \sin\left[\frac{\vartheta t}{2}\right] S_x. \tag{1.6}$$

Here we have dealt with the case where the system is separated from the environment, and the evolution is said to be *unitary*. That is, the states evolve under a unitary operator Eq. 1.4, and $U^\dagger U = \mathbb{1}$, so we say the system is closed. In this thesis, we will use the terms *"closed", "unitary"* and also *"Hamiltonian dynamics"* interchangeably to refer to the dynamics of a quantum system that has no coupling to any environment. The reality however, is that it is impossible to decouple a system from some environment[1] [27].

Partitioning the Hamiltonian

To solve open quantum systems, we take the Hamiltonian that accounts for everything that we are interested in: the system, the environment, and any system-environment interactions (for clarity we denote the total Hamiltonian as H_{Tot}). We will now partition the total Hamiltonian into those three parts, the environment Hamiltonian, H_B, the system Hamiltonian, H_S, and a term that will contain any interaction between the system and the bath, H_{SB}. The total Hamiltonian can now be written as

$$H_{\text{Tot}} = H_S + H_B + H_{SB}. \tag{1.7}$$

The environment we will consider throughout this thesis is a large reservoir [2] of harmonic oscillators,

$$H_B = \sum_q \omega_q b_q^\dagger b_q. \tag{1.8}$$

The system is the object that is of interest to the experimentalist, that is, the object that one will measure expectation values for. In principle it can be anything but in practice it is

[1] We will use the terms *"bath"* and *"environment"* interchangeably to mean the same thing.

[2] Generally, a reservoir means an infinite number of degrees of freedom of the environment [27], but one can consider an environment that is not a reservoir, see e.g. the Jaynes-Cummings model

usually (somewhat) simple. Our system will always be a two level system with pseudo-spin levels $|\uparrow\rangle$ and $|\downarrow\rangle$, and the Hamiltonian will be some variation on

$$H_S(t) = h_x(t)S_x + h_y(t)S_y + h_z(t)S_z, \tag{1.9}$$

with some time-dependent control fields $h_\bullet(t)$. Classic examples of systems are trapped ions [5, 28, 29], superconducting qubits [30, 31], semiconductor quantum dots [32].

Both the system and environment Hamiltonian only contain system and environment operators, respectively, while the interaction term is the term in the Hamiltonian that contains both system and environment operators. The most general interaction Hamiltonian is

$$H_{SB} = \sum_q A_q \otimes B_q, \tag{1.10}$$

where A_q is some system operator and B_q is some environment operator. Here we will consider one form of the interaction Hamiltonian exclusively, the deformation potential, where

$$H_{SB} = \hat{O} \sum_q \left(g_q b_q + g_q^* b_q^\dagger \right). \tag{1.11}$$

A two level system coupled via the deformation potential to a bosonic bath is called the spin-boson model [9, 33], and is a field of active research.

The key point is that the total Hamiltonian H_{Tot} will evolve under unitary dynamics as described by Eq. 1.4, however evolving H_{Tot} by a unitary rotation for a typical open system will quickly become impossible since one now must diagonalise an infinite number of bath modes, which will contain terms corresponding to excitations caused by the system-bath interactions. The fact that we cannot write down the equation of motion of a total Hamiltonian is not necessarily a bad thing, as much — but not all — of this information will be irrelevant to the subsequent dynamics of the system.

In the following sections we will build up various methods to account for the influence of the environment on the system, truncating all of the superfluous information stored in the bath and only storing information that is relevant for the trajectory of the system, which for this thesis is the thing we ultimately care about. As such, instead of writing down equations of motion for the total density matrix of the system plus the environment, we instead deal with a much more desirable object, the reduced system density matrix. As a result of the interaction between the system and the environment this object will no longer evolve under unitary dynamics.

Reduced density matrix

We need not restrict ourselves to pure states as described above. We can also consider the case where a system is prepared in a *mixed* state, that is a quantum statistical ensemble of states. The density matrix is defined as

$$\rho(t) = \sum_i p_i \left|\psi_i(t)\right\rangle\!\left\langle\psi_i(t)\right|, \tag{1.12}$$

which is time-evolved by a unitary by

$$\rho(t) = U(t, t_0)\rho(t_0)U^\dagger(t, t_0), \tag{1.13}$$

and the corresponding equation of motion of the density matrix is

$$\frac{d}{dt}\rho(t) = -i[H(t), \rho(t)], \tag{1.14}$$

which is called the Liouville-von Neumann equation, or simply the Liouville equation. Expectation values of an operator $\hat{O}$ can be calculated using

$$\left\langle \hat{O} \right\rangle = \mathrm{Tr}[\hat{O}\rho]. \tag{1.15}$$

We can recast the Liouville-von Neumann equation in a form that is more similar to the classical Liouville equation,

$$\dot{\rho}(t) = \mathcal{L}(t)\rho(t), \tag{1.16}$$

where we have introduced the *Liouvillian super-operator* $\mathcal{L}$. A super-operator is an operator that acts on an operator to yield another operator. The Liouvillian super-operator is defined that

$$\mathcal{L}(t)\rho(t) \to -i[H(t), \rho(t)]. \tag{1.17}$$

From this definition constructing the Liouvillian super-operator may seem unnecessary, however the reason for doing this will become clear when we look at numerical recipes for finding the way the system evolves under the environment, where we will show explicitly how to write down the Liouvillian operator numerically. The time evolution of the density matrix if one knows the Liouvillian super-operator is

$$\rho(t) = \underset{\leftarrow}{T} \exp\left[\int_{t_0}^{t} d\tau \mathcal{L}(\tau)\right] \rho(t_0). \tag{1.18}$$

To make our life easier, instead of dealing with the total density matrix of the system and environment we wish to deal with an object that doesn't have a possibly infinite number of parameters. Since we define the system as the object that we will take expectation values for, all observables are of the form $\hat{O}_S \otimes \mathbb{1}_B$. We can therefore take the partial trace over the environment to yield the object of central interest, the reduced density matrix of the system,

$$\rho_S = \mathrm{Tr}_B\, \rho. \tag{1.19}$$

This means we can avoid writing down ρ_B explicitly, which is both unnecessary and computationally intractable. Expectation values can now be calculated as

$$\left\langle \hat{O} \right\rangle = \mathrm{Tr}_S[\hat{O}\rho_S], \tag{1.20}$$

we refer the reader to the standard literature [27] for further details, such as calculating the equation of motion of a reduced density matrix in the Schrödinger picture, Heisenberg picture.

1.1.1 Markovian methods

The standard, and simplest approach (see Breuer and Petruccione [27]) is to perform the Born-Markov approximation (generally referred to as the Markov approximation). This is where we first encounter the idea of "Markovianity", which ultimately means that the future of the evolution does not depend on the past, it only depends on the current state. The concept of classical "Markovianity" and "non-Markovianity" are well understood.

However the quantum analogues to both are a topic of hot debate [34], a discussion which we allude to later. For now, we regard quantum Markovianity as a series of approximations that yield a time-local equation of motion of motion of a system coupled to an environment, where the latter is "memoryless". This means that any information that flows from the system to the environment is irreversibly lost.

Let us start this discussion by introducing two concepts, a quantum operation and the dynamical map. Any operation or intervention on a system can be described by a completely-positive (CP) map [35]. Such operations could include applying a unitary or performing a projective measurement. The second is the quantum channel or dynamical map[3], $\mathcal{E}_{t_i:t_f}$. The dynamical map describes the evolution of the reduced density matrix of the system for any initial density matrix [27] (in this case taking us from time t_i to t_f). We will dedicate a significant amount of time to how to represent a quantum channel in section 1.2.2, but for now let us say that one such representation is the Liouvillian super-operator. For simplicity we say that the quantum channel is both completely positive, and trace preserving (CPTP), which ought be the case since $\mathrm{Tr}[\rho] = \sum_i p_i$ which is a sum of probabilities which ought to add up to unity[4]. We now introduce the first rigorous definition of Markovianity, CP-divisibility. CP-divisibility means that a dynamical map $\mathcal{E}_{0:t}$ can be decomposed into two dynamical maps [10, 36].

$$\mathcal{E}_{0:t} = \Upsilon_{t':t}\mathcal{E}_{0:t'}, \tag{1.21}$$

where $\Upsilon_{t':t}$ is a CPTP map and this decomposition can be done for all $t' \in [0, t]$.

We now jump back from the definitions, to more practical arguments. The simplest way to solve an open system is to derive equations of motion that are "time-local", that is differential equations that only depend on the current time, and do not require knowledge of the history of the solutions. This is the origin of the description "Markovian". More precisely, we write down a Markovian master equation as follows [27]

$$\dot{\rho}_S(t) = \mathcal{L}(t)\rho(t), \tag{1.22}$$

where we see that the equation of motion only depends on the current time. The most general form of a time-local master equation is Gorini-Kossakowski-Sudarshan-Lindblad master equation, which we refer to as the Lindblad master equation,

$$\mathcal{L}\rho_S = -i[H, \rho_S] + \sum_k^{d^2-1} \gamma_k \left(A_k \rho_S A_k^\dagger - \frac{1}{2} A_k^\dagger A_k \rho_S - \frac{1}{2} \rho_S A_k^\dagger A_k \right), \tag{1.23}$$

where H is close to, but not quite the system Hamiltonian (see microscopic derivation for a specific example). This is the most general form of of a generator of a quantum dynamical semigroup. As far as we are concerned, it is the most general equation of motion of a quantum channel that is CP-divisible[5]. The first term alone is just the Liouville equation generating unitary dynamics on the reduced density matrix, the rest of the account for the influence of the environment on the system, and include dissipation. It is derived, not using microscopic theories, rather rigorous arguments using quantum dynamical semigroups (see ref. [27] and the references therein). Nevertheless, for clarity of understanding the Markov approximation within different contexts, it is instructive to provide a sketch of a derivation

[3] we use "dynamical map" and "quantum channel" interchangeably but some authors make a distinction between the two

[4] See Nielsen and Chuang [35] for a more precise discussion

[5] The above discussion is true for experiments that are prepare-evolve-measure type. See Ref.[34] for a more precise discussion.

of this equation using a microscopic example to gain an intuitive feel of what the failure of the Markov approximation should look like.

Microscopic derivation

We sketch the important steps of the microscopic derivation for an illustration of the approximations and to give the reader an idea of when they would be appropriate. We start by introducing the interaction picture Liouville equation. This is found by introducing the "free" Hamiltonian $H_0 = H_S + H_B$, and transform operators into the interaction picture using

$$\tilde{O}(t) = e^{iH_0(t-t_i)} O e^{-iH_0(t-t_i)} \tag{1.24}$$

$$= U_0^\dagger(t, t_i)\rho(t)U_0(t, t_i) \tag{1.25}$$

where $\tilde{O}$ is an operator in the interaction picture. The Liouville equation for the evolution of the total density matrix in the interaction picture is

$$\frac{d}{dt}\tilde{\rho}(t) = -i[\tilde{H}_{SB}, \tilde{\rho}(t)]. \tag{1.26}$$

Integrating this equation gives

$$\tilde{\rho}(t) = \rho(0) - i\int_0^t ds[\tilde{H}_{SB}(s), \tilde{\rho}(s)], \tag{1.27}$$

which we can now substitute back into eq 1.26, and since we are interested in the dynamics of the reduced density matrix we take the partial trace over the environment to give

$$\frac{d}{dt}\tilde{\rho}_S(t) = -\int_0^t ds\, \mathrm{Tr}_B[\tilde{H}_{SB}(s), [\tilde{H}_{SB}(s), \tilde{\rho}(s)]], \tag{1.28}$$

where we assume that $\mathrm{Tr}_B[\tilde{H}_{SB}(s), \rho(0)] = 0$. Equation 1.28 is still exact, so to make it tractable to solve we first perform the Born approximation, which is that the influence of the system on the bath is small so the bath is approximately unchanged by the system. The bath is therefore approximated to be time independent and the total density matrix is approximately a product state of the system and environment density matrices:

$$\tilde{\rho}(t) \approx \tilde{\rho}_S(t) \otimes \rho_B. \tag{1.29}$$

This gives us a simplified equation of motion

$$\frac{d}{dt}\tilde{\rho}_S(t) = -\int_0^t ds\, \mathrm{Tr}_B[\tilde{H}_{SB}(s), [\tilde{H}_{SB}(s), \tilde{\rho}_S(s) \otimes \rho_B]], \tag{1.30}$$

however, it is still difficult to solve, as propagating the system density matrix forward requires keeping track of the past history. This is where the Markov approximation, where we send $\tilde{\rho}_S(s) \to \tilde{\rho}_S(t)$ appears. The Markov quantum master equation is now

$$\frac{d}{dt}\tilde{\rho}_S(t) = -\int_0^t ds\, \mathrm{Tr}_B[\tilde{H}_{SB}(t), [\tilde{H}_{SB}(s), \tilde{\rho}_S(t) \otimes \rho_B]], \tag{1.31}$$

This is called the Redfield equation, and is local in time but not a Markov master equation since it depends on the choice of initial preparation. The final approximation is to send $\tilde{H}_{SB}(s) \to \tilde{H}_{SB}(t-s)$ and let the upper limit of the integral go to infinity. This is

reasonable provided the integral disappears fast for s beyond the timescale on which the bath correlation functions decay. Finally we arrive at the Markov master equation

$$\frac{d}{dt}\tilde{\rho}_S(t) = -\int_0^\infty ds\, \mathrm{Tr}_B[\tilde{H}_{SB}(t), [\tilde{H}_{SB}(t-s), \tilde{\rho}_S(t) \otimes \rho_B]]. \tag{1.32}$$

This is a Markovian master equation, however it is not guaranteed to define the generator of a dynamical semigroup, and in order to do so, one performs the rotating-wave approximation, or secular approximation. This approximation averages out terms that oscillate rapidly on timescales that one is interested in, and we refer the reader to the literature [27] for details.

Performing the Born, Markov and secular approximations one is guaranteed to obtain a CP-divisible map, and a dynamical semigroup, therefore the resultant dynamical map can be written using the Lindblad master equation. The Born approximation assumes the system-environment coupling is weak such that the bath density matrix is approximately time-independent. As discussed in Breuer, that is not to say that there are no system induced excitations in the environment, rather that they are not resolved on the timescale set by the Markov approximation. The Markov approximation says that the timescale over which the state of the system changes is much slower than the time taken for the bath correlation functions to decay, so we can no longer resolve times shorter than this environment correlation time.

This general method for obtaining a Markovian quantum channel $\mathcal{E}$ by deriving a Lindblad master equation can be performed on various systems in parameter regimes that allow for the approximations. The classic introductory example is the quantum optical master equation, considered in Breuer [27], where in the quantum optical regime the approximations are well satisfied. Another application is a derivation of the effects of phonons on semiconductor quantum dot excitons [37], which again is valid provided one stays in a certain parameter regime. This is very similar to the model we will use for the remainder of this thesis, so we will delay a discussion to the limits of the validity of this equation until later.

1.2 Non-Markovian methods

Simple, time-local equations of motion are have yielded a rich area of physics, including optimal-control examples [3–5]. The issue is that they are poor approximations for interesting quantum devices [9, 10]

The problem with non-Markovian methods is they are extremely computationally costly. To give an example: consider the Augmented Density Tensor (ADT). We will defer a comprehensive introduction to below, but for now we give a cartoon picture that it's an object that stores the history of the system density matrix over a bath correlation time τ_C. While exploiting the fact that excitations in the environment caused by the system after time τ_C are not relevant to the future evolution, we are still nevertheless left with an object that scales poorly. If we have to store every time-step Δt over τ_C to give k time-steps, that means the size of this tensor is d_S^{2k} numbers, which scales exponentially with τ_c.

This simple, loosely described, relatively naive way of evolving a non-Markovian system shows the difficulty in dealing with non-Markovian open quantum systems. As a result, there are relatively few examples of optimal control in non-Markovian systems other than

the methods we are about to describe. One such approach is to expand the system to incorporate a few modes fo the environment [11–13], however this becomes intractable when one considers more than a few modes of the environment. Optimal control has been applied using a time-local dissipator describing non-Markovian dynamics using lowest order perturbation theory [14], as well as hierarchical equations of motion [15]. Both of these techniques are effective if the environment can be modelled as a small number of Lorentzians. Optimal control has been performed using stochastic Liouville equations [16]. Another approach is to assume that the dissipation is described by a fixed time-local dissipator [17–19]. One issue with this strategy is that one must consider the effect of the time-dependent control fields on the dissipator [38].

1.2.1 Tensor networks in open quantum systems

This thesis is based on representing quantum channels as products of large amounts of tensors. The standard way to write down non-trivial tensor network operations is Einstein notation, however to simplify discussions we will build up an alternative way to describe tensors and how to sum over indices.

Einstein notation means that the components of a tensor A can be written as A_{ij} where i and j are the two indices. The rank of a tensor is the number of indices required, so in our example the fact that only two coordinates i and j are needed to describe the component A_{ij}, A is a matrix. The second convention is that repeated indices are summed over, so $C^{\alpha\gamma} = \sum_\beta A^{\alpha\beta}B^{\beta\gamma}$ can be written by dropping the summation sign. Since we make no distinction between covariant and contravariant vectors; our choice of superscripts and subscripts is mostly down to avoiding confusion between denoting an index and an exponent. While Einstein notation is a powerful method that can perform all of the operations we are about to describe, it becomes unwieldy for describing a large quantity of tensors to be summed over, such as those seen in Matrix Product States (MPS) and Density Matrix Renormalisation Group (DMRG). To given an example, suppose we wish to perform manipulations on an MPS. An example could be the matrix product operator for the propagator of the TEMPO (to be introduced later) algorithm, which is is written using Einstein notation as

$$\mathcal{B}^{j_n,j_{n-}\ldots,j_1}_{i_{n-1}\ldots,i_1} = [b_0]^{j_n}_{\alpha_1}\left(\prod_{k=1}^{n-2}[b_k]^{\alpha_k,j_{n-k}}_{\alpha_{k+1},i_{n-k}}\right)[b_{n-1}]^{\alpha_{n-1},j_1}_{i_1}, \tag{1.33}$$

which, taken out of context, is very difficult to visualise[6].

Now we introduce tensor network diagrams, and we will see how this graphical notation simplifies equations (such as Eq. 1.33) and discussions. This notation dates back to Penrose, however in order to adapt it to be useful for open quantum systems we will strongly follow the notation introduced by Wood et al. [39]. Each tensor is represented by a node, depicted as either a circle, square, or triangle. An index for a tensor is represented by drawing a leg. The main point is that when two indices (legs) are connected it is assumed that they are summed over. The number of legs (indices) indicates the rank[7] of the tensor. This is shown graphically in Fig. 1.1. A scalar is represented by zero legs (a), and we assume that a vector in Hilbert space is a node with a single leg pointing to the left (b). A rank-2 tensor (a matrix), is a two legged object (d). Here we encounter a

[6] This is shown diagrammatically in Fig. 1.14.

[7] Here rank simply means the number of indices used to specify a tensor, though the term can also refer to the number of non-zero singular values

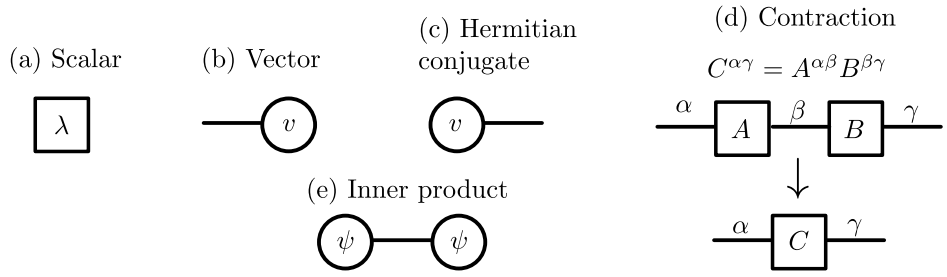


Figure 1.1: Various tensor networks: (a) a scalar, with zero legs, (b) a vector $|v\rangle$ with one leg pointing to the left, (c) the Hermitian conjugate of the vector $\langle v|$, (d) matrix-matrix multiplication, a simple example of a tensor contraction. (e) Inner product $\langle\psi|\psi\rangle$. Diagram adapted from Ref. [39].

discrepancy between the conventions in Wood et al. and the standard TEMPO method literature. Wood et al. stipulate that the orientation specifies whether they represent a vector in Hilbert space, the Hermitian conjugate of a vector, or an operator. In the TEMPO literature, spatial orientation has the meaning of the progress of time, and the difference between a rank 0,1,2... tensor is solely the number of legs it has. We will sketch how these two methods can be married. For now we will adapt the convention by Wood et all, that a vector in Hilbert space $|v\rangle$ has a single leg pointing to the left Fig. 1.1 (b), and the dual of that vector $\langle v|$ is defined as having a single leg pointing to the right (c). Finally, an operator has legs pointing in both directions (d). We will try to avoid confusion by clearly stating when the meaning of the orientation of a tensor network leg has changed. The choice of this direction (which is the opposite direction to conventional quantum circuits) was chosen to bear resemblance to Dirac notation. It means that the inner product $\psi_i^*\psi^i = \langle\psi|\psi\rangle$ can be expressed rather succinctly in Fig. 1.1 (e).

Now that we have defined a meaning to the orientation of a leg, and state that changing the orientation of a leg from left to right implies taking the Hermitian conjugate, we can now define a transpose and conjugate. Firstly the complex conjugation of an operator is denoted by a bar, Fig. 1.2 (a). Now since the Hermitian conjugate of a vector is defined with the leg pointing to the left, we can define taking the transpose of a vector as bending the leg around to the right (b). Similarly this can be done with operators (c), since $A^T_{ij} = A_{ji}$. The trace now takes the simple form of connecting the two legs together since $\mathrm{Tr}[A_{ij}] = A_{ii}$ (d).

For this chapter we will represent summing over a basis by using a hatched box, such the eigenvalue decomposition can be shown in Fig. 1.3 (a). Sliding an operator around the bend as shown in Fig. 1.3 (b) transposes the matrix, and we refer the reader to the full description in Wood et al. [39] for a full justification. The real power in this notation can be seen when we introduce row and column vectorisation as shown in Figs. 1.4 (a) and (b). By bending the indices from one side to the other and saying that they point along the same direction we have achieved vectorisation. Whether row vectorisation is defined as bending the leg above (as is the case here), or below is arbitrary, so long as we are consistent.

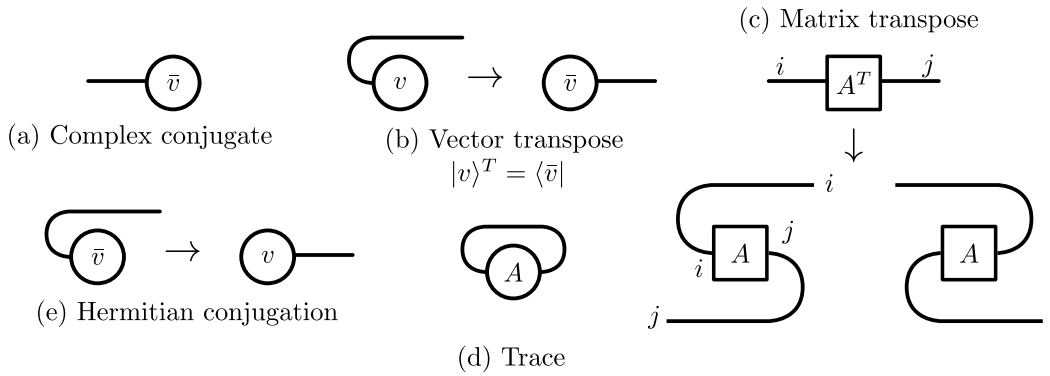


Figure 1.2: Manipulating tensors: (a) complex conjugation, (b) transposing a vector, (c) transposing a matrix, (d) obtaining the trace of a matrix. Hermitian conjugation (e) follows from diagrams (a) and (b). Diagram adapted from Ref. [39]

.

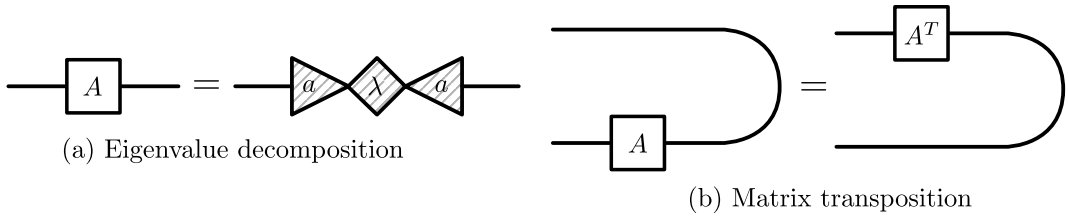


Figure 1.3: (a) Eigenvalue decomposition, (b) transposing a matrix. Diagram adapted from Ref. [39].

1.2.2 Representations of dynamical maps

Liouville super-operators

There are many equivalent ways to represent a CPTP dynamical map. The first form we introduce is the Liouville super-operator, which will be the dominant representation of a dynamical map in this thesis. It will become clear why we have spent considerable time detailing how to represent operations such as vectorisation in graphical notation.

To start off let us define the time evolution of a density matrix $\rho_0 = \sum_i p_i |\psi_i\rangle\langle\psi_i|$ given by

$$\rho(t) = U(t, t_0)\rho(t_0)U^\dagger(t, t_0).$$

The density matrix is given in diagram 1.5 (a), and assuming the above conventions, the time evolution is shown in diagram 1.5 (b).

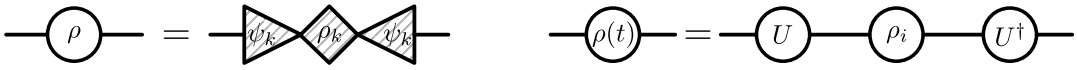
Here, we can now show an explicit way to construct the Liouville equation 1.16, $\dot{\rho}(t) = \mathcal{L}(t)\rho(t)$. If we assume that the Liouvillian super-operator is time-independent, the formal solution is

$$\rho(t) = \exp[\mathcal{L}(t - t_0)]\rho(t_0). \tag{1.34}$$

Now so far we have not given any concrete ways to construct $\mathcal{L}$, other than defining it as $\mathcal{L}\rho = -i[H, \rho]$. We start by constructing the case where a density matrix evolves unitarily under a time-independent Hamiltonian H of a closed system. The final density matrix $\rho(t)$ is given by diagram (b) in Fig. 1.5. The key step is that we vectorise the

(a) Column vectorisation (b) Row vectorisation

Figure 1.4: Vectorisation:(a) Column vectorisation, (b) Row vectorisation, and showing an operator that takes two legs, both of Hilbert space dimension d_{HS} and converts it to a single leg of dimension d_{HS}^2. (c) converting a row vectorised vector a column vectorised vector by exchanging legs. Diagram adapted from Ref. [39].



(a) Density Matrix (b) Time evolution

Figure 1.5: Construction and time-evolution of a density matrix. (a) a density matrix as an outer product of vectors. (b) the time evolution of the density matrix using unitary operators. Diagram adapted from Ref. [39].

density matrix (Fig. 1.6 (a)). After vectorising the density matrix, we now say that we are in Liouville space. By sliding the right hand unitary along the wire (Fig. 1.6 (b)), we can group together the two unitary operators (in tensor network notation grouping two operators together is taking a tensor product), and define this as the Liouville super-operator (which we call L in this diagram). L is our Liouville propagator. It propagates a vectorised density matrix forward in time under the Liouvillian $\mathcal{L}$. Once the network has been multiplied out we are free to unvectorise the result to get the evolved density matrix $\rho(t)$. We see that introducing the Liouvillian super-operator has allowed us to represent evolution under a dynamical map as a matrix-vector multiplication. Similarly commutator relations are now matrix vector multiplications in Liouville space. Liouville super-operators can be constructed for left (d) and right (c) acting operators $\hat{O}_L$ and $\hat{O}_R$ by taking tensor products with the identity. The trace can also be taken in Liouville space by contracting with an operator (which we will soon call the cap in TEMPO language) which is the vectorised identity[8]. The Liouville super-operator representation can be used to represent any CP-map [39].

We can go a step further and show the tensor network for the discretised time-dependent Liouville equation

$$\rho(t) = \underset{\leftarrow}{T} \exp\left[\int_{t_0}^{t} d\tau \mathcal{L}(\tau)\right] \rho(t_0) \to e^{\mathcal{L}(t_n)\Delta t} \dots e^{\mathcal{L}(t_2)\Delta t} e^{\mathcal{L}(t_1)\Delta t} \rho(t_0), \tag{1.35}$$

using vectorised density matrices as

$$\rho^i(t) = [L_n]^{\alpha_i}_{\alpha_{n-1}} \left(\prod_{k=0}^{n-1} [L_k]^{\alpha_k}_{\alpha_{k-1}} \right) \rho^{\alpha_0}(t_0), \tag{1.36}$$

$$L_k = \exp[\mathcal{L}(k)\Delta t] \tag{1.37}$$

[8]In fact, the vectorised identity matrix is also the unnormalised Bell state $|\Phi^+\rangle = \sum_i |i\rangle \otimes |i\rangle$; see Wood et al. [39] for further details.

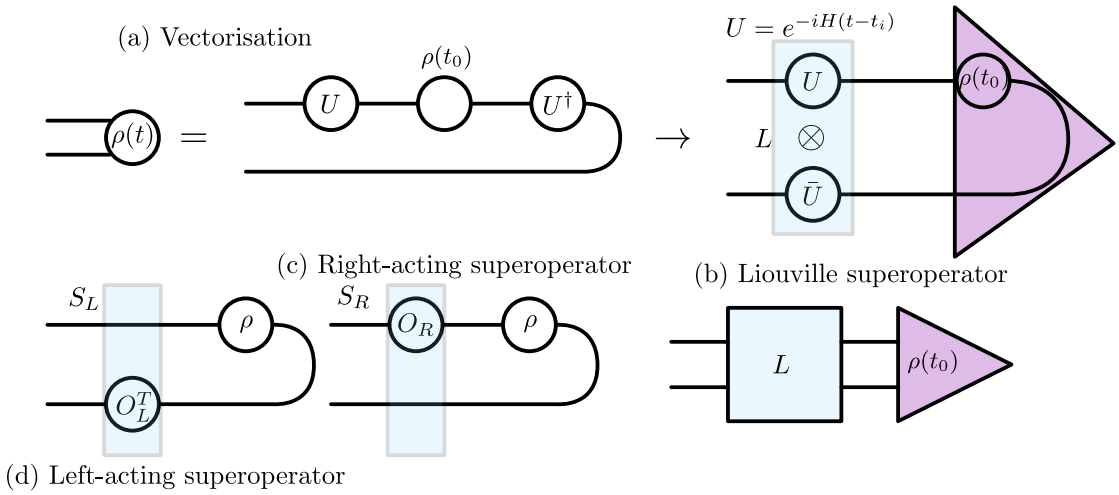


Figure 1.6: Construction of a Liouville super-operator for a state evolving as a closed system under a unitary U. (a) Vectorisation of the density matrix at the final time. (b) Sliding the unitary around the wire, showing how the tensor product of the two unitaries is a super-operator in Liouville space. Finally, construction of a Liouville super-operator consisting of a single right (c) or left (d) acting operator in Hilbert space on the density matrix by taking the tensor product of the operator and the identity.

In principle this equation can account for non-Markovian behaviour, provided one knows how to construct a non-Markovian Liouvillian super-operator. This is a difficult task and our non-Markovian solutions will not look like this equation. If we restrict ourselves to a Markov case, we can write the Lindblad equation as a Liouvillian super-operator, as shown in figure 1.7 (a). Equation 1.36 now looks like diagram 1.7 (b). Here we can see the simplicity and ease of comprehension brought by this tensor network notation. The general

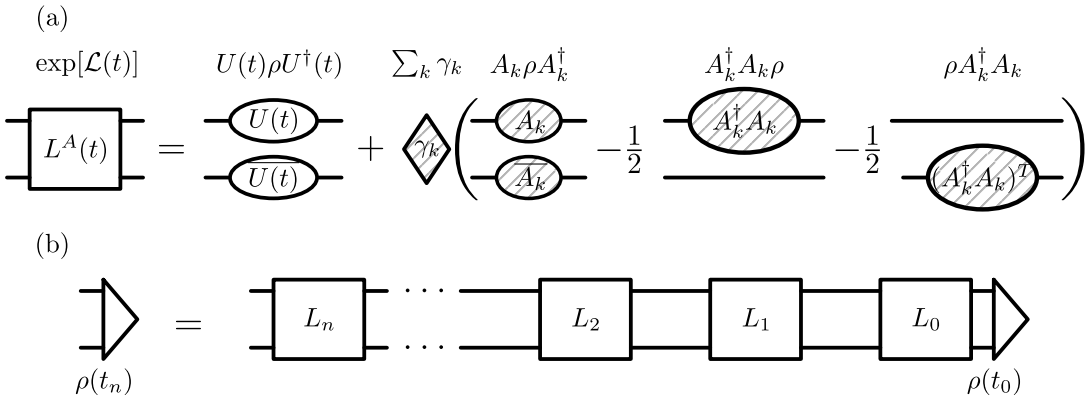
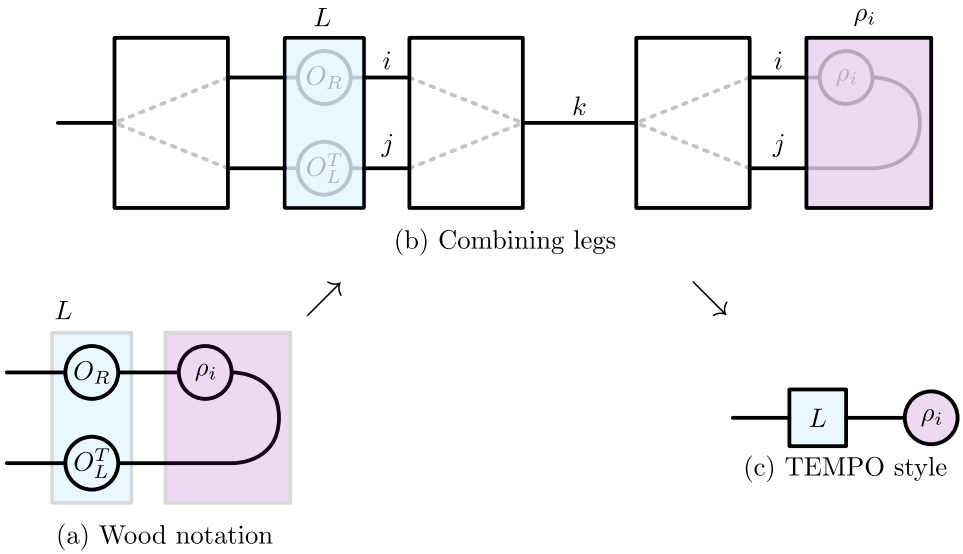


Figure 1.7: Time discretised evolution of a density matrix using a Liouvillian of a Lindblad dissipator with jump operators A_k and dissipation rates γ_k. (a) Constructing the Liouvillian at a discrete time. (b) Evolving the initial density matrix, showing the time ordering of the integration in discrete time.

formula for applying a dynamical map in the Liouville super-operator representation is

$$\mathcal{E}(\rho)_{ij} = L_{ijkl}\rho_{kl}. \tag{1.38}$$

This is shown diagrammatically in figure 1.6 (b). The discretised product of Liouvillians in Fig. 1.7 (b) can be brought into the form of fig 1.6 (b) by contracting the n super-operators together to yield a single super-operator as in diagram 1.6 (b).



(b) Combining legs

(a) Wood notation

(c) TEMPO style

Figure 1.8: Connecting Wood conventions and TEMPO conventions. (a) Performing a standard operation $\rho_f = O_R \rho O_L$. This has been vectorised into Liouville space, where the operation is now $\rho_f = L\rho_i$. (b) An operation that converts two legs of dimension d_{sys} into a single leg of dimension d_{sys}^2, with the map described in Eq. 1.39. We are now free to simplify the diagram to just having one leg (c). Assumptions about the meaning of the orientation of legs in Fig. (c) are now unnecessary, so we are free to reassign the meaning to something else, such as time.

Having used the convention of Wood et al. regarding the meaning of spatial orientation of the legs, we now introduce how we can transform from this notation to the standard TEMPO notation. Each index in diagram 1.6 (b) is the size of the system Hilbert space dimension d_{sys}. We are free to combine these two indices into a single index of size d_{sys}^2. An example of the index conversion map for row vectorisation (which is what `numpy` does by default) is

$$\begin{aligned} A^{ij} &\to [\text{vec}[A]]^k \quad \dim[A] = (m \times n) \\ i &= \texttt{k//m} \\ j &= \texttt{k\%m} \\ k &= \texttt{mi + j}. \end{aligned} \tag{1.39}$$

Which in matrix form for a (2×2) matrix looks like

$$\begin{bmatrix} A^{00} & A^{01} \\ A^{10} & A^{11} \end{bmatrix} = \begin{bmatrix} A^{00} \\ A^{01} \\ A^{10} \\ A^{11} \end{bmatrix}. \tag{1.40}$$

Note that here we have drawn the result of a row vectorisation as a column matrix. This is to be consistent with states in physics (i.e. kets) being represented by column vectors, which is the spirit of the spatial orientation convention of Wood. This is the form of the

legs that is commonly seen in the TEMPO literature, where a single leg of d_{sys}^2 can really be decomposed into two legs where one leg acts to the left of the operator, and the other to the right. This can be seen visually in diagram 1.8. To be clear, for this section we explicitly draw the two legs for clarity. From section 1.2.4 onwards we use the TEMPO convention where each leg is a single vectorised leg of dimension d_{sys}^2, and we drop the idea of spatial orientation denoting Hermitian-conjugation, rather the direction in space on the diagram is the order of time. We will clearly state any exceptions to this convention.

Kraus representation

The most intuitive representation for a CPTP map is the Kraus representation. Kraus' theorem states that a map $\mathcal{E}$ is CPTP if and only if it can be written in the form [35, 39, 40]

$$\mathcal{E}(\rho) = \sum_\alpha K_\alpha \rho K_\alpha^\dagger, \tag{1.41}$$

where the Kraus operators satisfy the completeness relation

$$\sum_\alpha K_\alpha^\dagger K_\alpha = \mathbb{1} \tag{1.42}$$

This is shown diagrammatically in figure 1.9 The choice of the Kraus operators is not

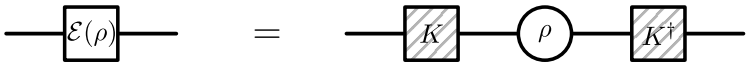


Figure 1.9: Using the Choi matrix to apply the dynamical map to a given density matrix. Diagram adapted from Ref. [39].

unique. A unique choice is the *canonical* Kraus representation [39, 41], where

$$\text{Tr}[K_\alpha^\dagger K_\beta] = \lambda_\alpha \delta_{\alpha\beta} \tag{1.43}$$

We use this representation to compute the fidelity of a quantum gate [42] in chapter 5.

Choi Matrix

The Choi matrix representation is an application of the Choi-Jamiołkowski isomorphism which gives a bijection between linear maps and linear operators [39, 43]. The Choi-Jamiołkowski isomorphism can be thought of having the quantum channel $\mathcal{E}$ acting on half of the unnormalised Bell state $|\Phi^+\rangle = \sum_i |i\rangle \otimes |i\rangle$, so the Choi matrix can be defined as[9]

$$\Lambda = (\mathbb{1} \otimes \mathcal{E}) \left|\Phi^+\right\rangle\!\left\langle\Phi^+\right|. \tag{1.44}$$

More intuitively, we can think of the Choi matrix as being a vectorisation of the Kraus operators (this operation itself can be seen as performing the Kraus map on half of the unnormalised Bell states). The evolution of a state using the Choi matrix as a representation of the quantum channel is given by

$$\mathcal{E}(\rho) = \text{Tr}_S[(\rho^T \otimes \mathbb{1}_{S'})\Lambda], \tag{1.45}$$

where S' is the same dimension of the system, but we emphasise that we trace over the system density matrix part, leaving open two indices that are the evolved state (which

[9]The Choi matrix Λ is dependent on the vectorisation, here we assume row vectorisation since that is what `numpy` does, see Wood et al. for further details on column vectorisation.

has the same dimension of the system). Alternatively, this can be expressed in Einstein notation as

$$\mathcal{E}(\rho)_{jl} = \Lambda_{ijkl}\rho_{ik}. \tag{1.46}$$

Both of these become easier to understand in tensor network notation. We see below that

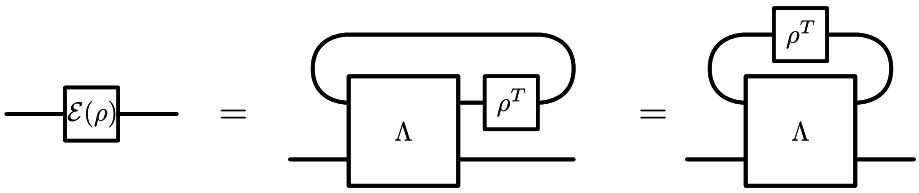


Figure 1.10: Kraus' theorem in tensor network form. Diagram adapted from Ref. [39].

the Choi matrix is just a reformulation of the Kraus operators.

We will further reinforce the message that all of these methods are simply alternative ways to represent the evolution of an open system when we describe how to transform between all three representations. In particular, we will see that the Choi matrix and Liouville super-operator are equivalent by reshuffling the indices, and the canonical Kraus representation can be obtained via the spectrum of the Choi matrix. Then one can ask, if they are all equivalent, why are we spending time introducing all three? The answer to this is that they make it easier to perform certain calculations. As stated previously, Pedersen et al. [42] have a formula for evaluating the average fidelity of a quantum gate — without requiring one to average over many different input states, which is a highly non-trivial task which makes optimisation of a quantum gate possible, as we shall explain in chapter 5. We can recast the soon-to-be introduced process tensor as a Choi matrix in order to obtain a measure of non-Markovianity. Finally, we shall mostly be working with the super-operators, as they compute commutators in a way that is clear and can be time-discretised, allowing us to build up the main tool we use in this thesis — the process tensor.

1.2.3 Feynman path integral

The standard approach to quantum mechanics is based on canonical quantisation of physical observables, building up an associated operator algebra. Instead, here we introduce the Feynman path integral [44–46], which takes an alternative approach [47]. The method we focus on in this thesis for simulating non-Markovian dynamics is based on formulating the problem as a Feynman path integral, and we will later show, in Sec. 3.2.3, how this can be used to solve an analytical problem. The path integral constructs the propagator,

$$U(q', q; t) = \left\langle q' \middle| e^{-iHt} \middle| q \right\rangle \tag{1.47}$$

which is the probability amplitude for a particle to propagate between generalised coordinates q and q'

Here we provide a general sketch of the path integral strategy and refer to the citations above and reference [47] for further details. The general idea is to divide the unitary evolution operator into infinitesimal time-steps,

$$e^{-iHt} \to \left[e^{-iH\Delta t}\right]^N. \tag{1.48}$$

Frequently we can use simplifications such as expanding the Hamiltonian like

$$e^{-iH\Delta t} \to e^{-iT\Delta t}e^{-iV\Delta t} + \mathcal{O}(\Delta t^2) \tag{1.49}$$

where the error vanishes without approximation since we will take the limit of $\Delta t \to 0$. The key is that at each time slice we can insert one or more resolutions of the identity. This allows us to write out the propagator as

$$\left\langle q' \middle| \left[e^{-iH\Delta t}\right]^N \middle| q \right\rangle = \left\langle q' \middle| \mathbb{1} e^{-iT\Delta t}e^{-iV\Delta t} \mathbb{1} e^{-iT\Delta t}e^{-iV\Delta t} \mathbb{1} \ldots \mathbb{1} e^{-iT\Delta t}e^{-iV\Delta t}. \middle| q \right\rangle \tag{1.50}$$

The resolutions of the identity are chosen to make it easy to evaluate the operators. An example is

$$\begin{aligned} \mathbb{1} &= \int dq_n \, |q_n\rangle\langle q_n| \\ \mathbb{1} &= \int dq_n \int dp_n \, |q_n\rangle\langle q_n|p_n\rangle\langle p_n| \, . \end{aligned} \tag{1.51}$$

The second choice of $\mathbb{1}$ is chosen such that the operators $\hat{V}$ act on eigenstates to the right,

$$\exp\left[-i\hat{V}\delta t\right] |q_n\rangle = \exp\left[-iV(q_n)\delta t\right] |q_n\rangle \, , \tag{1.52}$$

and $\hat{T}$ acts on eigenstates to the left,

$$\langle p_n| \exp\left[-i\hat{T}\delta t\right] = \langle p_n| \exp\left[-iT(p_n)\delta t\right] , \tag{1.53}$$

where we have temporarily put hats on $\hat{T}$ and $\hat{V}$ to emphasise that in the above equations they have gone from being operators to complex numbers.

This means that instead of dealing with non-commuting quantum operators, we can recast the propagation of states in terms of an integral of complex numbers, at the expense of requiring a sum over all possible trajectories of the quantum particle. The final step is to take the continuous limit, which will yield

$$D(p,q) \sim \lim_{N\to\infty} \prod_m \prod_n dp_m dq_n. \tag{1.54}$$

The Hamiltonian formulation of the path integral is

$$\left\langle q' \middle| e^{-iHt} \middle| q \right\rangle = \int D(p,q) \exp\left[i \int_0^t dt' (p\dot{q} - H(p,q))\right] , \tag{1.55}$$

and the Lagrangian formulation is

$$\left\langle q' \middle| e^{-iHt} \middle| q \right\rangle = \int D(q) \exp\left[i \int dt' L(q,\dot{q})\right] . \tag{1.56}$$

1.2.4 Quasi-adiabatic propagator path integral

Here we present the first numerical representation of a non-Markovian dynamical map, the Quasi-Adiabatic propagator Path Integral (QUAPI). QUAPI propagates an extended representation of the system that includes its history, called the augmented density tensor (ADT), forward in time. QUAPI can be used to obtain multi-time correlations, but in order to keep the discussion simple we restrict ourselves to expectations of the reduced density matrix, and refer the reader to the literature for further details [7, 48–50].

The evolution of the reduced density matrix of a system between some position coordinates s' and s''

$$\rho_S(s'', s', t) = \mathrm{Tr}_B \left\langle s'' \middle| e^{-iHt} \rho(0) e^{iHt} \middle| s' \right\rangle \tag{1.57}$$

is given by the path integral

$$\begin{aligned}\rho_S(s'', s'; t) = \int ds_0^+ \int ds_1^+ \cdots \int ds_{N-1}^+ \int ds_0^- \int ds_1^- \cdots \int ds_{N-1}^- \\ \left\langle s'' \middle| e^{-iH_S\Delta t} \middle| s_{N-1}^+ \right\rangle \ldots \left\langle s_1^+ \middle| e^{-iH_S\Delta t} \middle| s_0^+ \right\rangle \left\langle s_0^+ \middle| \rho_S(0) \middle| s_0^- \right\rangle \left\langle s_0^- \middle| e^{iH_S\Delta t} \middle| s_1^- \right\rangle \ldots \\ \left\langle s_{N-1}^- \middle| e^{iH_S\Delta t} \middle| s'^- \right\rangle I(s_0^+, s_1^+ \ldots s_{N-1}^+, s_0^- \ldots s_{N-1}^-, s'', s'; \Delta t),\end{aligned} \tag{1.58}$$

where the influence functional

$$\begin{aligned}I(s_0^+, s_1^+ \ldots s_{N-1}^+, s_0^- \ldots s_{N-1}^-, s'', s'; \Delta t) = \mathrm{Tr}_B \Bigg[\\ e^{\frac{-iH_{\mathrm{env}}(s'')\Delta t}{2}} e^{\frac{-iH_{\mathrm{env}}(s_{N-1}^+)\Delta t}{2}} \ldots e^{\frac{-iH_{\mathrm{env}}(s_0^+)\Delta t}{2}} \rho_B(0) e^{\frac{iH_{\mathrm{env}}(s_0^-)\Delta t}{2}} \ldots e^{\frac{iH_{\mathrm{env}}(s_{N-1}^-)\Delta t}{2}} e^{\frac{iH_{\mathrm{env}}(s')\Delta t}{2}} \Bigg],\end{aligned} \tag{1.59}$$

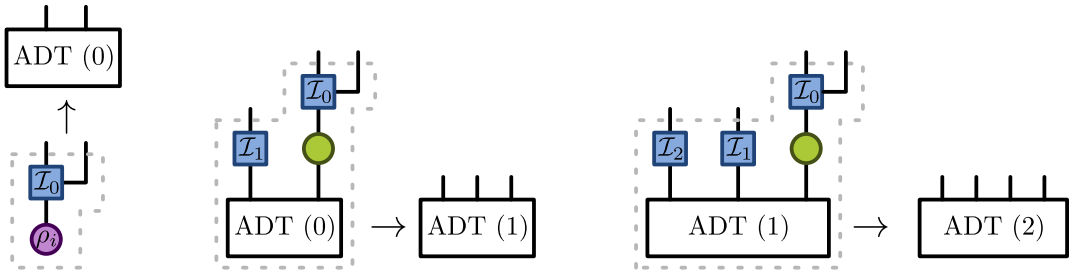
yields the effect of the environment on the system. Eq. 1.58 is found by inserting resolutions of the identity for the position coordinates at the infinitesimal time-steps s_i going forwards in time and backwards in time. It contains transition amplitudes of the system Hamiltonian $\left\langle s_{k+1}^+ \middle| e^{-iH_S\Delta t} \middle| s_k^+ \right\rangle$ forwards in time, as well as $\left\langle s_{k-1}^- \middle| e^{iH_S\Delta t} \middle| s_k^- \right\rangle$ backwards in time. After we trace over the environment, the influence from the environment in Eq. 1.58 appears in the influence functional. Since we work with a reservoir with an infinite number of degrees of freedom, we can send $\Delta t \to 0$ and allow $N \to \infty$. The calculation of this limit is the subject of reference [48]. The result is the Feynman-Vernon influence functional

$$\begin{aligned}I = \exp\Bigg[- \int_0^t dt' \int_0^{t'} dt'' \left(s^+(t') - s^-(t')\right) \left(C(t' - t'')s^+(t'') - C^*(t' - t'')s^-(t')\right) \\ - i \int_0^t dt' \int_0^\infty d\omega J(\omega) \left(s^+(t)^2 - s^-(t)^2\right) \Bigg]\end{aligned} \tag{1.60}$$

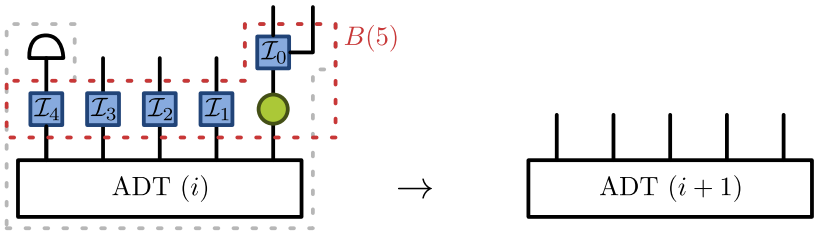
which is controlled by the *bath response* or *autocorrelation* function[10]

$$C(t) = \frac{1}{\pi} \int_0^\infty d\omega J(\omega) \left[\coth\left(\frac{\beta\omega}{2}\right) \cos(\omega t) - i \sin(\omega t) \right]. \tag{1.61}$$

[10] In writing this equation it is assumed that H_{SB} is the deformation potential 1.11, $\hat{O} \sum_q \left(g_q b_q + g_q^* b_q^\dagger\right)$.



(a) Growth phase



(b) Propagate phase

Figure 1.11: The QUAPI propagation scheme during (a) the grow phase, (b) the propagate phase assuming $k_{\max} = 5$.

The autocorrelation function is a measure of how the environment responds after a perturbation at a time t in the past. The autocorrelation function is controlled by the inverse temperature β and the spectral density,

$$J(\omega) = \sum_q |g_q|^2 \, \delta\left(\omega - \omega_q\right). \tag{1.62}$$

The influence functional then represents the reaction from the environment on the system after the perturbation [7]. Accordingly, if the system emits information from the system into the environment, the influence functional is the vehicle for which information is then restored back to the system.

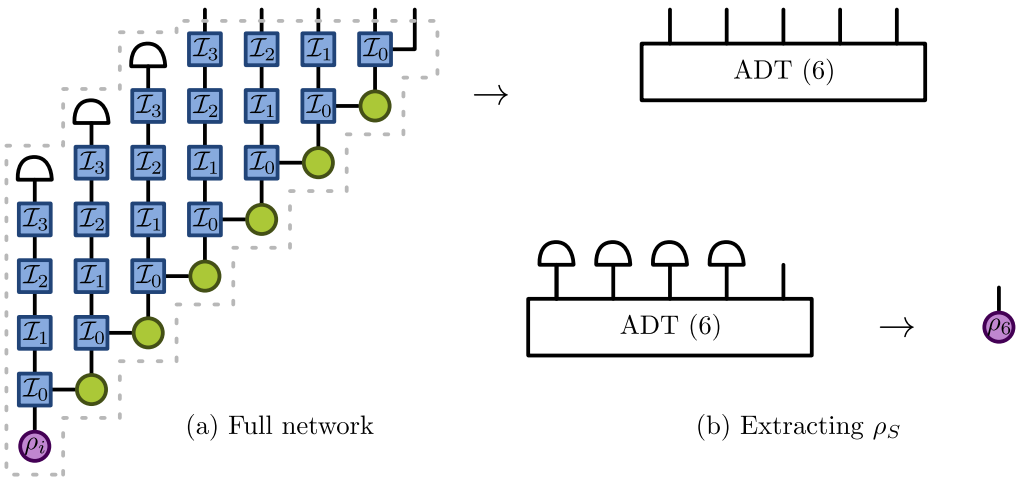
Equation 1.60 gives the influence functionals in continuous forms. In order to represent the dynamical map as a tensor-network, we re-discretise the influence functional into finite time-steps Δt. First we must introduce the Trotter splitting [51]. Here we wish to express the dynamical map as a time-discretised product of Liouvillians acting on vectorised density matrix. In this case the super-operator looks like $\mathcal{L}_S + \mathcal{L}_E$, where $\mathcal{L}_E = \mathcal{L}_B + \mathcal{L}_{SB}$. First order Trotterisation is to approximate this first term as

$$e^{(\mathcal{L}_S+\mathcal{L}_E)\Delta t} = e^{\mathcal{L}_S\Delta t}e^{\mathcal{L}_E\Delta t} + \mathcal{O}(\Delta t^2). \tag{1.63}$$

It would be prudent to introduce the higher order symmetrised Trotter splitting [49, 50, 52, 53],

$$e^{(\mathcal{L}_S+\mathcal{L}_E)\Delta t} = e^{\mathcal{L}_S\frac{\Delta t}{2}}e^{\mathcal{L}_E\Delta t}e^{\mathcal{L}_S\frac{\Delta t}{2}} + \mathcal{O}(\Delta t^3). \tag{1.64}$$

For clarity and consistency with the literature, we write down QUAPI with the first order Trotterisation, however since the numerical penalty in implementing the symmetrised Trotterisation is small compared to the overall gain, the TEMPO algorithm will use the latter, and so we will generally write down the TEMPO algorithm using the second order Trotterisation.



(a) Full network (b) Extracting ρ_S

Figure 1.12: Calculating the reduced density matrix after 6 time-steps with $k_{\max} = 3$. (a) The entire uncontracted tensor network (excluding the caps to extract the final density matrix) to obtain the 6th ADT. (b) Extracting ρ_S from the ADT

After dividing the influence functions up into N time-steps, the time evolved density matrix is found using

$$\rho(t) = \prod_{n=1}^{N-1} \prod_{k=0}^{n-1} I_k(n, n-k)\mathcal{U}(n)\rho(0), \tag{1.65}$$

where the system propagator is given by

$$\mathcal{U}(n) = e^{\mathcal{L}\Delta t}, \tag{1.66}$$

and the influence functions [54],

$$I_k(n, n-k) = -L^-_{O_S}(n\Delta t)\left(L^-_{O_S}((n-k)\Delta t)\Re(\eta_k) + iL^-_{O_S}((n-k)\Delta t)\Im(\eta_k)\right). \tag{1.67}$$

$$\eta_k = \begin{cases} \int_{k\Delta t}^{(k+1)\Delta t} dt' \int_0^{\Delta t} dt'' C(t'-t'') & k \neq 0 \\ \int_0^{\Delta t} dt' \int_0^{t'} dt'' C(t'-t'') & k = 0. \end{cases} \tag{1.68}$$

Here we have introduced a notation for a system super-operator,

$$L^{\pm}_{O_S} A = O_S A \pm A O_S, \tag{1.69}$$

that constructs the (anti)commutator.

In order to write Eq. 1.65 as a tensor network, we introduce two objects, the Augmented Density Tensor (ADT) and the object that iteratively propagates it. The state is now stored in the ADT, an extended state which includes the history of the system. The ADT is propagated by iteratively applying the tensor containing the influence functions (the B tensor in Ref. [52]). In order to express this as a tensor network, we rewrite the influence functional as a Liouvillian super-operator tensor

$$[\mathcal{I}_k]^{j'}_{j} = I(n, n-k)\delta^{j'}_{j} \quad k \neq 0. \tag{1.70}$$

where the orientation of the legs implies the direction of time (j' forwards in time and j backwards in time). The exception is

$$[\mathcal{I}_0]_j^{j'l} = \sum_{l'} I(n, n-k)\delta_j^{j'}\delta_{l'}^{l}, \tag{1.71}$$

as we need to add an extra leg in order to grow the ADT at each step and include the system unitary with the first influence functional in order to perform the Trotterisation. The QUAPI propagation scheme is now shown in figure 1.11. Fig. 1.11 (a) shows the growth phase iteratively. The ADT is iteratively contracted with the block of influence functions, where after each step the ADT gains an additional index (leg). The growth phase continues for the bath correlation time τ_C, however as previously mentioned, the history of the system longer than τ_C is assumed to be no longer relevant to the future evolution. Therefore influence functions larger than $k_{\max} = \frac{\tau_C}{\Delta t}$ will not affect the system density matrix. Therefore instead of adding influence functions greater than $k_{\max}$ we simply trace over the superfluous left most leg in Fig. 1.11 (c). This is the propagate phase where the size of the ADT and QUAPI propagator remain constant in size, of dimension $d_{\text{sys}}^{2k_{\max}}$ and $2d_{\text{sys}}^{2k_{\max}}$ respectively. Finally the density matrix at intermediate and final times can be found by summing over all of the influence function legs except the right most one, as shown in fig 1.12 (b).

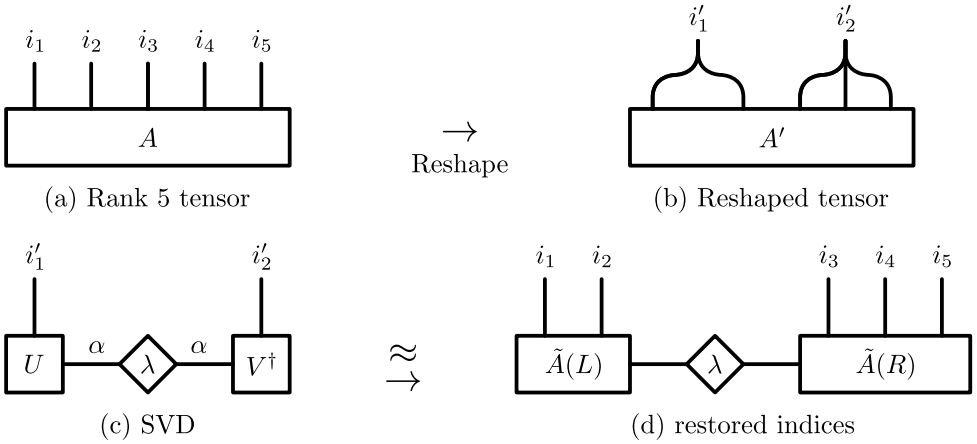
The QUAPI propagation scheme shown in fig 1.11 (a) reduces a non-Markovian propagation into an iterative series of tensor-tensor contractions. It bears some resemblance to the Markovian time-local tensor network in 1.7 (b), however instead of propagating a state, we instead propagate the ADT which is not a time local object, but includes the history of the system. If the response of the bath to a perturbation from the system decays quickly, as in, such that $C(t) \to \delta(t)$, the influence functions become a time local description, and we recover a Markov description and tensor network[11]. In the general case, it is assumed that the response of the environment decays over some finite time, which we have already referred to in our Markov derivation as τ_C. This time can be thought of the duration of the "time non-locality". The degree of time non-locality is intimately related to — but a distinct concept from — the degree of non-Markovianity, and controls the difficulty in computing the dynamics. The issue with the ADT is that the number of numbers in the ADT grows exponentially with τ_C.

1.2.5 Time-evolving Matrix-Product-Operators

In order to combat the exponential scaling of the QUAPI algorithm with respect to the bath correlation time τ_c, here we apply one of the pieces from the density matrix renormalisation toolkit: the matrix product state (MPS). An MPS is an efficient representation of a high rank tensor, assuming that the correlations are constrained in some way [52]. Strathearn et al. [52] realised that the QUAPI algorithm could be written in a way such that both the ADT, representing the state and the influence of the environment on the system, and the tensor that propagates the system forward, can be compressed using standard MPS algorithms.

In short the TEMPO algorithm takes the single rank $k_{\max}$ ADT and factorises it into a product of $k_{\max}$ tensors that are at most rank four tensors. The process for doing so is

[11] An example is described in Ref. [49], for an Ohmic spectral density $J(\omega) = \gamma\omega$ in the limit that $\beta \to 0$, $\Re[C(t)] \to \delta(t)$ and $\Im[C(t)] \to \delta'(t)$ and all influence functions do nothing except $\mathcal{I}_0$ which yields a pure dephasing model.



(a) Rank 5 tensor (b) Reshaped tensor

(c) SVD (d) restored indices

Figure 1.13: Expressing a rank five tensor as an MPS, diagram adapted from Ref [54]. (a) A sample rank 5 tensor with indices labelled. (b) The tensor after the indices have been grouped into yield a matrix. (c) Performing the SVD decomposition. (d) Approximating the matrix by discarding $\sigma_\alpha < \sigma_{\text{cutoff}}$.

shown in figure 1.13. First we re-write the object that takes us forward one time-step in Fig 1.11 (a) as a product of rank 4 tensors, where the individual tensors are the b_k tensors

$$[b_k]^{j'l}_{jl'} = \delta^{j'}_{j}\,\delta^{l}_{l'}\mathcal{I}(k) \tag{1.72}$$

The process for constructing the MPS is shown in 1.13. This involves grouping all of the legs into two to obtain a matrix and performing the singular value decomposition (SVD) on this matrix. The SVD of a matrix is given by

$$A^{ij} = U^{i\alpha}\lambda^{\alpha\alpha}\left[V^\dagger\right]^{\alpha j}, \tag{1.73}$$

where $\lambda^{\alpha\alpha}$ is a diagonal matrix whose diagonal elements are the singular values σ_α. These singular values are ordered from highest to lowest with increasing α. Figs. 1.13 show the construction of an MPS in diagram form. Figs. 1.13 (a) and (b) show the reshaping of the tensor, and Figs. (c) and (d) show the factorisation and compression of the MPS. The key point of MPS methods is that they are efficient when these singular values decay to negligible amounts at some point. This allows us to set a cut-off point, σ_{cutoff}, set every singular value $\sigma < \sigma_{\text{cutoff}}$ to zero. Since a portion of the matrices U and $V^\dagger$ are now multiplied by zeros, we are free to truncate the size of the index α. This then sets the size of the internal bond leg in the MPS. A sample of the full tensor network is shown in fig 1.15 (a).

We have now discovered all three precision parameters for the TEMPO algorithm. To recap, the first is the time-step of the discretisation of the influence functional Δt. The second is the time taken for the autocorrelation function to decay, τ_C, and its discrete analogue $k_{\max}$. These two parameters are both intrinsically related to the autocorrelation function Eq. 1.61 and QUAPI. The time-step must be coarse enough to resolve all of the features in the real and imaginary parts of the autocorrelation function [12]. The maximum correlation time must be long enough such that the autocorrelation function has well decayed so we can safely say that times longer than the cut-off time have no effect on

[12]From this one can imagine how this would cause issues with an environment that is nearly Markovian, as think of the time-step that is necessary to resolve something that is nearly a delta function

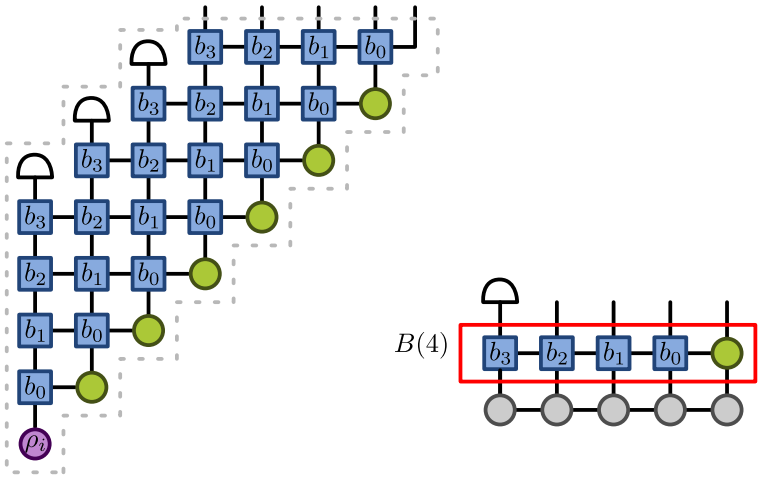


Figure 1.14: TEMPO propagation scheme. B operator given by Eq. 1.33.

future dynamics. Finally the SVD truncation parameter. This parameter controls the SVD truncation which prevents the TEMPO network from being exponentially difficult to represent. While there may be systematic ways to derive theoretical values that are required for specific scenarios [54, 55], in practice it is safer to perform variational analysis to obtain sufficiently accurate and converged results. This is true of all three parameters, however it is especially true of the SVD cut-off since there is no physical warning when results go astray, other than incorrect results.

1.2.6 The process tensor

The process tensor is a general way to represent a non-Markovian process by considering only the experimentally accessible quantities [8, 22]. It is a completely general framework that stores all possible trajectories for the system for a given set of control operations.

We shall first derive a process tensor in an MPO form by contracting the environment influence functions of the TEMPO algorithm. We will then spend some time outlining the more general properties of the process tensor, which we will later show is a structure that means we can efficiently obtain the gradient for optimisation.

Process tensor in MPO form

The process tensor is easiest understood from diagram 1.15. The original TEMPO algorithm simply contracted every tensor in the network together. This had to be done for any given set of system propagators $\{U\}$. For each system propagator there is approximately $k_{\max}$ influence tensors b_k. This indicates that the vast majority of the time is spent contracting the bath influence tensors. As outlined in reference [23, 24], one realises that the vast majority of the tensor network has no dependence on the initial state as well as the system propagators (and thus the control parameters for the system). Therefore one is free to pre-contract the influence functions b_k into a single MPO at each time-step (Figs. 1.15 (c–f)). If one were to do this as we just described, the bond dimension of the internal leg would grow exponentially. The key to preventing this, is after each contraction of a set of b_k tensors, an SVD sweep is performed on the states, except this time it is performed vertically rather than horizontally. This is performed iteratively until all of the influence

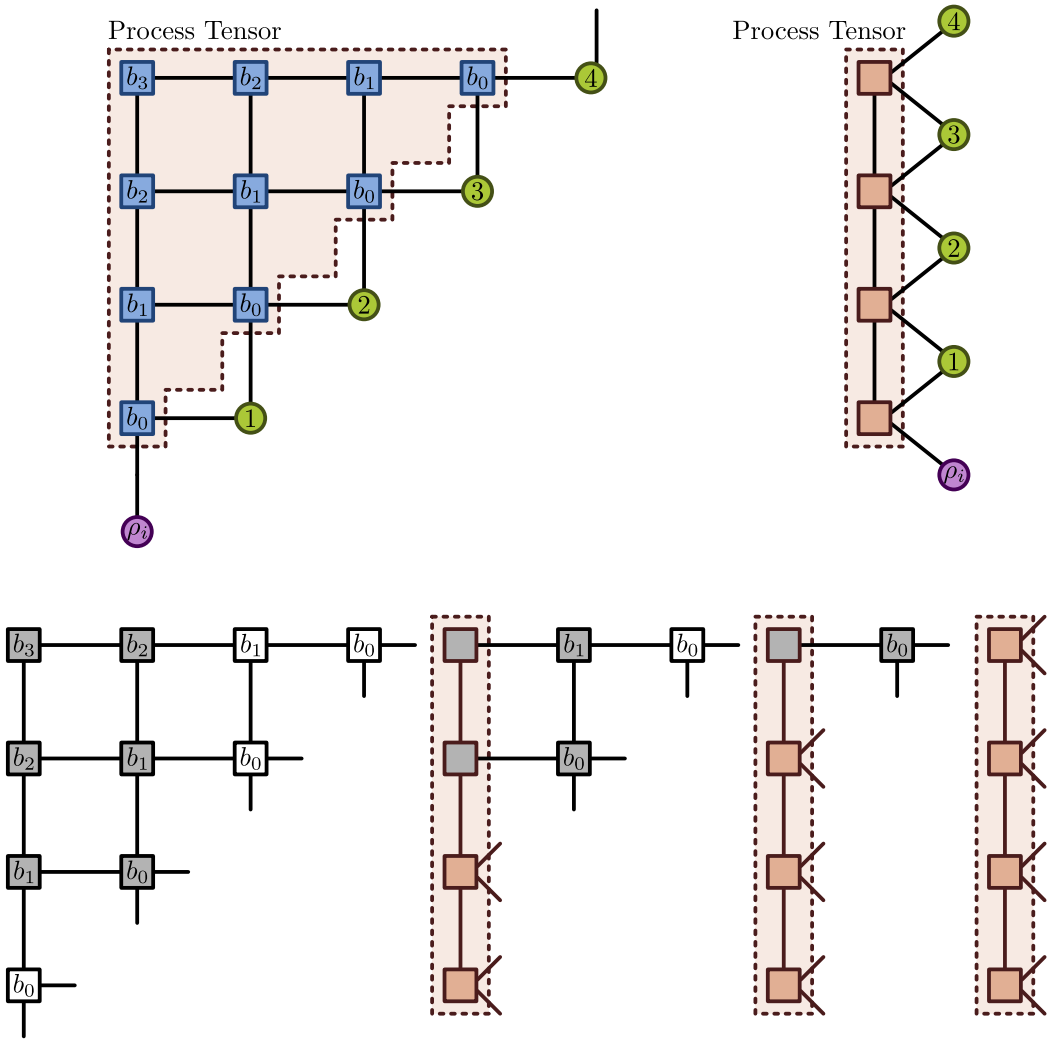


Figure 1.15: Constructing the process tensor: The TEMPO tensor network (a) with the environment influence functions b_k highlighted to show what will be contracted to become the process tensor (b). The contraction process is shown in Figs. (c-e), where the grey boxes indicate the tensors that will be contracted together. After each contraction, an SVD is performed on the area shown in the brown box to keep the internal bond dimension from increasing exponentially. Diagram adapted from Ref. [54].

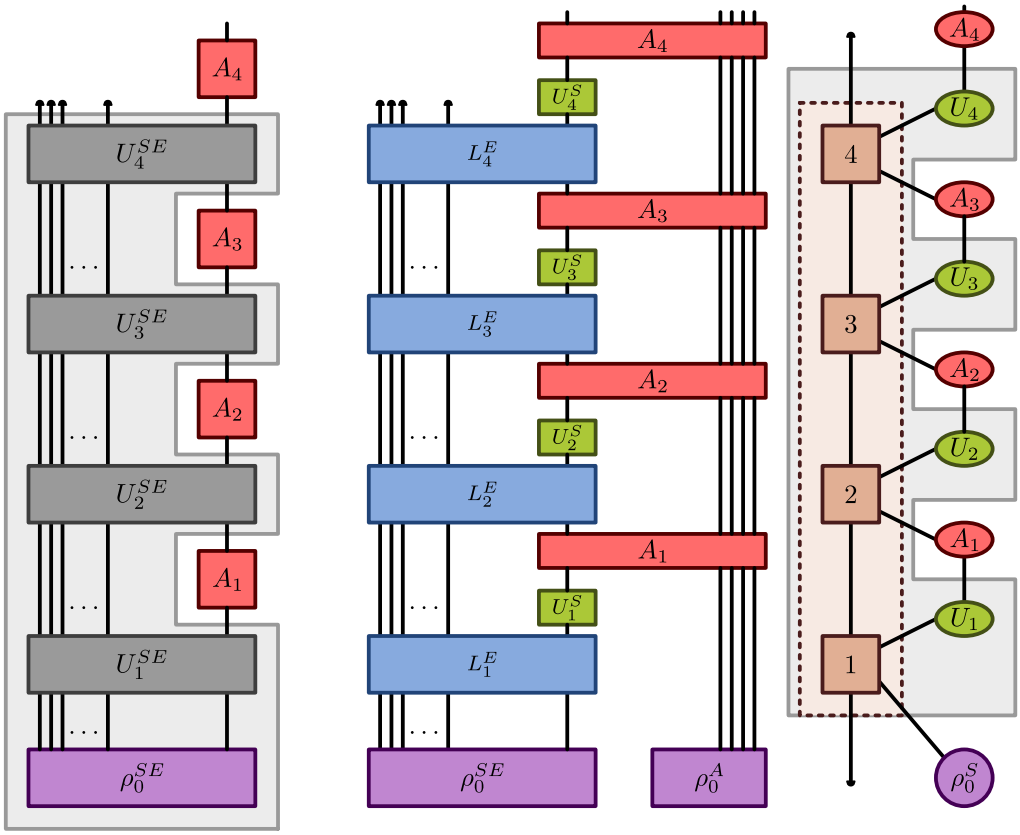
tensors b_k have been absorbed into a single vertical row of contracted influence tensors, connected to the system propagators 1.15 (b).

The PT in MPO form consists of the brown box in figure 1.15 (b), and is shown with none of the system propagators in (e). This object pertains to a specific environment, but as of now knows nothing about the system. We call the legs that connect between the process tensor MPOs (brown boxes), and the system propagators U (green circles) the system legs, which are of dimension d_{sys}^2. The internal legs that connect the MPO tensors vertically are the internal bond legs. Without any MPS compression they grow exponentially in size, roughly at the scale of d_{sys}^{2k} where k is the number of legs from the bottom, and plateauing at $d_{\text{sys}}^{2k_{\max}}$. As with the TEMPO algorithm, the SVD truncation makes this a representable number, and typical bond dimensions for this thesis are between 50 and 200. Nevertheless it is still significantly larger than the dimension of 4 for the system leg, so it is clear that having a process tensor that can be repeatedly used is desirable for use with multiple different systems. This is a significant advantage when considering a task such as optimisation of the system parameters, since for an optimisation problem one will need to evaluate some cost function (such as fidelity between a final state and a target state) for various different control parameters to maximise the cost function. To give an example, the SLM based optimisation of an ensemble of quantum dots in reference [24] took 11 hours to complete on an Intel i7 processor using the PT-MPO, whereas the conventional TEMPO algorithm would have taken over a month in this specific problem.

Process Tensor as a representation of a quantum stochastic process

So far we have described the PT in MPO form, shown in 1.15 (b). The process tensor is an altogether more general idea. It is at heart a universal object that describes a quantum process, the idea being that the description is based solely off the objects that one has access to in an experimental setup. The generic process tensor is shown in figure 1.16. An experimentalist has the ability to prepare an initial state ρ_i. At any time (at any time-step), the experimentalist can choose to perform an intervention on the system, denoted by $\{A\}$. The mathematical description of "an intervention" is to say that we can perform any mathematical operation described by a completely positive (CP) map. A CP map could represent a projective measurement, the application of a unitary on the system, or interaction with an ancilla system (shown in 1.16 (b)). The process tensor is entirely agnostic to the way for which it can be obtained, and in principle could even be obtained by tomography of an experimental setup [56]. We have provided the above description simply as an example of how one could construct a process tensor, and proposing a structure that yields an efficient representation. Without such preventative measures the process tensor would scale exponentially with the degree of non-locality. If the process tensor and the interventions act on a fixed basis the process tensor description reduces to a classical stochastic process, and is intimately related to the definition of a quantum stochastic process [22].

The process tensor is not a dynamical map in itself, but the dynamical map can be constructed from it once the system interventions are specified. Since the process tensor is described based on any intervention that can be implemented at any time-step, such interventions can be to measure the multi-time correlations. For the duration of this thesis, we will assume that the process tensor is in MPO form for clarity. This means we retain the properties of the PT described in reference [8], and our analysis can in principle be performed on any PT, for example that computed by tomography.



(a) Process tensor (b) PT with ancilla (c) PT-MPO

Figure 1.16: The general process tensor. (a) Process tensor with a set of controls $\{A\}$ (red boxes). PT denoted by light gray box. (b) The PT approximated using the first order Trotterisation, and where the controls $\{A\}$ are an interaction with a correlated ancilla. (c) Process tensor in MPO form. PT containing just MPOs is the brown/dashed box, and PT with the system propagators given by light gray box

Nomeclature of TEMPO and TEMPO derived methods

Now that we have explained our methods, we will now establish some conventions of TEMPO derived nomenclature. From now on when we refer to `OQuPy`, we refer to the open-source Python package [57] containing a set of implementations simulating open quantum systems using various different methods. We refer to "the TEMPO algorithm" as the algorithm proposed by Strathearn et al. [52] where we do not contract the bath influence functionals, and additionally, the SVD sweeps are performed horizontally (as opposed to vertically as in PT-MPO). We refer to "TEMPO related/derived methods" as any method derived from taking the ADT and using MPS methods to put the tensor in MPO form.

The process tensor is where things start to get confusing, and a bit of clarity is required as to what exactly is being referred to. We will generally refer to the process tensor as the MPOs in diagram 1.15, for the reason that it is a better working definition for the scope of this thesis. At the same time we acknowledge that the process tensor is generally understood to include the system propagators, and only the system intervention slots are left open [8, 22, 23], as denoted by the dotted box in Fig. 1.16. From now on we will clearly state when we are referring to the general definition of the process tensor, rather than just the object that is the MPOs from the contracted influence functions.

1.2.7 Meaning of non-Markovianity

In section 1.1.1, we discuss what we casually refer to as the Markov approximation. More precisely we outlined how the approximations of a the weak system-bath coupling (Born), short bath correlation time (Markov), as well as the rotating wave approximation (secular) leads to equations of motion that are time-local, and called them Markovian. We outlined criteria for the validity of the Markov approximation, albeit somewhat crudely. Now we come to possibly the most contentious point of this thesis, the meaning of the term "non-Markovianity". To a reader who is uninitiated in this debate, it would at first appear that it is sufficient to say that a system is non-Markovian if it fails the criteria stipulated in section 1.1.1. And we immediately would like to argue that this is ultimately what is important, a non-Markovain environment boils down to something that violates the Markovian criteria.

The issue arises when one attempts to obtain a metric of non-Markovianity, as generally, we want some form of sensible measure of how far away one is from something that can be described using a Lindblad equation – strongly implying a distance. The short answer is there is no one such metric. The idea of what constitutes as non-Markovianity is a topic of hot debate [9, 10, 34]. In particular, Ref. [34] builds a hierarchy of the implications of the various measures. Here we instead discuss non-Markovianity from the context of the process tensor. References [8, 22] argue that the introduction of the process tensor unifies all previous ideas of non-Markovianity and measures of non-Markovianity.

1.2.8 Measures of non-Markovianity

Breuer Measure

One measure is that introduced by Breuer, Laine and Piilo [58, 59]. It is a measure that is entirely derived from the dynamical map. This measure is based on the trace distance

$$D(\rho_1, \rho_2) = \frac{1}{2} \operatorname{Tr} \|\rho_1 - \rho_2\|, \tag{1.74}$$

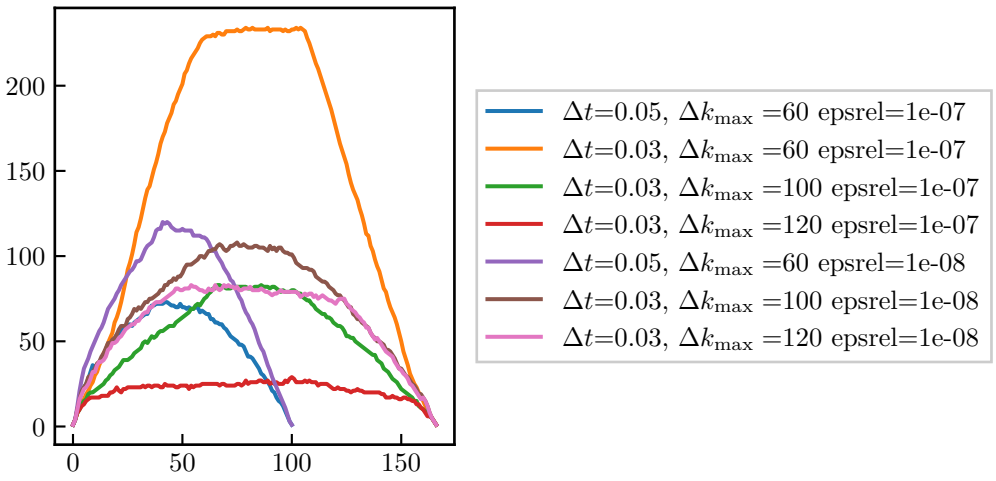


Figure 1.17: Bond dimension as a function of MPO index for the PT of a super-Ohmic spectral density.

where $\|A\|$ is the Hilbert-Schmidt norm $\|A\| = \sqrt{A^\dagger A}$. The key point being that the trace distance is interpreted as the distinguishably of quantum states [58], which is related to the information contained in those states. For a Markovian master equation with a Lindblad dissipator it can be shown that the trace distance of states corresponding to any fixed pair of initial states monotonically decreases as a function of time [58]. Reference [58] interprets the decrease in distinguishability of states as the loss of information to the environment This provides a neat and intuitive interpretation of the degree of non-Markovianity as the amount of increase in separation of states throughout the dynamics. Since the increase in separation is dependent on the choice of initial states, the increase in separation is calculated for all possible pairs of initial states on the Bloch sphere, and the measure is the maximum increase of separation found for a particular pair of states. We write this measure as [58, 59]

$$\mathcal{N} = \max_{\rho_1,\rho_2} \int_{\sigma>0} \sigma(t:\rho_1,\rho_2)dt$$
$$\sigma(t:\rho_1,\rho_2) = \frac{d}{dt}\,\mathrm{Tr}\left[\frac{\|\rho_1(t)-\rho_2(t)\|}{2}\right]. \tag{1.75}$$

Since this measure is based on the dynamics of the reduced density matrix, it is dependent on the choice of system Hamiltonian, and different Hamiltonians yield different non-Markovianity measures. This is in contrast to the other measure of non-Markovianity considered in this thesis which we discuss next.

Process Tensor as a measure of non-Markovianity

When introducing the process tensor formalism [8], Pollock et al. show in their accompanying letter [22] how the process tensor leads to a definition of non-Markovianity that the authors claim unifies all previously known measures of non-Markovianity.

The first step is writing the process tensor as a Choi state[13], Λ. This is a non-trivial

[13]On notation: since this thesis leans heavily on the conventions described in Wood et al. [39], we will

task for the general case for which we will not go into detail (since we can avoid doing so because we concern ourselves only with PTs that are written in the MPO form), but point to section 5.1.2 as to how to do that for a single Liouville super-operator. The reason we do this as the Choi state representation maps temporal correlations onto spatial ones [8]. Pollock et al. show how a Markov process is divisible, meaning that a Markovian Choi matrix can be written as a tensor product of CPTP maps

$$\Lambda_{k:0}^{\text{Markov}} = \mathcal{E}_{k:k-1} \otimes \ldots \mathcal{E}_{2:1} \otimes \mathcal{E}_{1:0}, \tag{1.76}$$

where the subscript is the indices for the time-steps the map spans over. The non-Markovianity measure $\mathcal{N}$ is given by

$$\mathcal{N} = \min_{\Lambda_{k:0}^{\text{Markov}}} \mathcal{D}\left[\Lambda_{k:0} \| \Lambda_{k:0}^{\text{Markov}}\right]. \tag{1.77}$$

In words, this means that the non-Markovianity measure is a CP-contractive quasi-distance $\mathcal{D}$ between the Choi matrix of a non-Markovian process and the closest Markov Choi state [22]. Reference [22] then state that if the Choi state is in a matrix product state then the meaningful measure is the size of the bond dimension of the MPS.

Here the "process tensor" refers to the object which includes the system propagators, however one is free to perform the same analysis without the system propagators, i.e. that of diagram 1.15 The way to go from the PT-MPO in the form of diagram 1.15 is to contract the system propagators into the PT while leaving the system intervention slots open, perform an SVD sweep, then read the bond dimensions. This leads to two measures of non-Markovianity, the bond dimension prior the introduction of the system propagators, shown in Fig. 1.17, and the bond dimension of the SVD including the system propagators. We did not examine the latter case numerically, however we believe that the quantity is going to be relatively similar to the former quantity. We suggest this because the PT can be thought of an object that knows all possible trajectories taken by a reduced density matrix for any given set of system propagators, and the numerical difficulty in representing this object is related to the number of possible trajectories that it can follow. Once the set of system propagators and interventions are specified, there are no free legs left in the tensor-network so the entire tensor-network can be reduced to a bond dimension of one (one possible trajectory). One can imagine how specifying the system propagators without specifying the external interventions can reduce the number of allowed trajectories, however there remains the issue that the PT still needs to account for the fact the external interventions can still be anything, thus not simplifying the problem significantly.

An issue with this measure of non-Markovianity is that the bond dimension in PT-MPO is entirely dependent on the TEMPO convergence parameters. Shown in figure 1.17 is the bond dimension for various control parameters, and one can clearly see that this measure of non-Markovianity is dependent on the choice of parameters, which should not be the case since the degree of non-Markovianity is an intrinsic property of the bath, not the parameters. The second problem with this measure is that it claims that the degree of non-Markovianity is a property of the environment, irrespective of the process. One can imagine a case where we consider an environment that according to the bond dimension is non-Markovian, but we now construct a Hamiltonian where the reduced equation of motion can be accurately described by a time-local equation of motion.

stick with this convention where a Choi matrix is expressed by Λ, as opposed to Υ in Pollock et al. [22], and the dynamical map is represented by $\mathcal{E}$, as opposed to Λ.

Comparing the two measures

The notable difference between the PT as a measure of non-Markovianity and the Breuer measure is the PT measure is not a measure based on the dynamical map of the reduced density matrix. The PT is based on an approach to represent a generic quantum stochastic process, and therefore fully accounting for any experimentally detectable phenomena, not just properties contained within the dynamical map.

So far we have not discussed the considerations for measuring a one-time probability versus measuring multi-time correlations. It has been extensively argued that the dynamical map does not encode all of the information necessary to observe anything that isn't a "prepare-evolve-measure" experiment, such as multi-time correlations [60]. Since the measure by Breuer, Laine and Piilo is entirely based on the dynamics of the reduced density matrix, it can fail to detect non-Markovian behaviour[14]. While this has been discussed thoroughly in the literature (see ref. [34], and the references therein for further information), it is not a discussion that is relevant to us, since in this thesis we only consider prepare-evolve-measure type applications. Therefore we conclude that within the scope of this work the Breuer measure is suitable for us. Ref. [22] explains how the PT as a measure of non-Markovianity can encapsulate demonstrable memory effects that other measures fail to account for. The PT also appears to excel as a method for comparing the degrees of non-Markovianity between different systems. However neither of these benefits are relevant in this thesis, since we only consider prepare-evolve-measure type experiments on a single system. Additionally, we do not know a CP-contractive quasi-distance that is not numerical parameter dependent as in Fig. 1.17.

In summary, we follow the conclusions in the introduction of [34] et al. and state that there is no one such idea of non-Markovianity and it is context specific. They agree with our initial assertion, that what is important is whether or not it is appropriate to to quote "the evolution of an open quantum system can be said to be Markovian or non-Markovian only when one has identified which of our conditions is relevant to the context of the work in question". We will use the measure by Breuer, Laine and Piilo to make claims about our optimisation exploiting the information back-flow from the environment back to the system, which can be seen as related to non-Markovianity. Ultimately this is a measure of information flowing from the environment to the system which we will later show that our optimiser is exploiting to improve the fidelity of state transfer.

1.3 Quantum Speed Limits

Here we take a slight diversion to learn about a fundamental part of quantum mechanics that we will use to assist our understanding of the fundamental limits, the Quantum Speed Limit (QSL). We follow the introduction of Ref. [61] and start by outlining the modern interpretation of the Heisenberg uncertainty — or, a more appropriately named indeterminacy principle — for recasting the energy time uncertainty bounds as an intrinsic bound on a timescale. We then outline what are the practical results of this and describe the generalisation to an open quantum system.

[14] An example of such is given in Fig. 2 of the supplement of Ref. [22].

1.3.1 What is time in quantum mechanics

In non-relativistic quantum mechanics, time is generally not an observable as one is unable to write it as a Hermitian operator. This poses an issue when we take a closer look at Heisenberg's uncertainty relation between energy and time.

Heisenberg's uncertainty principle

Heisenberg's uncertainty relation is the ubiquitously known statement that is at the forefront of our minds when we think about the abstract nature of quantum mechanics. Most physicists first encounter it in their first quantum mechanics course in terms of the momentum and position operators. The uncertainty of momentum and position is given by

$$\Delta x \Delta p \geq \frac{\hbar}{2}, \tag{1.78}$$

where we have that the variance of an operator A is given by

$$\Delta A = \sqrt{\langle A^2 \rangle - \langle A \rangle^2} \tag{1.79}$$

which under a standard textbook interpretation, says that the position and momentum of a quantum particle cannot be simultaneously determined with infinite precision [61]. Formally, the uncertainty principle arises from invoking the Cauchy-Schwarz inequality

$$\Delta A \Delta B \geq \frac{1}{2} |\langle [A, B] \rangle|, \tag{1.80}$$

and imposing canonical commutation relations,

$$[x, p] = i\hbar. \tag{1.81}$$

The correspondence principle holds as one can see that this uncertainty relation vanishes as $\hbar \to 0$. The second ubiquitous form of the uncertainty relationship is the energy-time uncertainty,

$$\Delta E \Delta t \gtrsim \hbar. \tag{1.82}$$

This relationship is familiar to undergraduate physics and is used across many areas of physics. This is despite the fact it has now become immediately clear that this cannot come from canonical commutation relations, since, as stated earlier, time cannot be written as an operator.

1.3.2 Finding the minimum time

The Mandelstam and Tamm bound

The Mandelstam and Tamm (MT) [62] bound solves this problem — in short — by saying that one should not think of an uncertainty relation, rather an intrinsic timescale for quantum evolution. Starting from the Liouville equation for the time evolution of an observable A,

$$\frac{\partial A}{\partial t} = \frac{i}{\hbar} [H, A], \tag{1.83}$$

and applying this to the Cauchy-Schwarz inequality, equation 1.80, yields

$$\Delta H \Delta A \geq \frac{\hbar}{2} \left| \left\langle \frac{\partial A}{\partial t} \right\rangle \right| \tag{1.84}$$

We now choose to consider the case where we are interested in an observable A that is a projector onto the initial state, ψ_i, so $A = |\psi_i\rangle\langle\psi_i|$. Because of this, we can write the variance of A as

$$\Delta A = \sqrt{\langle A^2\rangle - \langle A\rangle^2} = \sqrt{\langle A\rangle - \langle A\rangle^2}, \tag{1.85}$$

and $\langle A\rangle|_{t_i} = 1$. Integrating equation 1.83 gives

$$\frac{1}{\hbar}\Delta H t \geq \frac{\pi}{2} - \sin^{-1}\sqrt{\langle A\rangle|_t}. \tag{1.86}$$

We also choose to consider processes where the final state is orthogonal to the initial state, as will be the case for the majority of this thesis. In this case, $\langle\psi(0)|\psi(t_f)\rangle = 0$. This allows us to write the minimum time take for a quantum system to evolve between two orthogonal states as

$$\tau_{\text{QSL}} \equiv \frac{\pi}{2}\frac{\hbar}{\Delta H}. \tag{1.87}$$

The Margolus and Levitin bound

Despite the fact that the following issue will not crop up in this thesis, while introducing quantum speed limits, it would be prudent to note that ΔH can give a unreasonable measure for the speed of a quantum evolution [61, 63]. The Mandelstam and Tamm bound can be made arbitrarily small if the variance of the Hamiltonian can diverge while the average energy is finite [61, 64]. The Margolus and Levitin (ML) bound is an alternative derivation that does not suffer form this problem. The derivation starts by expanding the initial state $|\psi_0\rangle$ in the energy eigenbasis

$$|\psi(0)\rangle = \sum_n c_n |E_n\rangle. \tag{1.88}$$

The time evolved states according to the Schrödinger equation are

$$|\psi(t)\rangle = \sum_n c_n \exp\left[\frac{-iE_n t}{\hbar}\right]. \tag{1.89}$$

We can now calculate the overlap between the initial and time-evolved states

$$S(t) \equiv \langle\psi(0)|\psi(t)\rangle = \sum_n |c_n|^2 \exp\left[\frac{-iE_n t}{\hbar}\right]. \tag{1.90}$$

The quantum speed limit can be found by estimating the real part of $S(t)$ using the trigonometric inequality

$$\cos(x) \geq 1 - \frac{2}{\pi}(x + \sin(x)), \tag{1.91}$$

which is true as long as $\forall x \geq 0$, which applies here as long as we assume that the ground state energy $\langle H\rangle$ is non-negative. We can now write

$$\begin{aligned}
\Re(S) &= \sum_n |c_n|^2 \cos\left(\frac{E_n t}{\hbar}\right) \\
&\geq \sum_n |c_n|^2 \left[1 - \frac{2}{\pi}\left(\frac{E_n t}{\hbar} + \sin\left[\frac{E_n t}{\hbar}\right]\right)\right] \\
\Re(S) &\geq 1 - \frac{2}{\pi}\left(\frac{\langle H\rangle t}{\hbar} + \Im(S)\right).
\end{aligned} \tag{1.92}$$

Since we are considering the case where the initial and final states are orthogonal, $S(t_f) = 0$, and the real and imaginary parts vanish. By setting $\Re(S)$ and $\Im(S)$ to zero, we find that

$$t_f \geq t_{\mathrm{QSL}} \equiv \frac{\pi}{2}\frac{\hbar}{\langle H \rangle}. \tag{1.93}$$

It has been shown by Levitin and Toffoli [65] that the unified bound is tight,

$$t_{\mathrm{QSL}} = \max\left\{\frac{\pi}{2}\frac{\hbar}{\Delta H}, \frac{\pi}{2}\frac{\hbar}{\langle H \rangle}\right\}, \tag{1.94}$$

and we refer the reader to the literature [61, 65].

Non-Markovian speed limits

So far we have dealt with the case where we are considering the QSL between two orthogonal states evolving under a time-independent Hamiltonian. Here we outline the quantum speed limit for a system under the influence of a non-Markovian environment. Our starting point will be a generic quantum master equation

$$\dot{\rho}(t) = \mathcal{L}\rho(t), \tag{1.95}$$

where the time evolution of a state ρ is given by integrating a Liouvillian super-operator $\mathcal{L}$. In order to arrive at the QSL for an open system, rather than using the minimal time approach, we will instead use the perspective of considering the maximal speed of a quantum evolution.

Here, we will skip the rigour and present the ideas and the results, and refer the reader to the literature [61] for further information. Our starting point is the following question: what is the shortest path between two quantum states $|\psi_1\rangle$ and $|\psi_2\rangle$. Wootters [66] shows this to be

$$\ell(|\psi_1\rangle, |\psi_2\rangle) = \cos^{-1}[|\langle\psi_1|\psi_2\rangle|]. \tag{1.96}$$

and concludes that the shortest path connecting vectors in Hilbert space, which is the angle between these vectors, is the geodesic under the Fisher-Rao metric. Now that we have the shortest distance, the shortest speed is the rate of change of the distance. The geometric quantum speed limit is therefore the upper bound of the speed,

$$v \equiv \dot{\ell}(|\psi(t_i)\rangle, |\psi(t_f)\rangle) \leq v_{\mathrm{QSL}}, \tag{1.97}$$

and the bound on the minimum time is the reciprocal of the average speed [67–70]

$$t_{\mathrm{QSL}} = \frac{t}{\int_0^t v_{\mathrm{QSL}} dt}. \tag{1.98}$$

So far we have only considered pure states. For mixed states the appropriate distance metric is the Bures angle [41, 71, 72],

$$\mathcal{B}(\rho_1, \rho_2) = \cos^{-1}(\sqrt{\mathcal{F}(\rho_1, \rho_2)}), \tag{1.99}$$

where we also define the quantum fidelity

$$\mathcal{F}(\rho_1, \rho_2) = \left(\mathrm{Tr}\left[\sqrt{\sqrt{\rho_1}\rho_2\sqrt{\rho_1}}\right]\right) \tag{1.100}$$

which, despite first appearance, is invariant under the exchange of ρ_1 and ρ_2. Here we state that in this thesis we deviate from the standard convention of naming the Bures distance of "$\mathcal{L}$", as we use this symbol to denote the Liouvillian super-operator. The Bures angle is the Fisher-Rao distance maximised over all possible purifications [61]. From these starting points Taddei et al. [73] obtain a metric inspired by differential geometry by considering the quantum Fisher information. Del Campo et al. apply the Cauchy-Schwarz inequality to derive a ML type bound for a non-Markovian quantum channel.

We will focus on the result of Deffner and Lutz [74], who derive a MT and an ML type bound for non-Markovian systems using a geometric approach and the von Neumann trace equality to obtain an ML type bound, and the Cauchy-Schwarz inequality to obtain a MT type bound. The maximal quantum speed is given by

$$v_{\mathrm{QSL}} = \frac{\|\mathcal{L}\rho(t)\|_\bullet}{2\cos(\mathcal{B})\sin(\mathcal{B})}, \tag{1.101}$$

and the minimum time required to evolve between two states

$$t_{\mathrm{QSL}} \equiv T_D = \frac{(\sin[\mathcal{B}(\rho_i,\rho_f)])^2}{\frac{1}{t}\int_0^t d\tau \|\mathcal{L}\rho(\tau)\|_\bullet}, \tag{1.102}$$

where $\|A\|_\bullet$ corresponds to either the Hilbert-Schmidt[15] norm, $\|A\|_{\mathrm{hs}}$, the trace norm[16] $\|A\|_{\mathrm{tr}}$, or finally the the operator norm[17], $\|A\|_{\mathrm{op}}$, norm, In the case of unitary dynamics between pure states, equation 1.102 reduces to the MT bound (Eq. 1.87) when one uses the Hilbert-Schmidt norm, and the ML bound (Eq. 1.93) when one uses the trace norm, as $\|H(t)\rho(t)\|_{\mathrm{tr}} = \mathrm{Tr}[|H(t)\rho(t)] = \langle H(t)\rangle$ if the instantaneous eigenvalues of $H(t)$ are all positive [74]. We can go further and say that the operator norm is smaller than the trace norm (which is obvious from the definition), and the trace norm is always smaller than the Hilbert-Schmidt norm. Therefore, we find that the operator norm yields the sharpest bound of the problem. There are other bounds as well, such as that of Pires at al. [75] which involves constructing an infinite family of bounds. This, however, comes with the issue that the properties of the QSLs strongly depend on the choice of distance and the task of finding a distance that yields a tight bound is unresolved [76]. These bounds also suffer from the issue that the quantum speed limit is defined based on the trajectory of the dynamics, $\sim \int(\|\dot{\rho(t)}dt\|_\bullet)$. The bounds by Sun et al. [77], Del Campo et al. [78] have been shown to be looser than equation 1.102 [79] .

Deffner and Lutz show by example that this bound is both tight, and achievable, however this is not always true [75, 76, 80, 81]. Campaioli et al. [76, 79] provide bounds that have been shown to be tighter than equation 1.102. They argue that the looseness of Eq. 1.102 is due to the Bures angle not selecting the geodesic correctly [76]. The bound by Campaioli et al. is instead expressed in terms of the generalised Bloch vector [82]. Ref. [79] gives the generalised Bloch vector for general systems, but for two level systems it reduces to the

[15]The Hilbert-Schmidt norm, which for finite dimensional Hilbert space is identical to the Frobenius norm, is given by the square root of the sum of the squares of the singular values.

$$\|A\|_{\mathrm{hs}} = \sqrt{\sum_i \sigma_i^2} = \sqrt{\mathrm{Tr}[A^\dagger A]},$$

where σ_i are the singular values of A [74]

[16]The trace norm for a Hermitian operator is the sum of the singular values, $\|A\|_{\mathrm{tr}} = \sum_i \sigma_i$

[17]The operator norm for a Hermitian operator is the largest singular value, $\|A\|_{\mathrm{op}} = \sigma_1$

usual Bloch vector $\boldsymbol{r} = r_x\sigma_x + r_y\sigma_y + r_z\sigma_z$. The QSL between the initial Bloch vector $\boldsymbol{r}_i$ and $\boldsymbol{r}_f$ is

$$t_{\mathrm{QSL}} \equiv T_C = \frac{\|\boldsymbol{r}_i - \boldsymbol{r}_f\|}{\frac{1}{t}\int_0^t d\tau \|\mathcal{L}\rho(\tau)\|_{\mathrm{hs}}}. \tag{1.103}$$

We will later compare τ_D and τ_C in our system of interest.

1.4 Physical Models

So far we have described a general framework for simulating open quantum systems. For the sake of readability we have so far restricted ourselves to a bath of bosonic operators, but this need not be the case. For Markovian systems, equations of motion have been derived for fermionic and multiple environments [83]. As previously discussed, our methods that use the process tensor are agnostic to how the process tensor is actually obtained, all we have assumed is that it can be efficiently represented in an MPO form. PT-MPOs can be derived for fermionic environments [84], multiple environments [85, 86], and chains of open quantum systems [87].

Therefore, since this thesis is based on simulations using the `OQuPy` package, we can safely say these results can be reproduced with most quantum systems of interest.

1.4.1 Specifying the environment

As described in the introduction, we construct the process tensors by contracting the environment influence functionals. The influence functionals are constructed from integrating the environment autocorrelation functions 1.61

$$C(t) = \frac{1}{\pi}\int_0^\omega d\omega J(\omega)\left[\coth\left(\frac{\beta\omega}{2}\right)\cos(\omega t) - i\sin(\omega t)\right].$$

Here the only thing we need to specify is an inverse temperature β and an environment spectral density

$$J(\omega) = \sum_q |g_q|^2 \delta\left(\omega - \omega_q\right).$$

Since we are interested in performing quantum operations, for simplicity we have targeted physical systems that can be treated as two-level systems. It should be mentioned that all of our methods scale exponentially with the size of the system Hilbert space.

1.4.2 Semiconductor quantum dots

The system we consider in this thesis a semiconductor quantum dot (QD). Here we consider semiconductor QDs that are zero-dimensional heterostructures, so they are nanoscale islands of a semiconductor on a different substrate. These QDs are nanoscale in all three directions, and formed of a material that has a lower band gap than the substrate. The QD can be thought of as a finite three dimensional quantum well. Instead of the energy levels of the QD forming bands such as that seen in the substrate, we instead observe discrete energy levels.

Here we consider InGaAs quantum dots in a GaAs matrix. The highest quality dots are grown by molecular beam epitaxy, depositing the QD one layer at a time [32]. These dots are controlled by optical excitation which produces an exciton within the QD. Excitons

also form in bulk semiconductors, however usually their lifetimes are shorter as their binding energies are low and they disassociate thermally [88]. In InGaAs, the potential well of the dot is typically 100-300 meV deep for the electron and 30-100 meV deep for the hole [32], so the exciton is spatially bound by the potential well more so than by Coulomb attraction. The statistical nature of the manufacturing process means that each dot will vary in shape, size, composition, and accordingly, optical properties. Therefore when making an ensemble of quantum dots, the optical properties will most likely be different. There are various experimental techniques to isolate quantum dots, such as apertures, masks and lenses [89], or making the dot density lower[18]. Because the energy levels are somewhat tunable [91], QDs have been found to be very useful for many applications such as quantum dot lasers [92], single photon sources [93–95] and qubit registrars [96–99].

Under the correct experimental conditions, that is a laser with a narrow spectral width relative to the energy separation with neighbouring transitions, the QD can be controlled such that it can be approximated as a two level system [32, 91, 100, 101]. Ramsay et al. [102] show how the dominant form of dephasing for an optically excited exciton is acoustic phonons, and outline a suitable spectral density. The spectral density for this model takes the form [103] ($\hbar$ included for consistency with literature)

$$J(\omega) = \frac{\hbar A}{\pi k_B}\omega^3 \exp\left[-\frac{\omega^2}{\omega_c^2}\right]. \tag{1.104}$$

This is a super-Ohmic spectral density with a coupling constant strength $2\alpha = \frac{\hbar A}{\pi k_B}$. The fact that it is super-Ohmic arises because the phonons can be approximated using a linear dispersion relation in a three dimensional bulk semiconductor (see Ref. [104] for further details). The Gaussian form of the high frequency cut-off arises from the spatial confinement of the electron and hole in the QD [32, 104]. References [103, 105] give experimentally fitted values for parameters to give a spectral density that is representative of a QD. Reference [37] derives a Lindblad equation for a QD.

[18] See references [32, 89, 90] for a review on the optical structure of QDs.

Chapter 2

Gradients using process tensors

Here we detail how the PT in MPO form leads to an efficient method to obtain the gradient of an objective function for optimal control.

2.1 Introduction to optimal control

It is a fundamental desire to know how to proceed when presented with a range of options to achieve a certain goal. Humans have always attempted to apply intuition to identify the optimal method for solving a problem. Thus optimal control is a field of mathematics that humans are intimately familiar with on a day to day basis. It sees widespread use throughout engineering ans science, seeking to identify the best solution to achieve a problem. Here we briefly outline the framework of optimal control and state how it can be significantly aided by including the gradient.

In order to understand optimal control, we ask what are we trying to achieve. All choices have a fundamental goal in mind, such as driving a car from coordinates a to coordinates b, designing the shape of antennas to receive a radio signal, or identifying control sequences for quantum operations. With each question, we can also ask "how well have we achieved our goal"? In the case of driving between coordinates, two questions that could be answered are have we arrived at our intended coordinates, and if so, how long did it take to get there. Both of these answers introduce a function of merit, or *objective function*. One such objective function for the former is the distance between our final coordinates and target coordinates, the latter being the time taken to get from our initial to final coordinates. We also introduce the control parameters. Numerical optimisation varies the control parameters seeking to extremise some objective function F.

2.1.1 Extremising an objective function

There are various different categories of numerical optimisation. We will make this introduction brief and restricted to the cases that we are interested in for this thesis, and refer the reader to the many suitable textbooks [106] for further information.

To start off, we give a definition of optimisation as minimising or maximising some function of merit or objective function F. It is assumed that we have control over some set of control parameters $\{c_a\}$, such that $F(c_1, c_2 \dots c_N)$, and we say that this problem is an N dimensional optimisation problem. Often, an optimisation is subject to some constraints, which are scalar functions of the control parameters $\{c_a\}$ that define certain equalities or

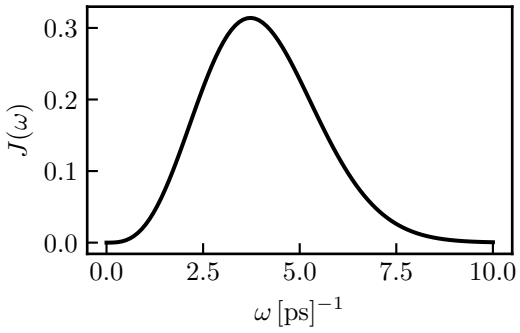


Figure 2.1: Super-Ohmic spectral density used for the results in this chapter 2.4.

inequalities that the control parameters must satisfy. In this case, the problem is now cast as

$$\min[F(c_1, c_2 \ldots c_N)] \qquad \text{subject to:} \quad x(c_a) = 0 \tag{2.1}$$

$$y(c_a) \leq 0 \tag{2.2}$$

where x and y are some constraint functions. In our case there will be bounds on the range of allowed parameters $\{c_a\}$. The next goal is to classify the extrema obtained by our optimisation. As in elementary calculus, we classify them as local extrema or global extrema. In the absence of knowing the analytical fitness function or particular information about the form of the fitness function, it is not possible to know if we have found the global extreme points. Optimisation can be categorised as local optimisation: a procedure that finds the nearest optimal solution where the gradient is zero and then terminates, as opposed to a global optimisation which will search for many local minima across a wide variety of control parameters and attempt to find the global minimum.

2.2 Physical Model

In this chapter and the following chapter we consider an spin-boson model which is modelled by

$$H = H_S(t) + \sum_q^\infty \omega_q b_q^\dagger b_q + S_z \sum_q^\infty \left(g_q b_q^\dagger + g_q^* b_q \right), \tag{2.3}$$

As described in the introduction, this describes a two level system with pseudo-spin operators $S_\bullet$. For these two chapters we choose a particular model which is representative of the phonon decoherence in an InGaAs quantum dot, with the super-Ohmic spectral density [37, 103, 105]

$$J(\omega) = 2\alpha \frac{\omega^3}{\omega_c^2} \exp\left[-\frac{\omega^2}{\omega_c^2} \right], \tag{2.4}$$

with a high frequency cut-off $\omega_c = 3.04$ and a dimensionless coupling constant of $\alpha = 0.126$. We choose the coupling between the system and the environment to be along S_z.

2.2.1 Parametrising the system Hamiltonian

Our general parametrisation for a time-dependent system Hamiltonian is

$$H_S(t) = h_x(t) S_x + h_y(t) S_y + h_z(t) S_z, \tag{2.5}$$

where $h_\bullet$ are the controls. Since we work with discretised time, we treat the controls as piecewise constant over whole time steps. To denote when dealing with discrete time we will instead express $H(t_n)$ where $t_n = n\Delta t$. For this chapter we will focus on controlling h_x and h_z while setting h_y to zero. This is in analogy to an optical transition of a quantum dot driven by a laser pulse, in which case h_z and h_x are the detuning and Rabi frequency respectively. In this case our control parameters c_a are $h_x(t_1), h_x(t_2) \dots h_x(t_{N_t})$ and $h_z(t_1), h_z(t_2) \dots h_z(t_{N_t})$. This gives $N = 2N_t$ control parameters.

2.2.2 State transfer

In this chapter we we restrict ourselves to state transfer. State transfer means that we assume that we know precisely what are initial state is, and that our goal is to seek control parameters that will steer us to our target state. We assume (though this need not be the case) that the initial state is a pure state, $|\psi_i\rangle$, with its associated density matrix $\rho_i = |\psi_i\rangle\langle\psi_i|$ The goal is to find protocols that will steer the dot from a given initial state, ρ_i, to a desired target state ρ_t. Next we define the objective function F. One suitable objective function is the quantum fidelity [35]

$$F(\rho_f, \rho_t) = \mathcal{F}(\rho_f, \rho_t) = \left(\mathrm{Tr}\sqrt{\sqrt{\rho_f}\rho_t\sqrt{\rho_f}}\right)^2. \tag{2.6}$$

An optimal protocol is found by numerically maximizing F over N control parameters which determine the system propagators at each discrete time step. Such numerical optimization becomes significantly faster if one can efficiently calculate the gradient of F with respect to the N-dimensional vector of control parameters, ∇F. A naive calculation of ∇F requires of order $2N$ evaluations of F, strongly limiting the size of problems that can be treated. Such gradients can however be efficiently obtained using the adjoint method [21], which has been applied to unitary or Markovian dynamics [20, 107].

2.3 Adjoint Method in PT-MPO

In order to obtain an efficient representation of the gradient we explain how we can exploit the fact that the PT in MPO form is a multi-linear map to obtain the gradient with respect to all control parameters. We will introduce the algorithm using a simplification where we state that instead of considering some arbitrary target density matrix ρ_t, we instead consider a pure target state $|\psi_\sigma\rangle$, and defer a description to how this can be generalised to the fidelity with respect to any target density matrix until later. The reason this simplifies the problem is because if the target state is a pure state the fidelity between a final density matrix ρ_f and a target density matrix $\sigma = |\psi_\sigma\rangle\langle\psi_\sigma|$ is now

$$\mathcal{F}(\rho_f, \sigma) = \mathrm{Tr}[\rho_f\sigma] = \langle\psi_\sigma|\rho_f|\psi_\sigma\rangle\,. \tag{2.7}$$

The interesting thing is when we vectorise the density matrix into Liouville space, performing the trace in equation 2.7 in index notation is $\mathcal{F} = (\sigma^T)^i\rho_f^i$. The key point being that we can compute the fidelity by taking the final state computed by contracting the process tensor, and contract that object with the *transposed target state* to obtain the fidelity. The network for the fidelity (a scalar with zero free legs) is shown in Fig. 2.2 (a).

Our goal is to calculate the derivative of the fidelity with respect to some control parameter, i.e. $\frac{\partial\mathcal{F}}{\partial c_a}$ for every c_a, $\nabla\mathcal{F}$. In order to obtain an efficient method for doing this, we use

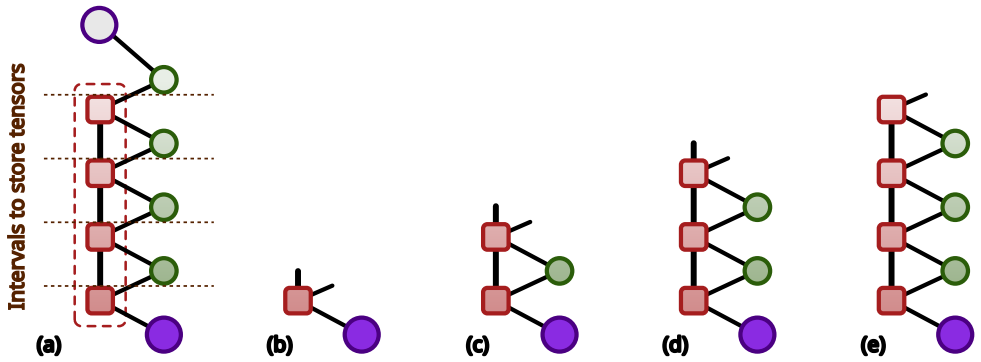


Figure 2.2: Illustration of the forward propagation calculation for a process tensor with four time steps. The left-hand diagram is the scalar product of the time-evolved state with the derivative of the objective function with respect to that state (top purple circle). In forward propagation the initial state is evolved forwards in time, to successively compute the diagrams shown in panels (b–e). To compute the gradient these tensors are stored.

the chain rule to split this derivative into the derivative of the fidelity with respect to the rank two system propagator U^{ij},

$$\frac{\partial \mathcal{F}}{\partial c_a} = \sum_n \frac{\partial \mathcal{F}}{\partial U_n^{ij}} \frac{\partial U_n^{ij}}{\partial c_a}, \tag{2.8}$$

where n labels the n^{th} time-step system propagator. We can simplify this equation based on the parametrisation of the system Hamiltonian we have just defined. Because $h_\bullet(t_1), h_\bullet(t_2) \dots h_\bullet(t_{N_t})$ are our control parameters, each control parameter only affects a single system propagator, so $\frac{\partial U_n}{\partial h_\bullet(t_k)}$ is only non-zero for $n = k$, and we can drop the sum over n in Eq. 2.8. Eq. 2.8 now reads

$$\frac{\partial \mathcal{F}}{\partial c_a} = \frac{\partial \mathcal{F}}{\partial U_k^{ij}} \frac{\partial U_k^{ij}}{\partial c_a}. \tag{2.9}$$

We have now recast this calculation in terms of two terms. The first term $\frac{\partial U^{ij}}{\partial c_a}$, is just the derivative of a $(d_{\text{sys}}^2, d_{\text{sys}}^2)$ matrix with respect to a scalar, so can be obtained via finite differences, or alternatively an automatic differentiation routine. Secondly $\frac{\partial \mathcal{F}}{\partial U^{ij}}$ which we get from the process tensor. The easiest way to introduce this is to recall the calculation of the time evolution from the initial state to the final state, shown in Fig. 2.2 (a), shown with an example of a PT with four time-steps. Here we note these diagrams are drawn for a first order Trotter splitting and we defer an explanation of the considerations required for a second order Trotter splitting to the appendix. The algorithm is easiest understood by stating the desired result. Due to the property that the PT is a *multilinear* map from the initial state to the target state (and thus the fidelity), $\frac{\partial \mathcal{F}}{\partial U_n}$ is just the network for the fidelity (diagram 2.2 (a)) except with the n^{th} system propagator removed from the diagram. This leaves two system legs open giving a $(d_{\text{sys}}^2, d_{\text{sys}}^2)$ matrix, as expected. Since we are trying to obtain the derivative with respect to every control parameter and due to the parametrisation in Eq. 2.5, we must obtain the derivative of the fidelity with respect to every system propagator.

The trick to obtain all of the derivatives efficiently is to perform a "forward propagation" forwards in time starting from the initial state, followed by a propagation backwards

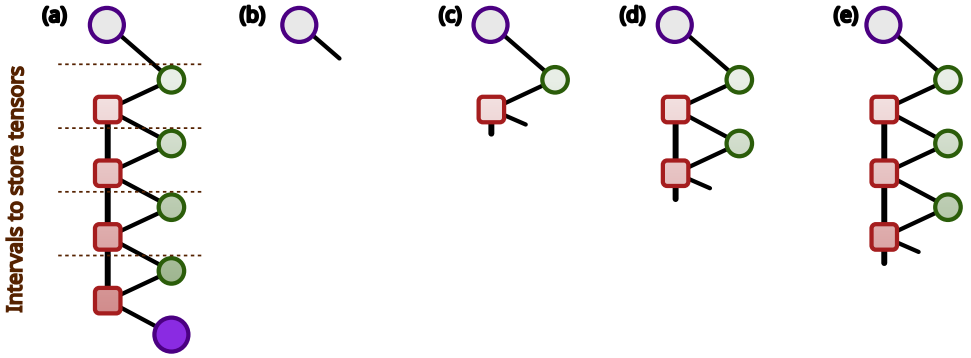


Figure 2.3: Illustration of the back propagation calculation for a process tensor with four time steps. The left-hand diagram is the scalar product of the time-evolved state with the derivative of the objective function with respect to that state (top purple circle). In back propagation this tensor is evolved backwards in time, to successively compute the diagrams shown in panels (b–e). The gradient is obtained by combining these tensors with those stored during the forward propagation.

in time starting from the transposed target state, while storing the required intermediate states. Stitching these intermediate states together will obtain the derivatives with respect to all system propagators U_n. In detail, we start from the initial state (Fig. 2.2 (b)) and propagate forward in time, while storing the rank two tensors (Figs. 2.2 (c–e), and the first step is an exception since it is a rank one tensor (b)) at the appropriate time-steps (indicated in the full diagram 2.2 (a)). Here we note that the crucial difference between this method and the adjoint method for Markovian or unitary dynamics is that instead of propagating the initial state, we are propagating rank two tensors, one leg being the state leg of dimension d_{sys}^2 which will connect to the system propagators, and the internal leg which connects to the next MPO. This internal bond leg harbours all the information about the system environment interaction history and allows us to obtain the non-Markovian dynamics.

The back-propagation involves much the same steps, except this time starting with the transposed target state and propagating the rank two tensor backwards, while storing at the appropriate intervals (Figs. 2.3 (b–e)). In order to obtain the $(d_{\text{sys}}^2, d_{\text{sys}}^2)$ tensors $\frac{\partial \mathcal{F}}{\partial U_n}$, we combine the saved forward propagated states and back propagated states. The first forward-propagated state (Fig. 2.2 (b)) is then combined with the last back-propagated state (Fig. 2.3 (e)) to give Fig. 2.4 (a). Next the second forward-propagated state is combined with the second last propagated state to give Fig. 2.4 (b), and so on. In this way we have discovered N_t rank two tensors which are the fidelity diagram with a single system propagator removed, which is what we set out to achieve. Finally the fidelity is obtained by combining the two terms in the chain rule, Eq. 2.9. Diagrammatically this is done by inserting $\frac{\partial U_n^{ij}}{\partial c_a}$ into the diagram at times n (shown as the yellow circle in Fig 2.4). This reduces to the scalar fidelity.

The advantage of this method is it obtains the full gradient $\nabla \mathcal{F}$, for minimal extra effort relative to computing the objective function in the first place. This is because, the forward-propagation is already required in order to compute the fidelity in the first place. Storing the tensors at the intervals simply increases the amount of memory that is required for this calculation. The back-propagation is an additional requirement, which is equally diffi-

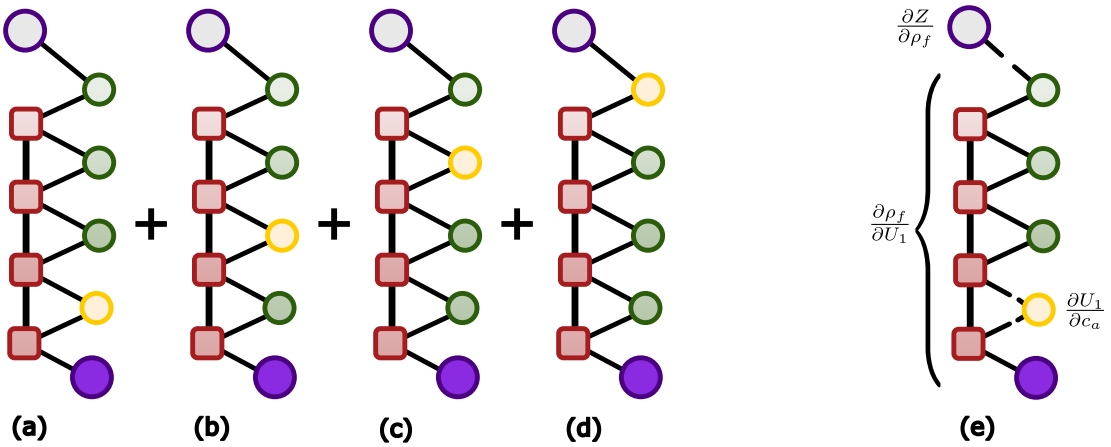


Figure 2.4: Diagrammatic representation of the derivative of an objective function with respect to a control parameter, Eq. (2.10). (a) is the contribution involving the propagator over the first time step, U_1, and (b–d) the contributions from subsequent steps. The yellow circle in each panel is the derivative of the propagator over that time step with respect to the control parameter. The purple circle at the top of each panel is the derivative of the objective function with respect to the final state. (e) shows how (a) is constructed by joining legs (contracting indices) among tensors corresponding to the factors in Eq. (2.10).

cult as the forward-propagation. Stitching the forward-propagation and back-propagation together involves summing over a single internal bond leg. Since this must be performed for each time-step, as well as adding in $\frac{\partial U_n^{ij}}{\partial c_a}$, this amounts to computational time that is roughly equivalent to a single propagation. As a result, we can obtain both the fidelity and the gradient with respect to the control parameters with a computational cost that is equivalent to performing three forward-propagations, irrespective of the number of control parameters. This is not significantly advantageous if you only have a couple of parameters, however in the parametrisation we have described, we have $2N_t$ control parameters, which practically speaking will mean anywhere between 100–300 control parameters. Performing three propagations to obtain the $\nabla\mathcal{F}$ is clearly advantageous to performing finite differencing, which requires $2N$ propagations.

2.3.1 General objective function F

Now that we have detailed the algorithm in the special case where the target density matrix $\rho_t = \sigma$ is a pure state, we outline the considerations for the general case. We can consider any objective function $F(\rho_f)$ that depends on the final state. One such example is the fidelity,

$$\mathcal{F}(\rho_f, \rho_t) = \left(\mathrm{Tr}\sqrt{\sqrt{\rho_f}\rho_t\sqrt{\rho_f}}\right)^2,$$

where we no longer assume that the target state ρ_t is a pure state. We also consider the case where a control parameter affects multiple system propagator. The chain rule step now reads

$$\frac{\partial F}{\partial c_a} = \sum_n \sum_{i,j,k}^{d_{H_S}^2} \frac{\partial F}{\partial \rho_f^i} \frac{\partial \rho_f^i}{\partial U_n^{jk}} \frac{\partial U_n^{jk}}{\partial c_a}. \tag{2.10}$$

We have re-added the sum over the system propagator time-steps, and see that instead of back-propagating the transposed target state we instead back-propagate the rank one tensor (in Liouville space) $\frac{\partial F}{\partial \rho_f^i}$. Provided that $\frac{\partial F}{\partial \rho_f^i}$ is not overtly challenging to calculate, this will not have a significant impact on the numerical cost of the algorithm. This is because all of the additional steps apply to $(d_{\text{sys}}^2, d_{\text{sys}}^2)$ which are cheap relative to steps that involve summing over the bond index. We further note that here we have drawn the diagrams for a the first order Trotter splitting for ease of understanding. There are additional considerations that need to be accounted for which are detailed in the appendix.

2.4 Optimisation Results

We perform optimisation of state transfer with an initial state

$$\rho_i = |\downarrow\rangle\langle\downarrow| \,. \tag{2.11}$$

We choose

$$\sigma = |\uparrow\rangle\langle\uparrow| \,, \tag{2.12}$$

and perform optimisation for a process time of 5 ps. We choose an initial guess of a constant pulse of area π conjugate to S_x,

$$H_S(t) = \frac{\pi}{5} S_x \,\text{ps}^{-1} \tag{2.13}$$

which would perform state transfer perfectly for unitary dynamics. We seek optimal control parameters $h_x(t)$ and $h_z(t)$ using the `BFGS` algorithm, without any constraints or bounds on the allowed protocol. The numerical parameters used for the optimisation were $\Delta t = 0.05$, $\Delta k_{\max} = 60$ and an SVD cut-off of 10^{-7} for calculations of baths at 1K and 5K, and $\Delta t = 0.025$, $\Delta k_{\max} = 120$, SVD cut-off 10^{-7} for the 20K calculation. This is a local optimisation algorithm, however we anticipate that the optimiser will not get stuck on local minima, since high dimensional (200 and 400 control parameters respectively) fitness function landscapes do not contain local minima, but rather at saddle points [108].

The optimal procedure are shown in Figs. 2.5–2.7 for an environment temperature of 1,5,20K respectively. The broad trend is the pulse intensity that the optimiser deems necessary to mitigate against decoherence increases significantly as we go to higher temperature.

There are two issues with this. The first issue is that achieving high fidelity state transfer by applying high intensity and short duration pulses is not a particularly interesting solution. This can be seen by comparing the pulse intensity to the spectral density. A narrow and intense pulse drives the dot in a region of the spectral density where there is almost no coupling between the system and the environment. The system is effectively being driven faster than the bath can react, and undergoes very little decoherence. It should be noted that the initial state and target state are both decoherence free subspaces in the absence of any driving Hamiltonian. One way of viewing this process, particularly in the 20K case, is that the optimiser is keeping the state close to the decoherence free subspace, and using small adjustments while "teeing it up" for the large pulse that hits it from spin down to spin up, then dealing with any residual non-Markovian effects before the final measurement. While it is nice to see that our optimiser is indeed doing its job correctly and identifying high fidelity methods for performing state transfer, we would like to see

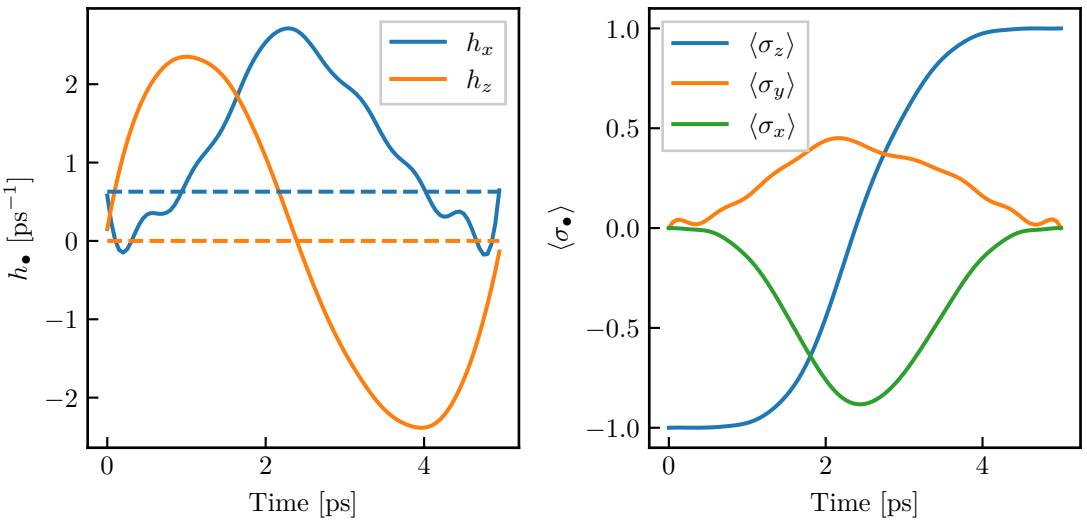


Figure 2.5: Optimal protocol (left) and dynamics (right) for an unconstrained optimisation of state transfer from $|\downarrow\rangle$ to $|\uparrow\rangle$ for a super-Ohmic spectral density at 1K. The optimal control protocol is shown using a solid lines in the left plot with the initial guess shown using dashed lines.

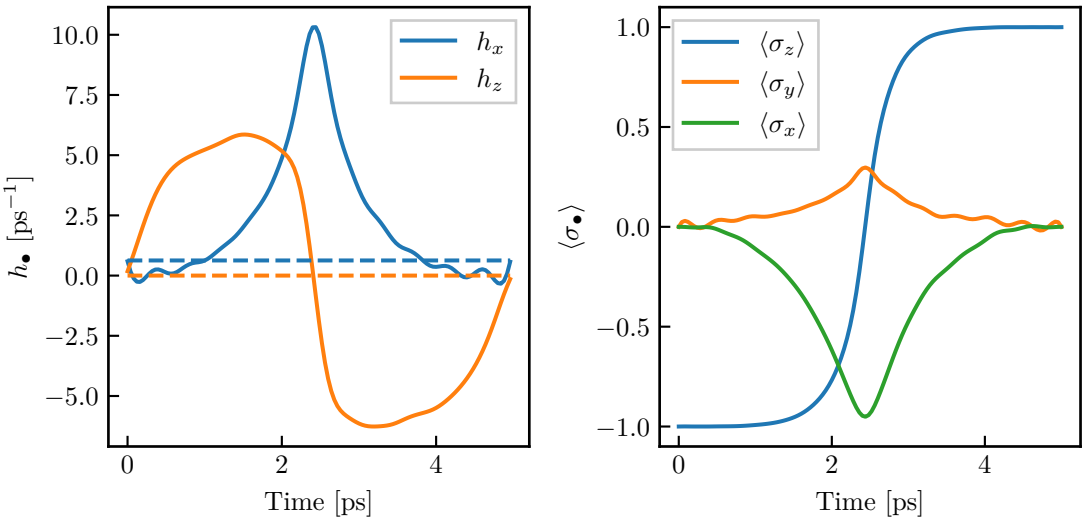


Figure 2.6: Optimal protocol (left) and dynamics (right) for an unconstrained optimisation of state transfer from $|\downarrow\rangle$ to $|\uparrow\rangle$ for a super-Ohmic spectral density at 5K. The optimal control protocol is shown using a solid lines in the left plot with the initial guess shown using dashed lines.

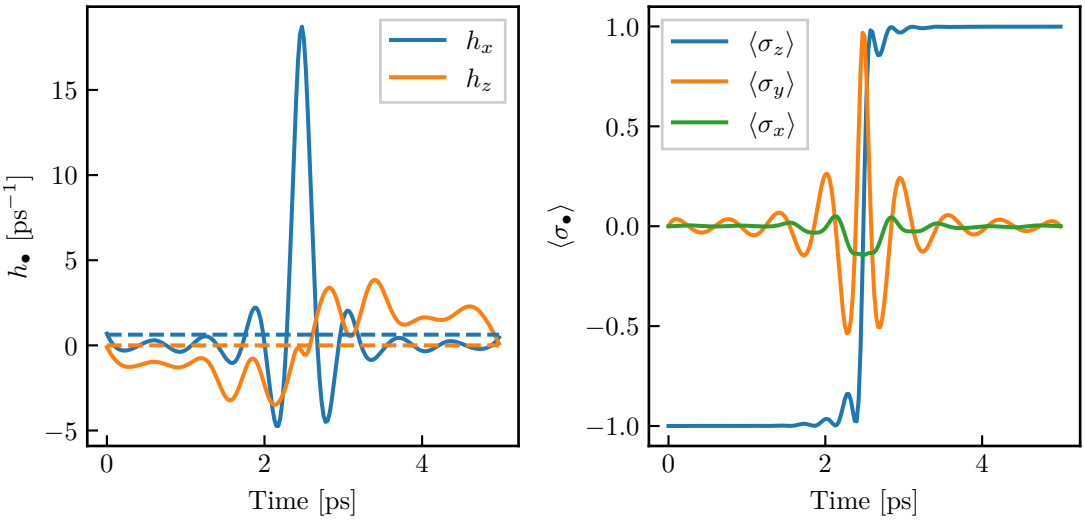


Figure 2.7: Optimal protocol (left) and dynamics (right) for an unconstrained optimisation of state transfer from $|\downarrow\rangle$ to $|\uparrow\rangle$ for a super-Ohmic spectral density at 20K. The optimal control protocol is shown using a solid lines in the left plot with the initial guess shown using dashed lines.

if the optimiser can address the highly non-trivial aspects associated with having non-Markovian dynamics, namely information backflow. Instead the optimiser has effectively identified a "cheat" solution.

The second issue is a practical issue, as it pushes us into parameter regimes where we violate approximations used in deriving our model. An example of a violation is the assumption that the quantum dot can be modelled as a two level system, since we have assumed that the pulse intensity will not access higher energy states such as the biexciton state. This is not an issue if one recasts the physical interpretation as being a paradigmatic tool to showcase what can be done using optimisation in non-Markovian systems, however adopting this approach diminishes the physical relevance of this results.

In order to mitigate the issue we impose bounds on the allowed pulse intensity. Imposing bounds results in a whole range of additional considerations and opens up a plethora of interesting physical interpretations, which are the subject of the next chapter.

Chapter 3

Optimising performance of open quantum systems

In the introduction we introduced the quantum speed limit (QSL): a bound on the shortest possible time to evolve between an initial state ρ_i and a final state ρ_f. In this chapter, we discuss the case where one wishes to engineer a system Hamiltonian that will steer our system from ρ_i to a target state σ, and ask the same question: what is the shortest time in which it is possible to do so. This task will, in general, necessitate dealing with the influence of noise – that is possibly non-Markovian – on the trajectory of our system. From the opening two sentences in this chapter, one would expect the answer to be related to the QSL. And indeed, in a closed system, the QSL does in fact give us the answer: it is simply a singular time: t_{QSL}, which is the shortest possible time to evolve between ρ_i and σ. This is not the case when one introduces the influence of a bath, as in general, one cannot identify a Hamiltonian that maps ρ_i to σ with perfect accuracy. In this chapter we show that when considering an open system instead of considering a singular time, t_{QSL}, one must instead look at a trade-off between speed and fidelity. The reason is that one cannot perfectly perform an operation in a dissipative regime, meaning that we must consider the fidelity as well as the speed associated with an operation. Furthermore, we find that that these two attributes can compete. We use optimal control to identify control protocols that will steer our initial state to the target state with as high of a fidelity as possible. We will see how in a Markov case the best one can do is operate at t_{QSL} for the corresponding closed system, whereas the optimiser is able to exploit the non-Markovian bath to prevent information loss to the environment and even recover it.

3.1 Background

The model we will work with is the same as that described in section 2.2, a two level system coupled to an environment with a super-Ohmic spectral density (SD), resembling acoustic phonons in an InGaAs quantum dot. We chose a definite initial state, ρ_i and attempt to drive our system to a desired target state $\sigma = |\psi_\sigma\rangle\langle\psi_\sigma|$, where we assume both states are pure states. Since for the contents of this chapter, our target state will be a pure state, we can assume the function of merit, the fidelity, is given by

$$\mathcal{F}(\rho_f, \sigma) = \langle\psi_\sigma|\rho_f|\psi_\sigma\rangle\,, \tag{3.1}$$

as in section 2.2. Again we emphasise that as explained in section 2.3, this is not a requirement and the methods required for mixed state target states are described in that section. The goal of this section is to identify the fastest speed that state transfer can be performed at. As observed in the results presented in section 2.4, one can generally find protocols that will drive the system to the desired target state with a relatively high accuracy.

As we discuss in the summary of the results in section 2.4, the means for achieving high fidelity state transfer is to drive the quantum dot with a increasingly larger field strength for shorter times. While this demonstrates the power of optimal control, it is a deceptively easy problem as one is accessing a part of the spectral density with very little coupling between the system and the bath, which itself minimises decoherence. It is also not useful, as one needs increasingly large laser pulses which will input an unacceptable amount of energy into a quantum dot.

Most physical systems have constraints on the controls that are possible. In the next chapter we will briefly deal with the case of a quantum dot driven by an electromagnetic field shaped by a spatial light modulator (SLM). This has the obvious advantage of being a direct implementation of how an experimentalist would parametrise an optimal control experiment, thus ensuring that all protocols that were found by the optimiser resulted in control sequences that are physical, and furthermore can be realistically generated by an experimentalist.

A disadvantage of that approach is that it may make it more difficult to identify those limitations which are fundamental, rather than representing a particular experimental set up. In addition, constraining the optimisation in this way may introduce local minima, making it more difficult to solve numerically and necessitating global optimisation procedures.

For these reasons, we will parametrise our two level system as in the previous chapter. Our Hamiltonian is

$$H_S = h_x(t)S_x + h_z(t)S_z, \tag{3.2}$$

where $h_\bullet(t)$ are the controls, which we treat as piecewise constant over the time-steps Δt. We denote each individual parameter in discrete time as $h_\bullet(t_n)$ where $t_n = n\Delta t$. As before this means that our solutions can be as smooth as our time-step allows, and we do not have to deal with the possibility of the optimiser getting stuck because of some non-trivial non-linear mapping between the control parameter(s) and the physical fields that they generate. We choose to parametrise the control field with the a large number of control points possible as we are fortunate enough to have access to a means of computing the gradient with respect to all parameters that does not increase in difficulty with an increasing number of optimisation parameters. Therefore, it will actually be beneficial to optimise over the maximal amount of control parameters as high dimensional spaces typically have saddle points rather than local minima [108], reducing the need for global optimisation, and increasing the probability that the fidelities associated with the minima we obtain are close to the global optimum. This will enable us to use optimal control to investigate the performance limits of quantum operations with non-Markovian decoherence.

3.2 Performance vs Fidelity

In order to prevent the arbitrarily fast dynamics the simplest solution is to introduce bounds on the control field strength.

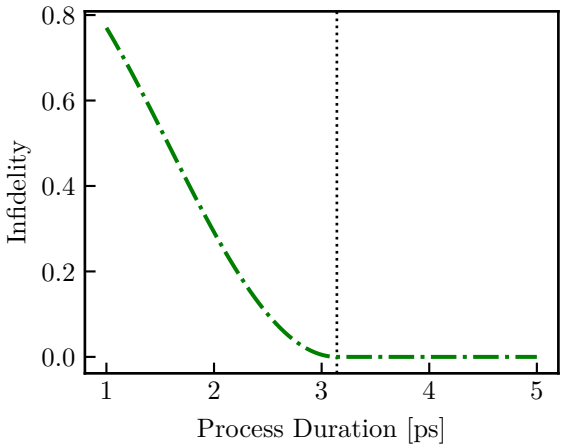


Figure 3.1: Infidelity, $1 - \mathcal{F}$ at the final time t_f of a closed Hamiltonian, Eq. 3.11, a π pulse. The vertical dotted line is $t_{\text{QSL}} = \pi$

Introducing Bounds

In the following, we will consider the case where we set bounds on the maximum allowed control field strength $|h_\bullet| \leq h_\bullet^{\max}$; we allow $h_z^{\max}$ and $h_x^{\max}$ to be different. We start this section by showing the key point of this thesis by considering a specific set of results for given parameters. For the duration of this section we choose to start from an initial state $\rho_i = |x_-\rangle\langle x_-|$ and a target state $\sigma = |x_+\rangle\langle x_+|$, and we set the bound $|h_x| \leq 5\,\text{ps}^{-1}$ and $|h_z| \leq 1\,\text{ps}^{-1}$. We start by considering the case of a closed system, and we show how the bounds give rise to a physical realisation of the QSL in a closed system. We then show how one can use optimal control to find protocols at each time-step that will maximise the fidelity for a given protocol duration. We then discuss the performance-fidelity trade-off and the nature of the dynamics.

3.2.1 Hamiltonian dynamics

The neat aspect about considering state transfer in the unitary case is that one can identify the solution by hand. We seek a (possibly time dependent) unitary $U(t)$ that satisfies

$$|x_+\rangle = U(t)|x_-\rangle. \tag{3.3}$$

Allowing driving along two fields conjugate to S_x and S_z as in Eq. 3.2, any form of h_x and h_z satisfying

$$\underset{\leftarrow}{T} \exp\left(-i\int_{t_i}^{t_f} d\tau H(\tau)\right) = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \tag{3.4}$$

will succeed.

As we are looking to evolve from ρ_i to σ in the shortest possible time, we wish to follow the geodesic on the Bloch sphere. Since unitary dynamics does not allow one to leave the surface of the Bloch sphere, the geodesic is a great circle on the Bloch sphere. Since our Hamiltonian only has terms conjugate to S_z and S_x, the simplest great circle is the equator, which is followed by just applying a field conjugate to S_z. So, we will look at optimising the case where only h_z is non-zero. Now we find that

$$\exp\left(-i\int_{t_i}^{t_f} d\tau h_z(\tau) S_z\right) = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}, \tag{3.5}$$

which is true when

$$\int d\tau h_z(\tau) = \pi n \quad \text{for n odd.} \tag{3.6}$$

For the duration of this chapter, we set the initial time t_i to be zero so t_f is equal to the process duration. In the absence of bounds any $h(t)$, any form where the area is π will drive the state form the initial state to the final state.

Now we introduce the bounds on our control parameters. This brings us to the key point, that as a result of imposing a bound on the maximum allowed driving strength, we also set a time, $\tilde{t}$ such that one cannot perfectly perform state transfer as it is not possible to get a pulse of area π with in that time. To find $\tilde{t}$ we saturate the bound and find the shortest time that will yield area π. Since our bound is constant, $h_z(t) = h_z^{\max}$, and the Hamiltonian that saturates the bound will also be time-independent, which is explicitly given by

$$H = h_z^{\max} S_z, \tag{3.7}$$

where $h_z^{\max}$ is the bound on h_z. This leaves us with the shortest time possible to implement a π pulse, $\tilde{t}$ being

$$\int_0^{\tilde{t}} d\tau \frac{h_z^{\max}}{\pi}, \tag{3.8}$$

$$\tilde{t} = \frac{\pi}{h_z^{\max}}. \tag{3.9}$$

For times longer than $\tilde{t}$ a simple protocol that will achieve state transfer perfectly is

$$H = \frac{\pi}{t_f} S_z, \tag{3.10}$$

and for times shorter than $\tilde{t}$, the best one can do is to set the driving to the maximum allowed by the bounds, $h_z^{\max}$ for the duration of the pulse. Accordingly, we choose the optimal Hamiltonian for the closed system to be

$$H = \begin{cases} h_z^{\max} S_z & t_f < \frac{\pi}{h_z^{\max}} \\ \frac{\pi}{t_f} S_z & t_f \geq \frac{\pi}{h_z^{\max}} \end{cases}. \tag{3.11}$$

Figure 3.1 shows the fidelity using the Hamiltonian Eq. 3.11 plotted for various process durations, t_f. In Figure 3.1, we can see the boundary between perfect and imperfect is separated by $\tilde{t} = \pi\,\text{ps}$, ($h_z^{\max} = 1\,\text{ps}^{-1}$). We now consider the relationship between $\tilde{t}$ and t_{QSL}.

Relation with the Quantum Speed Limit

Since in this case we have a time-independent Hamiltonian generating unitary dynamics, we can use the Mandelstam-Tamm (MT) bound, Eq. 1.87. The variance of our Hamiltonian for an initial state of $|x_+\rangle$ is $\frac{h_z}{2}$, so setting our maximum allowed driving to the bound, we find

$$t_{\text{QSL}} = \frac{\pi}{h_z^{\max}}. \tag{3.12}$$

Here we see that the minimum process duration, $\tilde{t}$, to drive our system from the initial state to the target state is exactly equal to t_{QSL}. The time-independent protocol we reasoned Eq. 3.11, saturates the QSL bound. This fact is known [109, 110].

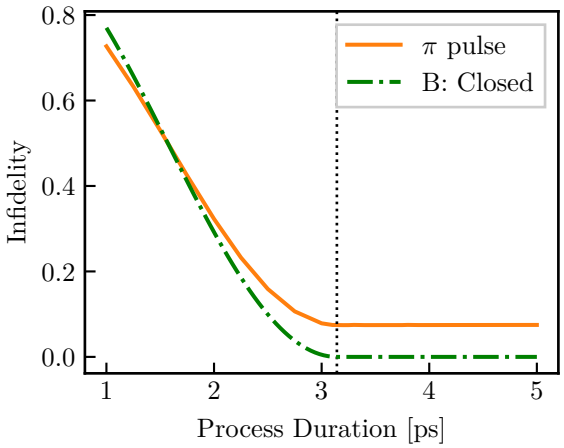


Figure 3.2: Infidelity for the constant π pulse protocol (orange/solid) computed in the open system. The vertical dashed line is t_{QSL}

The MT and Margolus-Levitin QSL, as described in sec. 1.3, set a bound on the minimum time required to evolve between an initial state and an orthogonal state. Equivalently, these bounds can be seen as a limit on the maximum speed of evolution of a state. One difficulty with quantum speed limits is they do not provide information on how to saturate them. Worse still, the bounds themselves are dependent on the Hamiltonian so one arrives in the Sisyphean situation where changing the Hamiltonian to perform the operation faster can simultaneously shorten the QSL accordingly, leaving the ratio of the QSL to protocol duration unchanged.

In this section, we have argued that the quantum speed limit coincides with the fastest time in which state transfer can be achieved in the presence of certain bounds on the driving strength. We have imposed bounds to prevent the possibility that the operation can be performed arbitrarily fast by using arbitrarily large driving Hamiltonians. However such bounds are directly relevant to the speed of operations. In the closed case the QSL defines a singular time separating the regime where the desired operation can be achieved perfectly from that where it cannot. Similarly optimal control yields the same time through a different analysis. In the next section we consider whether a similar connection exist in an open system.

3.2.2 Optimising the performance of state transfer

Constant π pulse in an open system

We start of by considering what is the dynamics of the optimal protocol in a closed system, Eq. 3.11, when applied to an open system. Figure 3.2 shows the result. The obvious difference is there is no region of this graph with perfect fidelity. It is clear that the idea of a quantum speed limit has become less well defined. Whereas previously we had a time, t_{QSL}, that was the barrier between an operation being perfect versus imperfect, there are no such binary outcomes. Since the Hamiltonian in equation 3.11 and the coupling operator between the system and the environment, S_z commute, this process defines the independent boson model (IBM) which is exactly solvable. We will make use of this fact in our later discussion of these results.

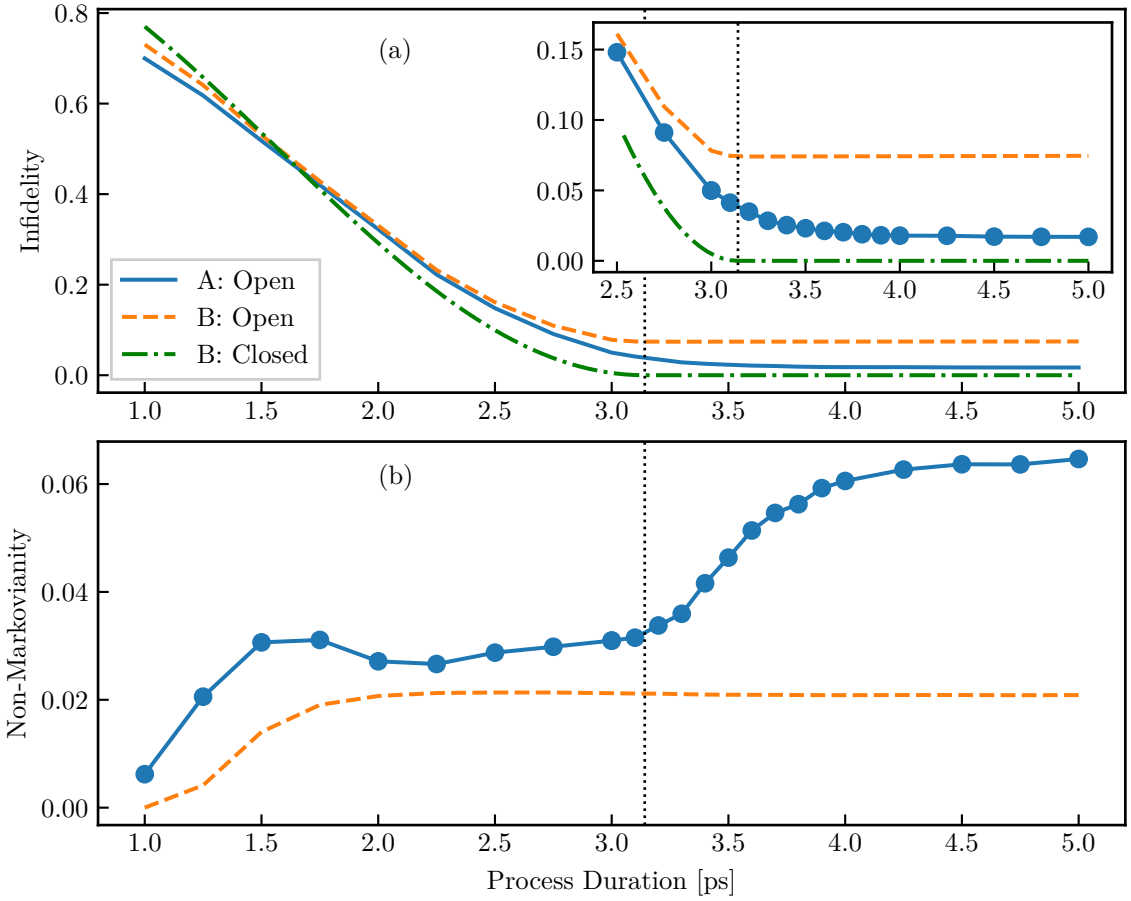


Figure 3.3: (a) Infidelity, $1-\mathcal{F}$ for state transfer as a function of process duration. Results given for the optimal protocol (control A, blue/solid) and for driving with a constant field of pulse area π (control B, orange/dashed). Also shown is the infidelity of control B applied to a closed system (green/dot-dashed). (b) Non-Markovianity measure of Breuer et al. [58, 59] as a function of the process duration for control A and control B applied in a closed system.

Performance versus fidelity

We now come to the central result of this chapter, as reported in [111]. We use the methodology built up in the previous chapters to perform optimisation on our two level system. We choose the initial guess to be the constant π pulse, Eq. 3.11, while treating the Hamiltonian, Eq. 3.2, as piecewise constant over the time-steps of PT-MPO. We perform optimisation to find the optimal Hamiltonian to drive our state from the initial state to the final state, using the `L-BFGS-B` algorithm to maximise the fidelity. This optimisation is performed individually on a set of process times, shown in Fig. 3.3. We are not performing optimisation to drive the two level system as fast as we can per se, rather we attempt to find optimal protocols that perform state transfer as accurately as possible, and consider the optimal fidelity at each protocol duration to investigate the interplay between two aspects of performance, namely speed and fidelity.

Figure 3.3 shows the fidelities associated with the three cases we consider: the optimal closed system dynamics, the closed system's optimal Hamiltonian applied to an open system, and finally the optimised protocol for an open system. For unitary dynamics

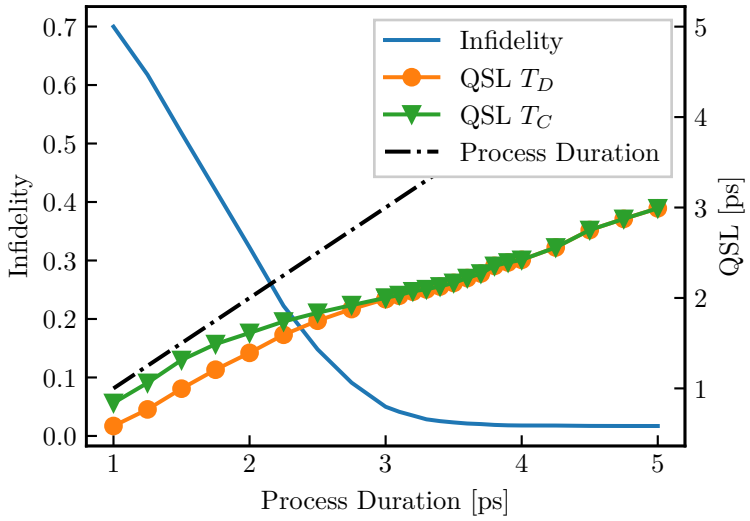


Figure 3.4: Infidelity (blue, left axis), and Deffner and Lutz QSL (orange/circle, right axis), Eq. 1.102 and the Campaioli QSL (green/triangle, right axis), Eq. 1.103

(green/dot-dashed), we can see that, as previously discussed, there are two binary outcomes, separated by $t_{\text{QSL}} = \pi$ (vertical dotted black line). In the case of the constant system π pulse, this control initially performs better than unitary dynamics for very short processes. This is due to the fact that system environment coupling takes the state below the surface of the Bloch sphere. As stated before as the unoptimised protocol duration increases and approaches the QSL the fidelity stops improving. Beyond t_{QSL} the fidelity is roughly constant. This itself is already a non-Markovian feature which comes from the decoherence function of the IBM approaching a constant value for times longer than $\sim 1/\omega_c \approx 0.33$ (we will return to this after solving the IBM). This is contrary to what one would see if one applied this protocol to a pure dissipation Markovian model, where the fidelity would worsen over time, rather than remaining constant. The optimised (blue/solid) fidelity follows a similar trend in process duration, however the remarkable trend is that beyond t_{QSL} the optimiser continues to improve the fidelity as the process duration increases.

3.2.3 Relationship between optimal control and the QSL

Here we recall our discussion on the connection between optimal control and the QSL in the case of unitary dynamics and the unoptimised protocol in an open system. We can see that the optimiser has failed to rectify the problem encountered with the initial guess, that we never achieve perfect fidelity. So, in order to give a qualified assessment as to what is the minimum time required to perform state transfer there is no corresponding notion, rather one must consider the speed (the duration of the protocol) and fidelity achieved simultaneously. This can be equivalently stated as: in order to consider a minimum time to perform state transfer, one must first decide on a tolerable fidelity for state transfer, and only then can one identify the shortest time to do so.

We have discussed at length how the quantum speed limit (QSL) sets a minimum bound on the time necessary to evolve between two states, and in order to illustrate the idea we have outlined the relationship between optimal control and the QSL in sec. 3.2.1 for a

closed system. The problem we encounter when discussing the QSL of an open system is that the meaning is obscured by the fact that the QSL is the minimum time necessary to evolve between two states — but an open system cannot be perfectly controlled, so, while we can talk about the QSL for the evolution between two states, the state you evolve to is not necessarily going to be the state that you wanted to evolve to. This section will be devoted to answer the question of how to interpret the QSL, trying to identify a connection between the QSL and optimal control, and to argue that optimal control is a better tool to understand how fast one can perform quantum operations in an open system.

QSL for optimal protocols

Shown in figure 3.4 is the quantum speed limit (QSL) obtained by Deffner and Lutz, ref. [74] equation 1.102, as well as that of Campaioli et al. [76] for evolution from the initial state and the *final* state. Also plotted is the actual process duration for each point, the idea being that the value of the two QSLs show the minimum possible time required for a process to evolve from the initial state to the *final* state (not the target state[1].) If the QSL is exactly the process duration than it is said that the protocol is operating at the QSL, and one is using the fastest possible protocol for state transfer, saturating the QSL.

Our protocols are obtained by optimal control, and it is worth re-emphasising how the solutions are obtained. We choose a discrete set of process durations and for each process duration we optimise to find the optimal protocol to perform state transfer to as high an accuracy as possible. Again, we are not optimising for speed, we are optimising for accuracy and identifying which speeds yield the highest accuracy. Now let us consider any of the long-time processes, i.e. process durations between four and five picoseconds. Since the optimiser is seeking solutions that perform state transfer with as high a fidelity as possible, one would not necessarily expect a solution to saturate the QSL, as it has more than ample time to perform the state transfer, rather it is seeking the best path to mitigate decoherence. However, as one approaches the closed QSL ($t = \pi$ [ps], for this bound), the optimiser no longer has the luxury of being afforded sufficient time to perform state transfer, and therefore one would expect to see process durations that are closer to t_{QSL}. The key result is that we do not come close to saturating t_{QSL}, save the trivially discussed short time situation, however we do see the expected behaviour where the QSL becomes more saturated, albeit minor in the region of interest.

There are two unavoidable issues with the QSL. Firstly, it is thought that these bounds are not necessarily tight or attainable [76]. The second problem happens when one tries to attain the QSL, as changing the protocol, and therefore the dynamical map, changes the QSL, as discussed in sec. 3.2.1. One could in theory perform optimisation on a given process duration to maximise saturation of the QSL, however that would be cheating as that just chooses a process that has an unnecessarily long QSL, and, would be difficult to constrain such that it also gives a final state close to the target state[2].

In order to understand the meaning of these formula, let us fall back to our understanding of the connection between QSL and optimal control, So far we have established ideas that

[1] We have examined the QSL where the Bures angle $\mathcal{B}(\rho_1, \rho_2)$ is expressed in terms of the initial state and *target* state, and the denominator just comes from the trajectory actually taken by the system, however one simply ends up with a looser bound.

[2] This could possibly be achieved using the target state in the Bures angle, rather than the final state. However we never tried this as we believe the ultimate goal is not to saturate some formula for the sake of it, rather to answer the umbrella question which is how fast can one perform state transfer.

connect the QSL and optimal control, in both closed and open systems. It is prudent to point out that we are not the first people to do this, as Caneva et al. [112] have studied this , and come to different conclusions.

Landau-Zener model

Caneva et all perform optimisation on the Landau-Zener Hamiltonian

$$H(\Gamma(t)) = \begin{pmatrix} \Gamma(t) & \omega \\ \omega & \Gamma(t) \end{pmatrix} \tag{3.13}$$

where $\Gamma(t)$ is the control function that they optimise using the Krotov algorithm, a gradient based optimisation method. The key point in their paper is they overcome the aforementioned problem of the QSL changing with different trajectories is by parametrising the system such that the QSL is invariant under changing the control parameters.

Like us, they seek optimal parameters to perform state transfer under the Landau-Zener model, and they examine the rate of convergence of the Krotov algorithm for different process durations. Caneva et al. find that the infidelity $1 - \mathcal{F}$ from Krotov algorithm converges asymptotically on a finite infidelity for any process duration less than t_{QSL}, whereas for process durations longer than t_{QSL}, they find that the Krotov algorithm converges exponentially towards zero. They then go a step further and say that optimal control can be used to characterise the quantum speed limit.

The the difference between our results and Caneva et al. is as follows. Firstly, Caneva et al. perform their analysis on a closed system (albeit a more useful parametrisation). A difficulty with trying to generalise their conclusions regarding the relationship between optimal control and the QSL is that it is not guaranteed that firstly there exists a solution that performs a quantum operation with an infidelity arbitrarily close to zero, nor that an optimiser can find it. The other difference is their parametrisation conserves the QSL, which ours does not.

What can we salvage from the QSL

In this chapter, we have attempted to find a link between optimal control and the QSL. It is my view that while QSLs for open systems can be calculated, the physical interpretation of them is greatly diminished. There there appears to be no clear notion of a quantum speed limit in an open system, rather it is a performance fidelity trade off, as seen in Fig 3.3. So, rather than seeking to abstract meaning from formulas 1.102 and 1.103, one should perform optimal control to examine the performance fidelity trade-off, and to view this as the meaningful answer to the question of how fast one can perform state transfer.

3.3 Understanding non-Markovian Behaviour

In this section we discuss the role of non-Markovianity in our results. Optimal control of Markovian dynamics has been extensively studied (see references [3, 5], and the citations contained therein), whereas optimal control of non-Markovian systems is relatively new (see references [11–19]). We see that the improvement in fidelity in Fig. 3.3 coincides with a marked increase in our measure of non-Markovianity. We suggest one possible mechanism that the optimiser is exploiting to prevent decoherence.

3.3.1 Measures of non-Markovianity

One way to start addressing the role of non-Markovianity is to plot a measure of non-Markovianity. We have discussed at length the measures of non-Markovianity in the introduction, and for the reasons outlined there we use that proposed by Breuer, Laine and Piilo [58, 59]. This is shown in Fig. 3.3 (b), for both the optimised, and unoptimised case. The measure starts at almost exactly zero for the unoptimised case, presumably because, since the measure is of the information backflow from the environment to the system, information must have flowed away from the system first in order for it to be brought back, and there has been insufficient time for that to happen. For longer process durations the measure increases and saturates at a value of roughly 0.02 shortly after 2 ps. This can be seen from the Bloch vector of the unoptimised protocol, the time for which the non-Markovianity measure stops increasing is the same time that the Bloch vector ceases to recover in length (see Fig. 3.6 (c) below). The optimised protocol always has a higher degree of non-Markovianity, even for lower times. The remarkable result happens after t_{QSL}, where the non-Markovianity measure almost doubles between the process durations of three and four picoseconds. We believe that this is due to the fact that in the region near and before the QSL, the optimiser is in "firefighting mode", where it is severely constrained on what changes it can make to the protocol because it is struggling to have enough time to allow for the procession around the Bloch sphere. For longer times (say times between 3.5 and 4 ps), the optimiser has more wiggle room, which it then uses to — quite literally — create wiggles on the control protocol.

3.3.2 Solving the Independent Boson Model

As mentioned previously, the initial guess used by our optimiser gives an analytically solvable dynamics. In this section we provide the solution from which we can understand the physics of what the optimiser is doing to improve the fidelity.

The Independent Boson Model (IBM) [25] occurs when the coupling operator and system Hamiltonian commute. In the case of our Hamiltonian, this looks like

$$H = h_z S_z + \sum_j \omega_j a_j^\dagger a_j + S_z \sum_j g_j.(a_j^\dagger + a_j), \tag{3.14}$$

which is exactly what our initial guess looks like. This Hamiltonian is diagonalised by performing the polaron transform [25, 113]

$$\hat{P} = \exp\left[S_z \sum_j \frac{g_j}{\omega_j}(a_j - a_j^\dagger) \right], \tag{3.15}$$

which leads to the result given in reference [113]. We partition the total Hamiltonian, Eq. 3.14, as to be composed of H_S, H_B and H_I, referring to the system, bath and interaction Hamiltonians respectively. We now define the "free" Hamiltonian as $H_0 = H_S + H_B$ and the interaction Hamiltonian as before. After applying the polaron transform, the total propagator is $U(t) = U_0(t)U_I(t)$, where the free propagator is $U_0(t) = e^{-iH_0 t}$, and the interaction propagator is

$$U_I(t) = \exp\left[2S_z \sum_j (\alpha_j(t) a_j^\dagger - \alpha_j^*(t) a) \right], \tag{3.16}$$

with the time dependent coupling given by

$$\alpha_j(t) = \frac{g_j}{2\omega_j}(1 - e^{i\omega_j t}). \tag{3.17}$$

We assume, as always, that the system and bath start in a product state,

$$|\psi(0)\rangle = (c_1 |\uparrow\rangle + c_2 |\downarrow\rangle) \otimes \rho_{\text{bath}}. \tag{3.18}$$

We now decompose the total wavefunction into one where the environment sees the system in a spin down state, and one where it sees a spin up state

$$|\psi(t)\rangle = c_1 |\uparrow\rangle \otimes \left|\psi^{\uparrow}_{\text{bath}}(t)\right\rangle e^{-\frac{i\omega_0 t}{2}} + c_2 |\downarrow\rangle \otimes \left|\psi^{\downarrow}_{\text{bath}}(t)\right\rangle e^{\frac{i\omega_0 t}{2}}. \tag{3.19}$$

It is convenient to write down the time evolution for the system raising and lowering operators, because we can use that $S_- |\uparrow\rangle = |\downarrow\rangle$ to find

$$\langle S_- \rangle = c_1 c_2^* \left\langle \psi^{\downarrow}_{\text{bath}}(t) \middle| \psi^{\uparrow}_{\text{bath}}(t) \right\rangle e^{-i\omega_0 t}, \tag{3.20}$$

where the equivalent term for $\langle S_+ \rangle$ is

$$\langle S_+ \rangle = c_1^* c_2 \left\langle \psi^{\uparrow}_{\text{bath}}(t) \middle| \psi^{\downarrow}_{\text{bath}}(t) \right\rangle e^{i\omega_0 t}. \tag{3.21}$$

Since our initial and target states are eigenstates of S_x, we can find

$$\langle S_x \rangle = \frac{1}{2}(\langle S_+ \rangle + \langle S_- \rangle), \tag{3.22}$$

from which we can write that

$$\langle S_x \rangle = \frac{1}{2}\left(c_1^* c_2 \left\langle U_{\uparrow}^{\dagger} U_{\downarrow} \right\rangle e^{i\omega_0 t} + c_1 c_2^* \left\langle U_{\downarrow}^{\dagger} U_{\uparrow} \right\rangle e^{-i\omega_0 t} \right). \tag{3.23}$$

Solving for the Bath Influence using the Coherent State Path Integral

The goal of the next section is to evaluate the bath influence terms $\left\langle U_{\uparrow}^{\dagger} U_{\downarrow} \right\rangle$ and $\left\langle U_{\downarrow}^{\dagger} U_{\uparrow} \right\rangle$ for a finite temperature bath. Writing the former out using the propagator, we see that for a single bath node

$$U_{\uparrow}^{\dagger} U_{\downarrow} = \exp(\alpha^* a - \alpha a^{\dagger}) \exp(i\omega a^{\dagger} a) \exp(-i\omega a^{\dagger} a) \exp(-\alpha a^{\dagger} + \alpha^* a), \tag{3.24}$$

which simplifies to

$$U_{\uparrow}^{\dagger} U_{\downarrow} = \exp[2(\alpha^* a - \alpha a^{\dagger})] = \mathcal{D}(\alpha), \tag{3.25}$$

which a displacement operator. Using similar steps we find that

$$\left\langle U_{\downarrow}^{\dagger} U_{\uparrow} \right\rangle = \exp[2(\alpha a^{\dagger} - \alpha^* a)], \tag{3.26}$$

where we note here that the two equations differ by replacing α with $-\alpha$ and will make use of this fact later to show that the solution to both is identical, so that we can deal with calculating $\left\langle U_{\uparrow}^{\dagger} U_{\downarrow} \right\rangle$ first. In order to calculate the expectation value of this operator in a finite temperature bath, we follow the usual coherent state path integral steps [47]. The expectation value of an operator $\langle \mathcal{D}(\alpha) \rangle$ is calculated as

$$\langle \mathcal{D}(\alpha) \rangle = \frac{1}{\mathcal{Z}} \sum_n \left\langle n \middle| e^{-\beta H} \mathcal{D}(\alpha) \middle| n \right\rangle = \frac{1}{\mathcal{Z}} \int \frac{d\bar{\psi} d\psi}{\pi^N} e^{-\bar{\psi}\psi} \left\langle \psi \middle| e^{-\beta H} \mathcal{D}(\alpha) \middle| \psi \right\rangle, \tag{3.27}$$

where $|n\rangle$ is a complete set of Fock states, $|\psi\rangle$ a coherent state, and the partition function is

$$\mathcal{Z} = \mathrm{Tr}[e^{-\beta\hat{H}}]. \tag{3.28}$$

Using the Baker-Campbell-Hausdorff formula, we can rewrite the displacement operator as

$$\mathcal{D}(\alpha) = e^{-2\alpha a^\dagger} e^{2\alpha^* a} e^{-2\alpha^*\alpha}, \tag{3.29}$$

where the exponent is expanded such that a appears on the right and $a^\dagger$ to the left, that is the order of the normal ordered Hamiltonian. This means that the equation 3.27 now looks like

$$\langle\mathcal{D}(\alpha)\rangle = \frac{1}{\mathcal{Z}}\int \frac{d\bar{\psi}d\psi}{\pi^N} e^{-\bar{\psi}\psi} \left\langle \psi \middle| e^{-\beta H} e^{-2\alpha a^\dagger} e^{2\alpha^* a} \middle| \psi \right\rangle e^{-2\alpha^*\alpha}. \tag{3.30}$$

Now we perform the usual path integral steps of time slicing such that $\beta \to M\Delta\tau$ and inserting resolutions of the identity at each infinitesimal time step. The Hamiltonian in order to compute the thermal average is $\hat{H} = \sum_k \omega_k a_k^\dagger a_k$, where we note that the Hamiltonian is normal ordered. Substituting this into the above equation gives

$$\frac{1}{\mathcal{Z}}\int \prod_{m=0}^{M-1} \left[\frac{d\bar{\psi}d\psi}{\pi^N} e^{-\bar{\psi_m}\psi_m}\right] \langle\psi_0|\ldots|\psi_m\rangle\langle\psi_m| e^{-\Delta\tau H} |\psi_{m-1}\rangle\langle\psi_{m-1}|\ldots$$
$$\ldots|\psi_1\rangle\langle\psi_1| e^{-2\alpha a^\dagger} e^{2\alpha^* a} |\psi_0\rangle e^{-2\alpha^*\alpha}, \tag{3.31}$$

where nothing extraordinary has happened other than the standard path integral steps. The only part that isn't standard is at the beginning of time, where

$$e^{2\alpha^* a}|\psi_0\rangle = e^{2\alpha^*\psi_0}|\psi_0\rangle, \tag{3.32}$$

with a similar result for $\langle\psi_1|$. The last part then reads

$$\ldots|\psi_1\rangle\langle\psi_1| e^{-2\alpha\bar{\psi}_1} e^{2\alpha^*\psi_0} |\psi_0\rangle e^{-2\alpha\alpha^*}. \tag{3.33}$$

Note, there exists an infinitesimal convergence factor of $\exp[-i\omega_n\delta]$, that has been omitted from the above equation. This appears because the field $\bar{\psi}$ is evaluated at a infinitesimal time-step after the field ψ ([47] page 168). This will be referred to below. Following the normal path integral steps[47], we find that

$$\langle\mathcal{D}(\alpha)\rangle = \frac{1}{\mathcal{Z}}\int \prod_{m=0}^{M} \left[\frac{d\bar{\psi}d\psi}{\pi^N}\right] \exp[-S]\exp[-2\alpha\bar{\psi}(0) + 2\alpha^*\psi(0) - 2\alpha^*\alpha], \tag{3.34}$$

$$S = \int_0^\beta d\tau (H(\bar{\psi},\psi) + \bar{\psi}\partial(\psi)). \tag{3.35}$$

The field ψ can be Fourier transformed using

$$\psi(\tau) = \frac{1}{\beta}\sum_n \psi_n e^{i\omega_n\tau}, \tag{3.36}$$

where ω_n are the bosonic Matsubara frequencies. This is where the convergence factor comes in, because $\bar{\psi}$ and ψ are evaluated at two infinitesimally small different times, so the Hamiltonian in Fourier space actually looks like

$$H = \frac{1}{\beta}\sum_n \bar{\psi}_n\psi_n e^{-i\omega_n\delta}. \tag{3.37}$$

Performing the Fourier transform,

$$\langle \mathcal{D}(\alpha) \rangle = \frac{1}{\mathcal{Z}} \int D(\bar{\psi}, \psi) \exp[-S], \tag{3.38}$$

$$S = \sum_n \left[\frac{\bar{\psi}_n \psi_n}{\beta} (\omega_0 + i\omega_n) - 2\alpha \bar{\psi_n} + 2\alpha^* \psi_n + 2\beta \alpha^* \alpha \right], \tag{3.39}$$

which is then integrated using

$$\int d(\bar{z}, z) e^{-\bar{z} w z + \bar{u} z + \bar{z} v} = \mathcal{Z} e^{\bar{u} v}, \tag{3.40}$$

which gets rid of the partition function to give

$$\langle D(\alpha) \rangle = \exp \left[\sum_n - \frac{4\beta \alpha^* \alpha}{\omega_0 + i\omega_n} + 2\beta \alpha \alpha^* \right], \tag{3.41}$$

which can be rewritten as

$$= e^{-2|\alpha|} e^{-\frac{4|\alpha|}{\beta} S}, \qquad S = \sum_n \frac{1}{\omega_0 + i\omega_n}. \tag{3.42}$$

The convergence factor mentioned above when passed through the Gaussian gets inverted, so changes sign. The sum we wish to calculate takes the form

$$S = \sum_n \frac{1}{\omega_0 + i\omega_n} e^{i\omega_n \delta}. \tag{3.43}$$

Using the usual Matsubara summation techniques[47], the sum is written as a standard form that looks like

$$S = \frac{1}{2\pi i} \oint dz \frac{\beta}{e^{\beta z} - 1} \frac{1}{z + \omega_0} e^{z\delta}, \tag{3.44}$$

where the convergence factor controls the convergence for $z < 0$ and the Matsubara weighting function controls convergence for $z > 0$. The contour integral yields

$$S = \text{res} \left[\frac{\beta}{e^{\beta z} - 1} \frac{1}{z + \omega_0} \right] = \frac{1}{e^{\beta \omega_0} - 1} = n_B(\beta \omega_0), \tag{3.45}$$

which is then substituted into equation 3.42 to give

$$\left\langle U_\uparrow^\dagger U_\downarrow \right\rangle = \exp[-4|\alpha|^2 n_B(\beta\omega_0)] \exp[-2|\alpha|^2]. \tag{3.46}$$

This is the point that we recall that $\left\langle U_\downarrow^\dagger U_\uparrow \right\rangle$ is recovered by setting α to $-\alpha$. Because α only appears inside an absolute magnitude sign, we see that the contribution from the spin up part of the bath and the spin down part of the bath are the same. Factoring this out of equation 3.23 and adding back the sum indexes we find that

$$\langle S_x \rangle = \frac{\exp\left[\sum_j -4|\alpha_j|^2 n_B(\beta\omega_j) - 2|\alpha_j|^2\right]}{2i} \left[c_1^* c_2 e^{i\omega_0 t} + c_1 c_2^* e^{-i\omega_0 t} \right], \tag{3.47}$$

and rewrite

$$\sum_j |\alpha_j|^2 = \sum_j \left| \frac{g_j}{2\omega_j} (1 - e^{i\omega_j t}) \right|^2 \tag{3.48}$$

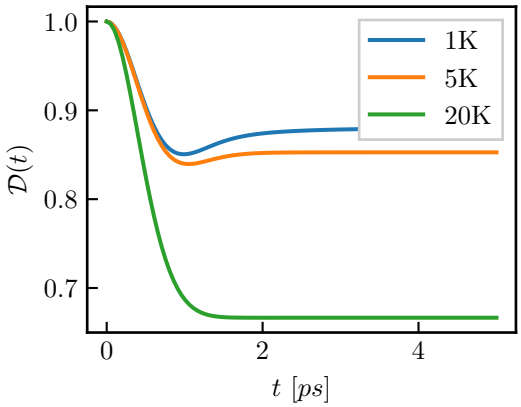


Figure 3.5: Decoherence function Eq. 3.52 plotted for various temperatures using a our super-Ohmic spectral density, $\alpha = 0.126$ and $\omega_c = 3.04$

as

$$= \sum_j \frac{|g_j|^2}{2\omega_j^2}(1 - \cos(\omega_j t)). \tag{3.49}$$

We convert the sum into an integral, $\sum_j |g_j|^2 \, \delta(\omega - \omega_j) \to J(\omega)$ which means the coupling constants become the spectral density

$$\sum_j |\alpha_j|^2 \to \int_0^\infty d\omega \frac{J(\omega)}{2\omega^2}(1 - \cos(\omega t))^2, \tag{3.50}$$

to give a final expression for $\langle S_x \rangle$ of

$$\langle S_x(t) \rangle = \frac{I_B}{2i}\left(c_1^* c_2 e^{i\omega_0 t} + c_1 c_2^* e^{-i\omega_0 t}\right), \tag{3.51}$$

and the decoherence function given by

$$I_B = \exp\left[-\int_0^\infty d\omega \frac{J(\omega)}{\omega^2}(1 - \cos(\omega t))\,(2n_B(\omega\beta) + 1)\right]. \tag{3.52}$$

These results agree with reference [114]. If we express the equation of motion of the density matrix as

$$\rho_{ii}(t) = \rho_{ii}(0) \quad i \in 0, 1 \tag{3.53}$$

$$\rho_{01}(t) = \rho_{01}(0) e^{-2\gamma(t)}. \tag{3.54}$$

Ref. [114] considers two situations: pure dissipation, where the decoherence is described by some fixed rate γ_0, so $\gamma(t) \to \gamma_0 t$, and secondly the IBM. The IBM solution is

$$\gamma(t) = \frac{1}{2}\int_0^\infty d\omega J(\omega) \coth\left(\frac{\omega\beta}{2}\right)\frac{1 - \cos(\omega t)}{\omega^2} \tag{3.55}$$

which reduces to Eq. 3.51 by expanding out the thermal factor. To be consistent with the literature we will stick with this notation, where in the exponent we can clearly see the spectral density term, the thermal term, and the half Fourier transform of the trigonometric term.

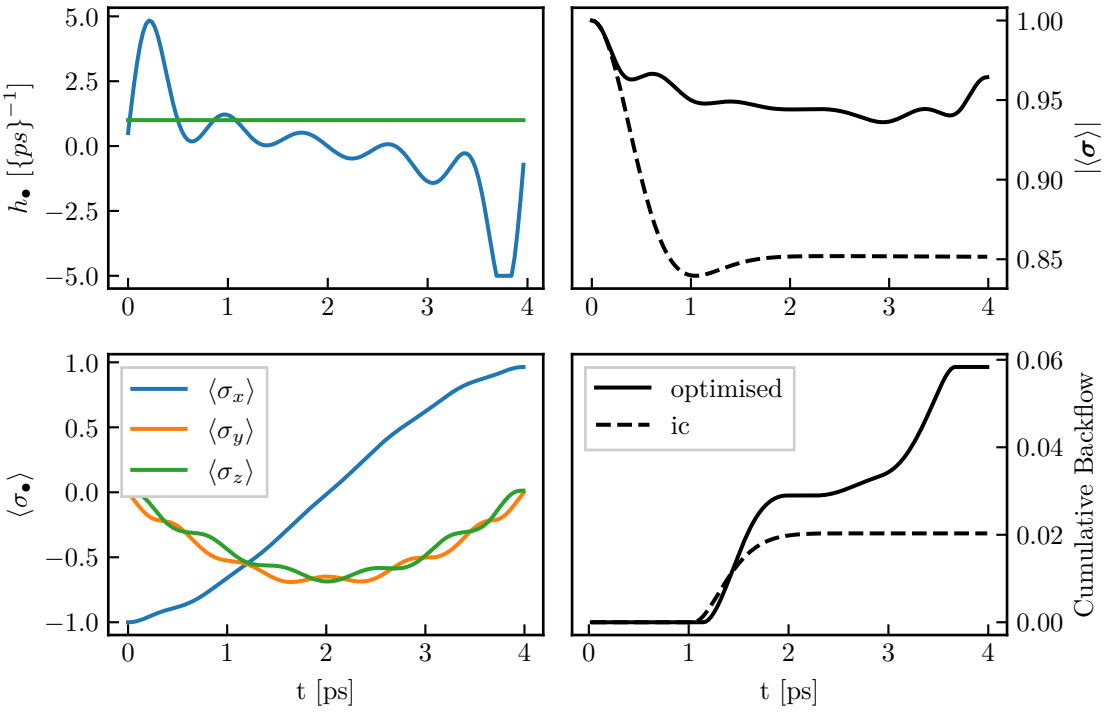


Figure 3.6: The control parameters $h_\bullet$ (a) and dynamics (b) for the optimal protocol for 4 ps, preventing the build up of entanglement between the system and the environment, suppressing the polaron formation. This can be seen in the Bloch vector magnitude (c) for the unoptimised (control B/dashed) and optimised (control A/solid). The cumulative information backflow for both protocols is shown in (d)

Physically, in the initial state when the system and the bath are in a product state, and the latter being in a thermal state, the oscillators thermally distributed in the harmonic oscillator. Since the propagator consists of the displacement operator $\mathcal{D}(\alpha)$, it causes an instantaneous displacement of the harmonic potential of the modes of the environment. This causes the build up of system-environment correlations — the formation of a polaron. We will shortly describe how we propose that the optimal protocol is attempting to suppress this formation of a polaron.

3.3.3 Reversal of the polaron formation

Here we outline a possible physical property that our optimiser is exploiting. Since the decoherence function for our initial guess of the optimisation problem is the IBM, the magnitude of the Bloch vector is I_B as shown in Fig. 3.5 (for 5K). The initial decay in the Bloch vector length of the unoptimised protocol is due to the build up of correlations between the system and environment, the formation of the polaron. This is followed by a small recovery in the Bloch vector length, followed by a long time limit where the fidelity is constant. This is in contrast to the optimised case, for which we show a sample protocol in Fig. 3.6. This protocol does not result in a sharp decay in the Bloch vector length, rather it follows a rather slow decay, until just before the end of the protocol where it exhibits a small increase.

We propose that this is because our optimiser is finding ways to suppress and reverse the

polaron formation, and describe the evidence we have to corroborate our claim.

Sketch of microscopic theory of polaron formation

As discussed previously at equation 3.19, we decompose the environment wave-function into the component of the environment that evolves associated with the system being in a spin up state, and the part that is evolving with a spin down state, with the respective probability amplitudes.

The modes of the bath are initially thermally distributed in their harmonic wells. When the interaction is switched on, the harmonic potentials are displaced by the influence of the state of the environment, in equal and opposite directions depending on whether the system is in a spin up state and spin down state. Since the switching of coupling is instantaneous, bath wave-packets begin oscillating about their new equilibrium position. Here we recall that the decoherence function $\mathcal{D}(t)$ is basically the overlap between the two states of the environment, so in order to suppress the decoherence, one wishes to keep $\left\langle \psi^{\downarrow}_{\text{bath}}(t) \middle| \psi^{\uparrow}_{\text{bath}}(t) \right\rangle$ as close to unity as possible.

Imagining the case where the bath consists of a single mode, when the system-environment coupling is switched on the mode will oscillate at a single frequency. Therefore after a single period of the harmonic potential that mode is in, the wave-packet will be back at the starting position, where the overlap between the two environment states is one. Since our environment has an infinite number of modes all oscillating at different frequencies the time at which the various packets come back to the starting position will be different, which causes the long time steady state of the decoherence function of the IBM.

Increase in non-Markovianity

The LPB measure gives a measure of the non-Markovianity by summing the increase in trace distances between all points on the Bloch sphere. This measure is based on the idea that Markovianity is information monotonically flowing out of the system, which manifests as a loss of information, which is seen in the distinguishability of pairs of states. Knowing this, it's interesting to plot the times where the trace distances increase that contribute to the non-Markovianity measure, since this would lead us to believe that this is the time which the information flows from the environment to the system. A sample is shown in Fig. 3.6 (d) for an optimal protocol for a process duration of 4 ps. For the unoptimised case, the information backflow is restricted to the region of the dynamics where the Bloch vector magnitude (and therefore decoherence function $\mathcal{D}(t)$) increases. This is in contrast to the optimised case. For the first picosecond, there is zero information backflow from the system to the environment. The initial wave of information backflow is more pronounced than that of the initial guess, and in contrast to the initial guess from one picosecond onwards there is almost always information flowing from the environment to the system. Since the unoptimised case is an exactly solvable model we can understand the physics of the decoherence, since it originates from the term $\left\langle U^{\dagger}_{\uparrow} U_{\downarrow} \right\rangle$, the loss of overlap of the two forms of the evolution of the environment wavefunctions. The period where we claim the optimiser is attempting to reverse the formation of the polaron is also the time when the cumulative trace distances are steadily increasing. So in order to retain as much information on our system as possible our optimiser causes backflow of information from our environment to our system, undoing the formation of the polaron while increasing the degree of non-Markovianity in Fig. 3.6 (d), giving the resultant non-Markovianity measure

in Fig. 3.3.

Dynamics after conclusion of suppresion protocol

In Fig. 3.7 we show the dynamics for the optimised protocol for a duration of 5 ps, followed by a time where we switch off the driving and let the state evolve further simply under the influence of the environment. For the first 5 ps we see the optimal protocol attempting to steer the state from the initial state to the final state, while suppressing the decay of the Bloch vector length, as we have described before. From Fig. 3.7, we can that the optimiser has achieved a good — but not perfect — state which is almost $|x^+\rangle$. Once the suppression protocol is switched off the trajectory of the system is bears some resemblance of the trajectory of the IBM.

We suggest that due to the similarity between the dynamics of the system post suppression protocol and the IBM solution of a state where the system and environment are in a product state, that protocol is attempting to prevent the build up of correlations between the system and environment. Our suggestion is that at $t = 5\,\mathrm{ps}$ amount of correlations between the system and environment is very low, therefore once the protocol ceases the correlations build up as they do in the IBM. To highlight this we have plotted the IBM solution starting from a product state at $t = 5\,\mathrm{ps}$, which is indeed similar to the decay seen in the Bloch vector length at that time.

3.4 Conclusions

Here we have shown that slow process durations can yield better fidelity of state transfer by exploiting non-Markovian effects. We have shown the performance of state transfer versus process duration trade-off for a particular system. We have examined the relationship between the quantum speed limit (QSL) for closed systems as well as the additional considerations needed for open systems. It is our view that rather than trying to identify meaningful QSLs for open systems it is preferable to perform optimisation and instead consider the speed-fidelity trade-off to answer the question of how fast one can perform state transfer. We present a possible mechanism for how the optimiser is improving fidelity by suppressing the formation of the polaron. So far we have only considered the performance-fidelity of super-Ohmic systems. This methodology could be extended to optimisation of gates, and other physical systems. Since the PT can be obtained from tomography, this could allow experimentalists to obtain a practical answer to how fast they should perform tasks in order to achieve a desired outcome.

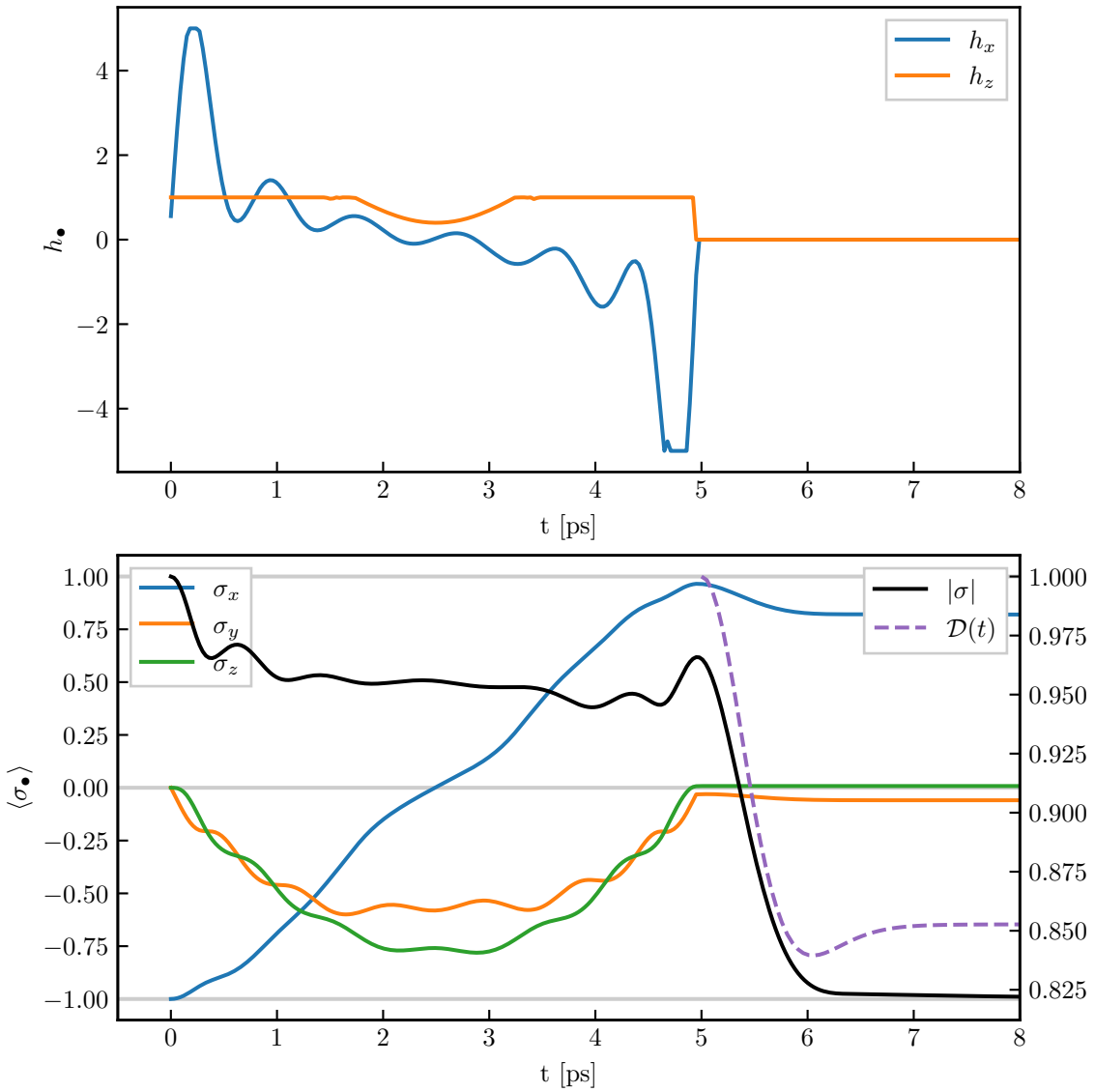


Figure 3.7: Evolution of the state with the optimal control for $t < 5\,\text{ps}$, and zero Hamiltonian for $t > 5\,\text{ps}$, after the process has been completed. (a) Shows the controls, (b) shows the dynamics of the spin expectation values (blue/yellow/green, left axis), as well as Bloch vector magnitude (black/solid, right axis). The purple/dashed curve is the decoherence function of the IBM for a product state at $t = 5\,\text{ps}$ (right axis).

Chapter 4

Optimal control of SLM pulse shaping

So far we have considered a system Hamiltonian of the form

$$H_S(t) = h(t)S_x + h_z(t)S_z, \tag{4.1}$$

and have described how our environment model is an approximation for a InGaAs quantum dot coupled to the bulk phonons. Yet we have not elaborated the physics of how the laser drives the quantum dot, nor have we indicated how one would shape a laser to give a control field $h_\bullet(t)$. In this chapter we describe some of our earlier work where instead of directly identifying optimal controls $h_\bullet(t)$, we instead find the optimal configuration of a standard pulse shaper used in experiment, a spatial light modulator (SLM). This setup guarantees that every possible parameter space configuration corresponds to an experimentally achievable control pulse. We describe how to model the SLM and use the process tensor to model the influence of the environment. Since this work was performed prior to the gradient based optimisation, here we used global optimisation rather than local optimisation. Additionally, we expect that due to the highly non-trivial mapping between the control parameters and the resultant electric field — in addition to being coupled to a non-Markovian environment — the objective landscape may be far more pathological than the case considered in the previous two chapters.

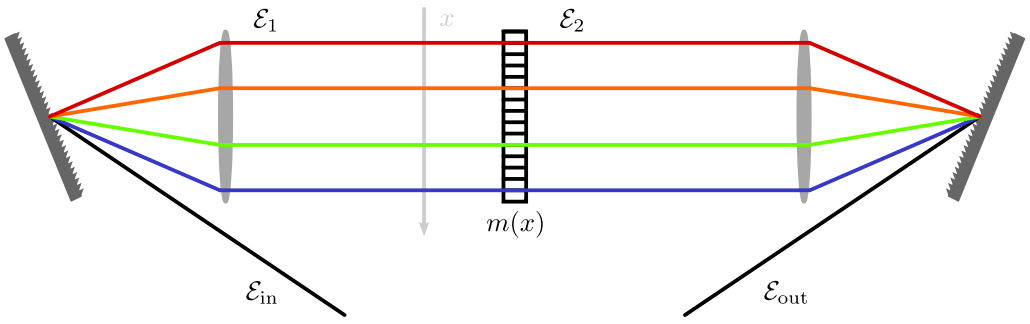


Figure 4.1: SLM pulse shaper setup.

4.1 Pulse shaping using an SLM

Here we given an overview of how we use an SLM to shape a pulse, and the considerations required for modelling numerically. We follow the findings of references [115–117] closely. The setup is shown in Fig. 4.1. We decompose the real electric field as [115]

$$E(t) = \frac{1}{2}\left(\mathcal{E}(t) + \mathcal{E}^*(t)\right) \tag{4.2}$$

where the complex electric field in the time domain is given by

$$\mathcal{E}(t) = \vec{e}E_0(t)\exp\left[-i\int \omega(t)dt\right] \tag{4.3}$$

where $\omega(t)$ is the (possibly time dependent) carrier frequency, $\vec{e}$ is the polarisation vector, and $E_0(t)$ is the pulse envelope. Here we assume that our source laser produces an unchirped Gaussian laser pulse, so

$$\mathcal{E}_{\text{in}} \propto \frac{1}{\tau}e^{-\frac{t^2}{\tau^2}}e^{-i\omega_c t} \tag{4.4}$$

where τ is the pulse duration and ω_c is the constant carrier frequency. Typical times for τ are between 30 and 300 fs.

This pulse is then applied to the diffraction grating which Fourier transforms to the frequency domain, which is applied to a lens. After the lens ($\mathcal{E}_1$ in diagram 4.1), the pulse is now spread out on the table in frequency space,

$$\mathcal{E}_1(\omega) \propto \exp\left[-\frac{(\omega+\omega_c)^2\tau^2}{4}\right]. \tag{4.5}$$

This laser is now passed through the SLM, where each SLM pixel modifies an individual region of the frequency of the pulse. The SLM is a discrete set of pixels that can attenuate the frequency component or cause a phase delay of the frequency component by up to 2π. Here we will not consider attenuating the electric field and only consider phase delays. Accordingly, we write the SLM modulation function as

$$m(\omega) = e^{i\varphi(\omega)}. \tag{4.6}$$

For simplicity we have written this as a continuous function but generally this will contain a discrete set of pixels $\{\varphi_i\}$, that are piecewise constant over each SLM pixel. The frequency space profile post SLM roughly looks like

$$\mathcal{E}_2(\omega) \propto \exp\left[-\frac{(\omega+\omega_c)^2\tau^2}{4} + i\varphi(\omega)\right]. \tag{4.7}$$

This is then Fourier transformed back into the time domain where $\mathcal{E}_{\text{out}}$ is passed to the quantum dot. This is a standard pulse shaping technique. For example, applying a parabolic phase delay would give a chirped output. Also note that since the phase delays are periodic, so

$$e^{i\varphi(\omega)} = e^{i\varphi(\omega)+k2\pi}, \tag{4.8}$$

and one only needs to consider $\varphi(\omega) \in [0, 2\pi]$ for all possible SLM configurations, and any phase delay profile (e.g. the parabola) can be wrapped around into this range.

So far we have only talked about the temporal and spectral shape of the laser pulse. Another consideration that must be accounted for is that the laser pulse also has a finite spatial width as well as temporal and spectral width. This means that after the pulse hits the diffraction grating and is focused on the SLM, each dispersed frequency component will have a finite spot size on the SLM pixels. We assume that the spatial profile of the laser pulse is given by,

$$S(x) = e^{-\frac{x^2}{\Delta x^2}}, \tag{4.9}$$

where Δx is the spatial width of the laser pulse in the time domain ($\mathcal{E}_{\text{in}}$ in Fig. 4.1). Then, reference [115] say the SLM modulating function, including the effect of a Gaussian finite spatial width of the laser pulse is

$$m(\omega) = \exp\left(-\frac{\omega^2}{\Delta\Omega_x^2}\right) * \sum_n \text{squ}\left[\frac{\omega - \Omega_n}{\Delta\Omega_p}\right] e^{i\varphi_n}, \tag{4.10}$$

where $\Delta\Omega_x = \Delta x \delta\Omega_p/\delta x$, δx is the SLM pixel separation in real space and $\delta\Omega$ is the corresponding pixel separation in frequency space, and $*$ denotes convolution.

If we assume that the input pulse is a temporal delta function, so $\mathcal{E}_{\text{in}}(\omega) = 1$, then the resultant electric field in the time domain is [24, 115]

$$\mathcal{E}_{\text{out}}(t) \propto \exp\left(-\frac{\Delta\Omega_x^2 t^2}{4}\right) \text{sinc}\left[\frac{\delta\Omega_p t}{2}\right] \sum_n \exp\left[i\Omega_n t + i\varphi_n\right], \tag{4.11}$$

which is the shaped field we use to control our dot.

As discussed in Ref. [115], we see that the finite nature of the spatial spot size blurs the discrete feature of the SLM mask. If spot size is too small, replica waveforms will arise. This is undesirable, since the goal here is to provide a framework for pulse shaping and optimal control to implement quantum operations. Implementing quantum operations would be difficult if effects from our control protocol for one operation have replica waveforms at long times, when we ultimately would wish to perhaps perform a second operation. Additionally, it is also challenging to simulate, as long time simulations are also computationally costly. On the other hand, if the spot size is too big, the blurring of the mask will give rise to substantial diffraction effects [115].

Equation 4.11 is derived assuming the SLM has perfectly behaving pixels, as well as a finite spatial spot size of the laser. SLMs are not perfectly sharp, and there are gap regions between the pixels whose properties are somewhat intermediate between the two adjacent pixels [115]. For simplicity we have neglected to simulate this, or any other considerations associated with a liquid crystal spatial light modulator.

4.1.1 Physical model

The physical model is the same as the previous two chapters, a InGaAs dot under the influence of phonon dephasing. The two level system is coupled to a bath with a super-Ohmic spectral density, Eq. 2.4. This time we pay particular attention to the physical origin of the system Hamiltonian. The Schrödinger picture Hamiltonian with the rotating wave approximation is

$$H_S(t) = \frac{\omega_{\uparrow\downarrow}}{2}\sigma_z + \frac{\Omega(t)}{2} e^{-i\int \omega_c(t)dt}\sigma^+ + \frac{\Omega(t)}{2} e^{\int \omega_c(t)dt}\sigma^-, \tag{4.12}$$

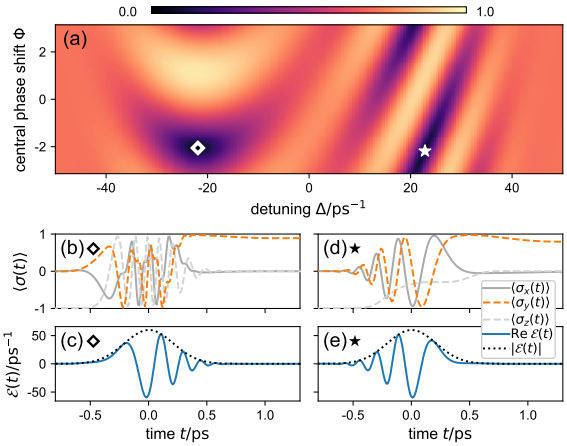


Figure 4.2: (a) Heat map of infidelity of state transfer of a quantum dot from the $|\downarrow\rangle$ state to the $|\uparrow\rangle + i\,|\downarrow\rangle$ state. Also shown is the dynamics and the electric field (b–e) for two minima marked by a diamond and star respectively. Figure reproduced from Ref. [24].

where $\omega_{\uparrow\downarrow}$ is the exciton transition energy, $\omega_c(t)$ is the laser carrier frequency, and the Rabi frequency is given by

$$\Omega(t) = E_0(t) \left\langle \downarrow \middle| \vec{\hat{d}} \cdot \vec{e} \middle| \uparrow \right\rangle . \tag{4.13}$$

Here $\vec{\hat{d}}$ is the dipole operator and $\vec{e}$ is the polarisation vector. Finally we also have the pulse area which is

$$\Theta_0 = \int \Omega(t) dt. \tag{4.14}$$

We transform the system Hamilton into a frame rotating at the frequency of the optical transition [24],

$$H_S(t) = \frac{\mathcal{E}(t)}{2}\sigma^+ + \frac{\mathcal{E}^*(t)}{2}\sigma^- \tag{4.15}$$

where

$$\mathcal{E}(t) = \Omega(t) e^{-i\Delta(t)t}, \tag{4.16}$$

where $\Delta(t) = \omega_c(t) - \omega_{\uparrow\downarrow}$ is the detuning frequency.

4.2 Results

Here we present the results. The optimisation was performed by my colleague Gerald Fux as part of our collaboration on Ref. [24].

We model an SLM that has 512 pixels. Additionally we also have access to the pulse length τ, the initial pulse detuning Δ and the pulse area This gives 515 potential optimisation parameters. We assume that the spatial spot size covers two pixels, so

$$\delta\Omega_x = 2\delta\Omega_p \tag{4.17}$$

4.3 State transfer

Consider the case of a dot driven from its ground state $|\downarrow\rangle$ to the $|y^+\rangle = (|\uparrow\rangle + i\,|\downarrow\rangle)$ state. In our first example we simplify the problem by reducing our optimisation problem to a

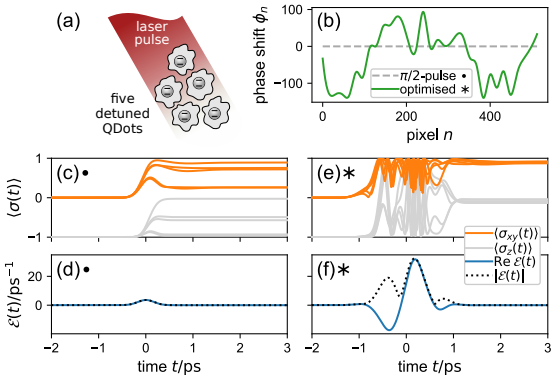


Figure 4.3: Optimal control of an ensemble of five quantum dots. (a) A schematic of the problem, 5 individual QDs with individual environments driven by a single laser. (b) The optimal phase mask as well as the $\pi/2$ protocol. Dynamics and electric field for the $\pi/2$ protocol (c–d) and the optimal protocol (e–f). Definitions of $\langle\sigma_{xy}\rangle$ given in main text. Figure reproduced from Ref. [24].

two-dimensional problem. We do this by parametrising the pulse as follows: the i^{th} phase delay φ_i for a corresponding frequency ω_i is given by

$$\varphi_i(\omega_i) = \Phi - 1300\omega_i^2, \tag{4.18}$$

which is a downward facing parabola controlled by a central phase shift Φ, which is our first optimisation parameter. A parabola of phase delays in frequency space gives linear temporal chirp (and also a broadened pulse [118]) that starts blue detuned and ends red detuned. The second control parameter is the detuning Δ of the initial pulse. We bound our control parameters as $\Delta \in [-50, 50]$ ps and $\Phi \in [-\pi, \pi]$.

Figure 4.2 shows the optimisation results. from the objective function landscape we see two local minima, marked by a star and a diamond respectively. The starred solution corresponds to a laser pulse that starts strongly detuned and finishes on resonance. This pulse is similar to that seen in adiabatic rapid passage [118]. It has the advantage of being insensitive to variations in Θ_0 but sensitive to variations in the detuning. The pulse corresponding to the diamond parameters starts on resonance and ends strongly red detuned. This means that the fidelity of the protocol is robust towards variations in detuning, but sensitive to changes in Θ_0 [24].

4.4 Optimisation of an ensemble

As a second example we consider a set of five individually detuned quantum dots. We assume that each of these dots couples to a separate environment, i.e. phonons emitted by one dot will not interfere with another dot. All five dots are driven by the same laser, and we seek an optimal protocol that will drive them to the equator of the Bloch sphere. The detuning of the individual quantum dots relative to the middle dot is chosen to be $[-10, 5, 0, 5, 10]$ ps respectively.

We reduce the 512 pixel SLM to 32 optimisation parameters by defining 32 points and fitting a cubic spline to interpolate between the optimisation points. Additionally we also optimise over the input pulse duration τ, detuning Δ, and pulse area Θ_0. The reason we use a cubic spline rather than fixing 16 pixels to have the same value and choosing that as

our 32 optimisation points is that having a large pixel creates a series of top-hat functions. The Fourier transform of a top-hat from the frequency domain to the time domain will yield a sinc function, which will broaden the pulse and give the aforementioned long time effects we don't want.

We expect a short $\frac{\pi}{2}$ pulse to drive the dot close to the equator of the Bloch sphere. Short because the shorter the pulse is, the broader the frequency space distribution, which lessens the dependence of the success of the operation on detuning frequency. This was taken as our initial guess. We used differential evolution to find optimal controls to drive the ensemble of quantum dots. This is shown in figure 4.3. The expectation value that is plotted in Figs. 4.3 (c) and (e) is

$$\langle \sigma_{xy}(t) \rangle = \sqrt{\langle \sigma_x(t) \rangle^2 + \langle \sigma_y(t) \rangle^2}. \tag{4.19}$$

The optimiser found a novel protocol that leads to a root mean square distance to the equator of the Bloch sphere of 0.1. This is in contrast with the performance of the $\frac{\pi}{2}$ pulse with the same initial pulse duration of $\tau = 245\,\text{fs}$. This protocol also compares well with the shortest $\frac{\pi}{2}$ pulse where $\tau = 30\,\text{fs}$, however we note that the the optimal protocol is not as robust to variations in the detuning of the individual quantum dots.

4.5 Relationship with previous work and conclusions

In contrast with our later work presented in chapters 2, 3 and 5, we simulate pulse shaping using a common experimental setup, and in doing so we ensure that every protocol discovered is a physical one. While there are obviously some deficiencies in our approach, such as assuming the SLM pixels are perfectly sharp with no gaps, or only considering acoustic phonons and hope we do not stray into a regime where other decoherence mechanisms are relevant, it nevertheless is a good approximation of what an experimentalist would expect to observe were they to perform performed closed loop feedback control on an InGaAs quantum dot.

An obvious criticism of optimising the pulse shape in the time domain and using every time-step as an optimisation point is that it is not guaranteed that the control field could be generated experimentally. However the main disadvantage of the SLM method is due to the non-trivial mapping between control parameters in this chapter (SLM phase delays) and the pulse shape, we believe the objective function landscape is far more pathological. As such for anything more than a two dimensional optimisation problem, it is possible for the optimiser to get stuck at a local minimum rather than the global minimum (hence the use of a global optimisation algorithm rather than a local one). This work was carried out prior to obtaining the gradient of the objective function using the adjoint method. Were we to re-examine this work, we could partly reduce the presence of local minima by considering a less restrictive parametrisation (such as optimising all 512 SLM pixels, or perhaps even more). However frequency space optimisation still remains a difficult problem, as, otherwise broadly similar pulses in the time domain may require drastically different $\varphi(\omega)$ functions.

In the other chapters we are not trying to directly mimic an experiment, we are examining the performance speed trade-off. The problem is set up in such a way that we do not expect there to be many local minima near the quantum speed limit. So in order to learn about the nature and mechanisms for non-Markovian decoherence and not to deal with distractions

as a result of our pulse shaper, we sacrificed the SLM pulse shaper. Nevertheless in this chapter we have identified non-trivial protocols to perform state transfer and show that it is possible to use an SLM pulse shaper to ensure physical pulses, and describe some practical considerations in doing so.

Chapter 5

Optimal control of quantum gates

So far we have with the problem of state transfer, given an initial state $|\psi\rangle$, how do I design control fields in order to take my system to state $|\sigma\rangle$. Here we present some preliminary work on how to perform optimisation to implement a quantum gate U.

5.1 Introduction

In order to use optimal control to identify control protocols to implement a quantum gate in an open quantum system we need two functions: a cost function, and the gradient of the cost function with respect to the control parameters. Previously, we have derived an efficient method for performing optimisation of state transfer by forward-propagating the initial state and back-propagating the target state. While considering state transfer, this method yields a simple representation for the cost function, Eq. 2.6, and an efficient method to compute the gradient of this cost function with respect to the control parameters.

The issue is that while constructing a quantum gate, there is no singular initial state and no singular final nor target state. Instead we have to deal with a set of possible initial states (e.g. for a two level system this would be any state on the surface of the Bloch sphere[1]), and the average fidelity between the set of achieved final states, and expected final states.

One possible solution to approximate a quantum gate function of merit could be to take the average gate fidelity of a finite number N of discrete states ρ_i, that are evenly spread about the surface of the Bloch sphere. Since this strategy involves computing fidelities for each ρ_k, where $k \in i$, it is easy to obtain the gradient with respect to that initial state. To do this, we can simply back-propagate the state that applying the unitary U to ρ_k would give, thus giving the gradient. The issue with this approach is that each evaluation of the gate cost and gradient function require N forward-propagations and back-propagations, making it N times more costly than performing optimisation of state transfer.

5.1.1 Fidelity of quantum gates

Fortunately, Pedersen, Møller and Mølmer [42] derive a formula for the average gate fidelity of a quantum operation. Reference [42] derive three useful properties. Firstly, for a unitary

[1] See ref. [42] as to why it is the surface of the Bloch sphere and not the entire volume of the Bloch sphere that matters

operator O of dimension n, the average of $|\langle\psi|O|\psi\rangle|^2$ ov states $|\psi\rangle$ on the unit sphere S is

$$\int_S |\langle\psi|O|\psi\rangle|^2 \, dV = \frac{1}{n(n+1)} \left(\mathrm{Tr}\left[OO^\dagger\right] + |\mathrm{Tr}[O]|^2 \right), \tag{5.1}$$

where dV is the normalised measure of volume of the sphere. Secondly, the fidelity between a unitary operation U and a target unitary operation U_t, is given by

$$\mathcal{F} = \int_S |\langle\psi|M|\psi\rangle|^2 \, dv \tag{5.2}$$

$$= \frac{1}{n(n+1)} \left(\mathrm{Tr}\left[MM^\dagger\right] + |\mathrm{Tr}[M]|^2 \right), \tag{5.3}$$

where $M = U_t^\dagger U$.

Finally Ref. [42] derive the average fidelity of a quantum gate. Recall the Kraus representation theorem, that any CPTP dynamical map $\mathcal{E}$ can be expressed as

$$\mathcal{E}\rho = \sum_\alpha K_\alpha \rho K_\alpha^\dagger,$$

where for simplicity we assume that we are using the canonical Kraus representation[2], such that $\sum_\alpha K_\alpha^\dagger K_\alpha = \mathbb{1}_n$. Now the fidelity between a given dynamical map described by a set of Kraus operators $\{K_\alpha\}$ and a target unitary U_t, the fidelity is given by

$$\mathcal{F} = \sum_\alpha \int_S |\langle\psi|M|\psi\rangle|^2 \, dv \tag{5.4}$$

$$= \frac{1}{n(n+1)} \left(\mathrm{Tr}\left[\sum_\alpha M_\alpha M_\alpha^\dagger\right] + \sum_\alpha |\mathrm{Tr}[M_\alpha]|^2 \right), \tag{5.5}$$

where $M_\alpha = U_t^\dagger K_\alpha$.

Eq. 5.4 gives us the fidelity of a quantum gate in terms of the Kraus operators without approximation. The most attractive feature of Eq. 5.4 is how easy it is to compute. For a general operation, the maximum number of Kraus operators required to specify a quantum channel is the system Hilbert space dimension squared, so for a two level system, Eq. 5.4 involves the multiplication of thirty two (2×2) matrices. We will address how to take the gradient later, but the obvious problem with this is that the process tensor is reduced to a Liouville super-operator, not a Kraus operator.

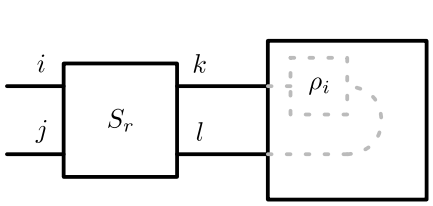
5.1.2 Transforms between representations of quantum channels

We now show how to transform between a Liouville super-operator and a Kraus operator. This section is based on the graphical calculus outlined in Wood et al. [39]. Here we re-introduce the assumptions described in the tensor network introduction. In order to transform a Liouville super-operator into the Kraus representation of a dynamical map, we must first convert the super-operator into a Choi matrix, from which we can perform a SVD to obtain the Kraus operators.

From super-operator to Choi matrix

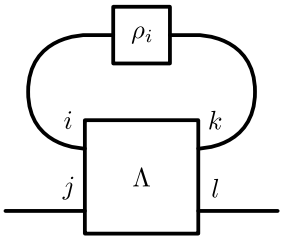
Converting a super-operator to a Choi matrix is simply a re-ordering of the indices. This can be easiest understood graphically in Fig. 5.1. In Fig. 5.1 (a) we see how to get a

[2]Eq. 5.4 holds for dynamical maps that are not in the canonical Kraus representation, see ref [42] for proof.



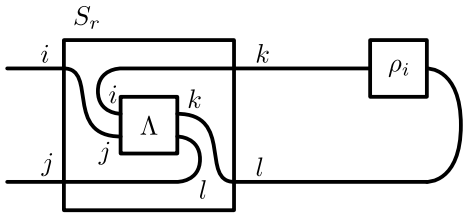
$$\mathcal{E}(\rho)^{ij} = S_r^{ijkl}\rho_i^{kl}$$

(a) Superoperator



$$\mathcal{E}(\rho)^{jl} = \Lambda^{ijkl}\rho_i^{ik}$$

(b) Choi Matrix



(c) Converting superoperator to Choi

Figure 5.1: Establishing the connection between applying a dynamical map in the super-operator representation (a) and the Choi representation (b). From there we can show how to reshape a super-operator into a Choi matrix (c). Diagram adapted from Ref. [39]

final state by applying a super-operator S_r (r denoting that it is row-vectorised) to a vectorised initial state ρ_i, while denoting the indices associated with both the evolution and the vectorisation. Fig. 5.1 (b) shows the equivalent while evolving using a Choi matrix Λ. From figure 5.1 (c) we can see how by moving the wires around and following the conventions outlined for vectorisation we can identify the scheme for converting between the two

Λ	$\rightarrow$	S_r		S_r	$\rightarrow$	Λ
i		k		i		j
j		i		j		l
k		l		k		i
l		j		l		k

From this we find that

$$\Lambda^{ijkl} = S_r^{kilj} \tag{5.6}$$

From Choi matrix to Kraus operators

Fig. 5.2 shows the application of a dynamical map in both the Choi and Kraus representation, demonstrating how to transition between the two. From this diagram we can see how we get the Kraus operators by grouping the two left and right indices together respectively, and performing the eigenvalue decomposition on this matrix. From there, we ungroup the two indices and bend the wire around ("unvectorise") the various eigenvectors. Each eigenvector now yields one Kraus operator. Since there are d_{sys}^2 eigenvectors, there are d_{sys}^2 Kraus operators.

Now we have described how to transform between a super-operator and a Kraus operator, we can now apply the average gate fidelity formula 5.4 to the process tensor. We do this by simply not specifying an initial state nor target state, contracting the entire PT to yield

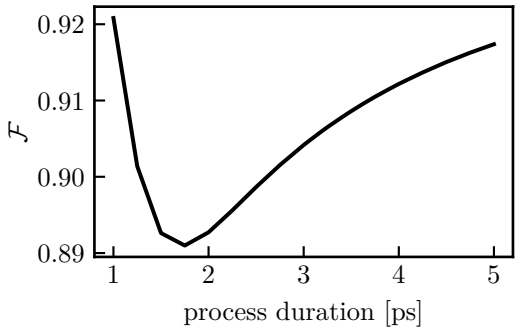


Figure 5.3: Fidelity vs process duration for a quantum gate initial guess.

a singular super-operator that propagates any given initial state to a final state. From there we perform the prescribed steps to obtain the Kraus operators

5.1.3 Back-propagating gates

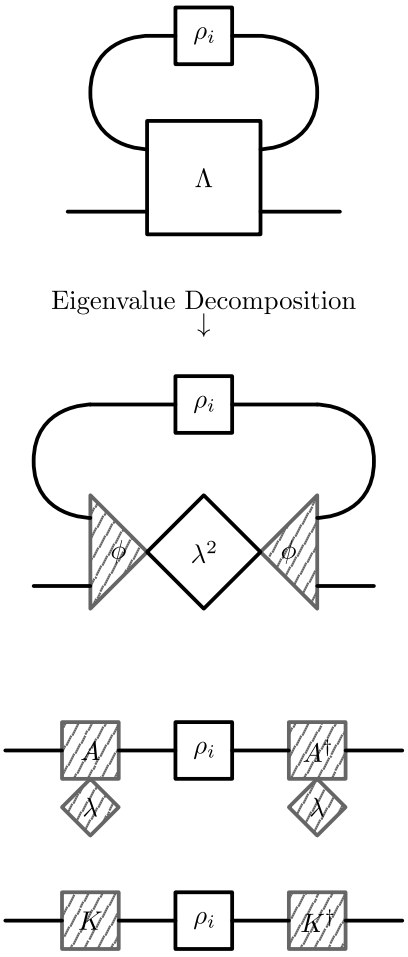


Figure 5.2: Performing Eigenvalue decomposition to obtain Kraus operators. Diagram adapted from Ref. [39]

So far we have described how to obtain the average gate fidelity. We have not yet derived how to back-propagate a quantum gate. Unfortunately, it is not as simple as back-propagating a generalised objective function as described in Sec. 2.3.1. The issue is: what is the

The chain rule Eq. 2.10,

$$\frac{\partial F}{\partial c_a} = \sum_n \sum_{i,j,k}^{d_{H_S}^2} \frac{\partial F}{\partial \rho_f^i} \frac{\partial \rho_f^i}{\partial U_n^{jk}} \frac{\partial U_n^{jk}}{\partial c_a}.$$

The issue lies in calculating $\frac{\partial F}{\partial c_a} = \sum_n \sum_{i,j,k}^{d_{H_S}^2}$, since our objective function depends on the dynamical map expressed in terms of Kraus operators, not the final state ρ_f. My proposed suggestion which is the subject of future work is that we should not convert the dynamical map obtained by PT-MPO into the Kraus representation, rather we should convert the gate fidelity formula Eq. 5.4 into the super-operator representation. Indeed Wood et al. [39] have a section on how this could be achieved.

5.2 Results

Nevertheless we still can compute the average gate fidelity. In Fig. 5.3, we show the fidelity for a target unitary of $U = S_x$ for an initial guess of

$$H = \frac{\pi}{t_f} S_x, \tag{5.7}$$

Which in a unitary case should yield a fidelity of one.

Chapter 6

Summary

In this thesis we have argued the case for the process tensor being the natural tool for optimisation of non-Markovian systems. By applying the adjoint method to the process tensor we obtain an efficient representation of the gradient of an optimisation problem. We contrasted this with our earlier work [24], where we instead used gradient-free global optimisation algorithm to search through parameter space. In that work we used an SLM in order to shape the pulse, and found the optimal SLM parameters to do so. This ensured that the optimal solutions were guaranteed to be experimentally achievable protocols.

We then used the adjoint method to seek optimal protocols to perform state transfer without any restrictions on pulse-shaping. In doing so we found that the optimiser naturally seeks uninteresting solutions where it attempts to say in decoherence-free subspaces, while driving the system using short and powerful pulses where the system is evolving so fast that the environment does not have time to react. In order to rectify this we introduce bounds on the maximum strength allowed for the control fields. This in turn introduces a bound-dependent maximum allowed speed, and draws us into a question that the QSL attempts to solve. We examine the success of state transfer for various different process durations, and ask what is the shortest time possible to perform state transfer, given a certain bound. We find that that slower operations yield higher fidelity — in contrast to what is generally expected for a Markovian theory. Because in an open system we cannot perform state transfer perfectly, we describe how there is no notion of a shortest time possible to perform state transfer, rather one should view it as a fidelity versus speed trade-off.

Optimisation using process tensors shows promise to be a broadly applicable technique. We indicate how this work could be extended to optimising quantum gates. We show an efficient method for obtaining quantum gate fidelity. With this one should be able to perform optimisation of quantum gates. Since one perform full counting statistics of heat transfer in the TEMPO algorithm [113], this work can be extended to the optimisation of thermal machines.

Appendices

Appendix A

Second order Trotterisation

A.1 Additional considerations for second order Trotterisation

A.1.1 Process Tensor

As mentioned in the introduction, there is minimal numerical penalty for considering the second-order Trotter splitting of the system-environment evolution, relative to the accuracy gain. However there are a few additional considerations for computing the gradient as in chapter 2. To recap, the first order Trotter splitting in terms of time is

$$e^{(\mathcal{L}_S+\mathcal{L}_E)\Delta t} = e^{\mathcal{L}_S\Delta t}e^{\mathcal{L}_E\Delta t} + \mathcal{O}(\Delta t^2). \tag{A.1}$$

The second order Trotter splitting is

$$e^{(\mathcal{L}_S+\mathcal{L}_E)\Delta t} = e^{\mathcal{L}_S\frac{\Delta t}{2}}e^{\mathcal{L}_E\Delta t}e^{\mathcal{L}_S\frac{\Delta t}{2}} + \mathcal{O}(\Delta t^3). \tag{A.2}$$

Eq. A.2 is indeed accurate to order $\mathcal{O}(\Delta t^3)$ provided that the system Hamiltonian is continuous over a whole time-step.

Fig. A.1 (a) shows the process tensor using second order Trotterisation. The brown boxes are the MPOs which are the same as the first order Trotterisation. The red boxes are the CP controls which are applied at the time that the expectation value is calculated (whole time-steps)[1]. Finally the system propagators are given by the light green and dark green circles. The light green one is the first half-propagator

$$U_n^l = \underset{\leftarrow}{T}\exp\left[\int_{t_s}^{t_s+\frac{\Delta t}{2}} dt'\mathcal{L}_S(t')\right] \tag{A.3}$$

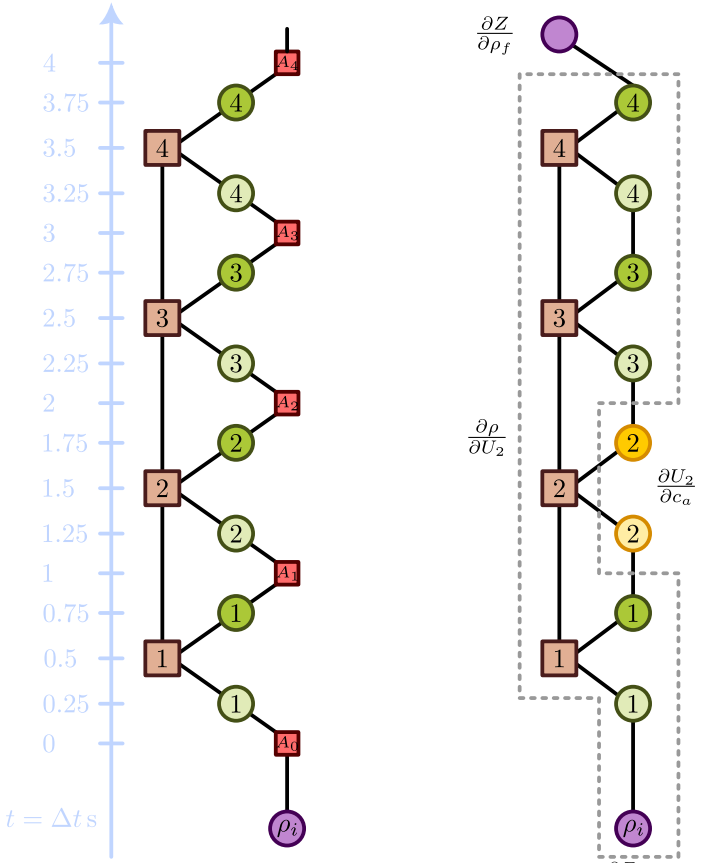
$$\approx e^{\mathcal{L}_S\left(t+\frac{\Delta t}{4}\right)\frac{t}{2}} \tag{A.4}$$

where $t_s \in \{0, \Delta t, 2\Delta t \ldots N\Delta t\}$ is the time of the current step. The dark green one is the second half-propagator

$$U_n^d = \underset{\leftarrow}{T}\exp\left[\int_{t_s+\frac{\Delta t}{2}}^{t_s+\Delta t} dt'\mathcal{L}_S(t')\right] \tag{A.5}$$

$$\approx e^{\mathcal{L}_S\left(t+\frac{3\Delta t}{4}\right)\frac{t}{2}} \tag{A.6}$$

[1] In `OQuPy` it is possible to apply a CP map before the state is recorded, as well as immediately after the state is recorded. This was omitted from diagram for clarity.



(a) Process Tensor (b) $\frac{\partial Z}{\partial c_a}$

Figure A.1: (a) Process tensor using second order Trotter splitting. MPOs in brown, first half propagator in light green, second half-propagator in dark green, system controls A in red. Light blue timescale is the times that the objects are applied at. (b) Gradient of an objective function with respect to a control parameter that only depends on the second propagator.

A.1.2 Gradient

We have stated that for the second order Trotter splitting to be accurate to order $\mathcal{O}(\Delta t^3)$, the system Hamiltonian must be constant over Δt. Careful inspection of our system parametrisation in chapters 2 and 3 will show that we ensured this by insisting that our Hamiltonian was constant over a time-step. It would also not be an issue were we to optimise the SLM configuration, provided that our resultant electric field was continuous. It is not a requirement for the Hamiltonian to be continuous between two time-steps.

It would not be a sensible idea to optimise $h_\bullet(t)$ over half half time-steps, as there is the possibility that the optimiser could introduce discontinuities. Additionally, expectation values are only valid over whole time-steps.

The gradient is shown in Fig. A.1 (b). In the case we describe in the main text where $\frac{\partial U_n}{\partial c(t_k)}$ is only non-zero for $n = k$, and we have Eq. 2.8 which we recall as

$$\frac{\partial \mathcal{F}}{\partial c_a} = \frac{\partial \mathcal{F}}{\partial U_k^{ij}} \frac{\partial U_k^{ij}}{\partial c_a}. \tag{A.7}$$

We now have two half propagators as seen in Fig. A.1 (b), we need to include

$$\frac{\partial U_k^l}{\partial c_a}, \ \frac{\partial U_k^d}{\partial c_a}, \tag{A.8}$$

where U_k^l is the first (light) half-propagator for the time-step k and U_k^d is the second (dark) half-propagator. In order to perform forward and back-propagation to get the gradient there are three ways to do so.

The first method, shown diagrammatically in Fig. A.2 involves storing a forward-propagated and back-propagated state before every single half system propagator. This is not a good idea as it means if there are N time-steps, there are $2N$ stored forward-propagated as well as $2N$ back-propagated states. It is especially bad as the stored states contain an index that is the internal bond dimension, therefore will quickly consume large amounts of RAM for a computationally challenging calculation. This is the method that was performed for the calculations shown in this thesis, but has since been superseded by the two other methods described. First we add in $\frac{\partial U_k^l}{\partial c_a}$ (Fig. A.2 (a)), and on the next half time step we add in $\frac{\partial U_k^d}{\partial c_a}$(Fig. A.2 (b)). $\frac{\partial Z}{\partial c_a}$ is the sum of these two scalars.

A better idea is to store the forward-propagated states after every MPO. This means that there are N states stored per propagation, which consumes half as much RAM as the previous algorithm. The gap where the system propagators are removed does not span across one time-step, rather it spans across two different time-steps. With the exception of the first case it contains the dark system half-propagator from one time step and the light system half-propagator from the next time step. As a result we fill in each gap twice, once with $\frac{\partial U_k^d}{\partial c_a}$ and U_{k+1} and a second time with U_k^d and $\frac{\partial U_{k+1}^l}{\partial c_a}$ (Figs. A.3 (a–b)). $\frac{\partial Z}{\partial c_a}$ is again obtained by summing the two contributions from *the same* time step (Figs. A.3 (b–c)).

An alternative way to recast this is by treating the two half-propagators as a single rank four tensor. This is shown in Fig. A.4. The stored tensors from the forward propagation are rank two tensors but during the chain rule step we must add in an extra MPO. However this means we do not have to sum the two contributions from the half-propagators. This is the method that is implemented in `OQuPy`.

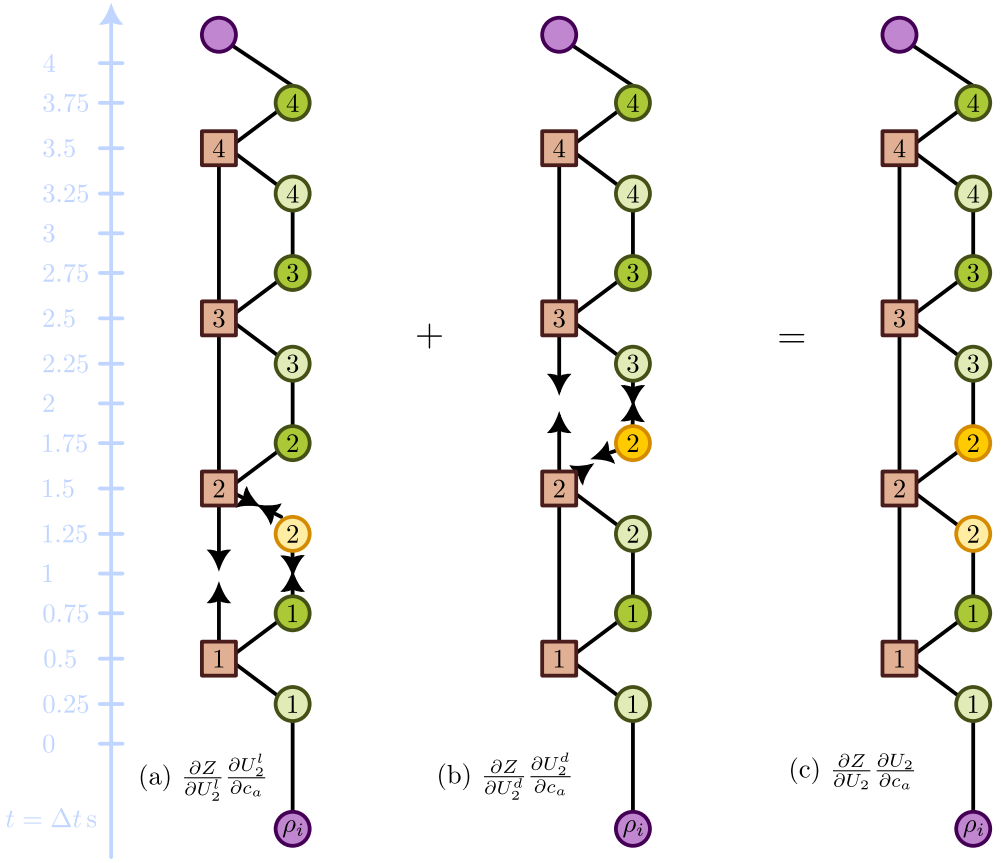


Figure A.2: Gradient by storing the state after each half-propagator

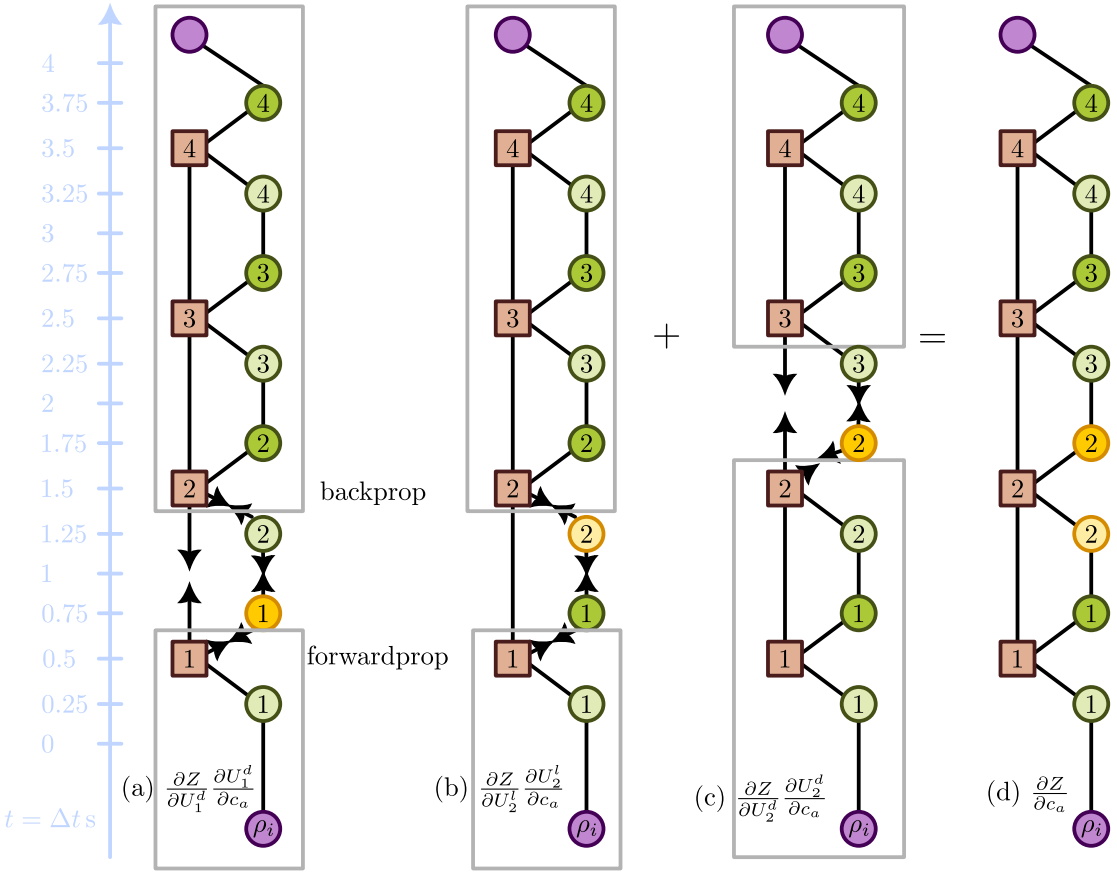


Figure A.3: Gradient by storing the state after each MPO

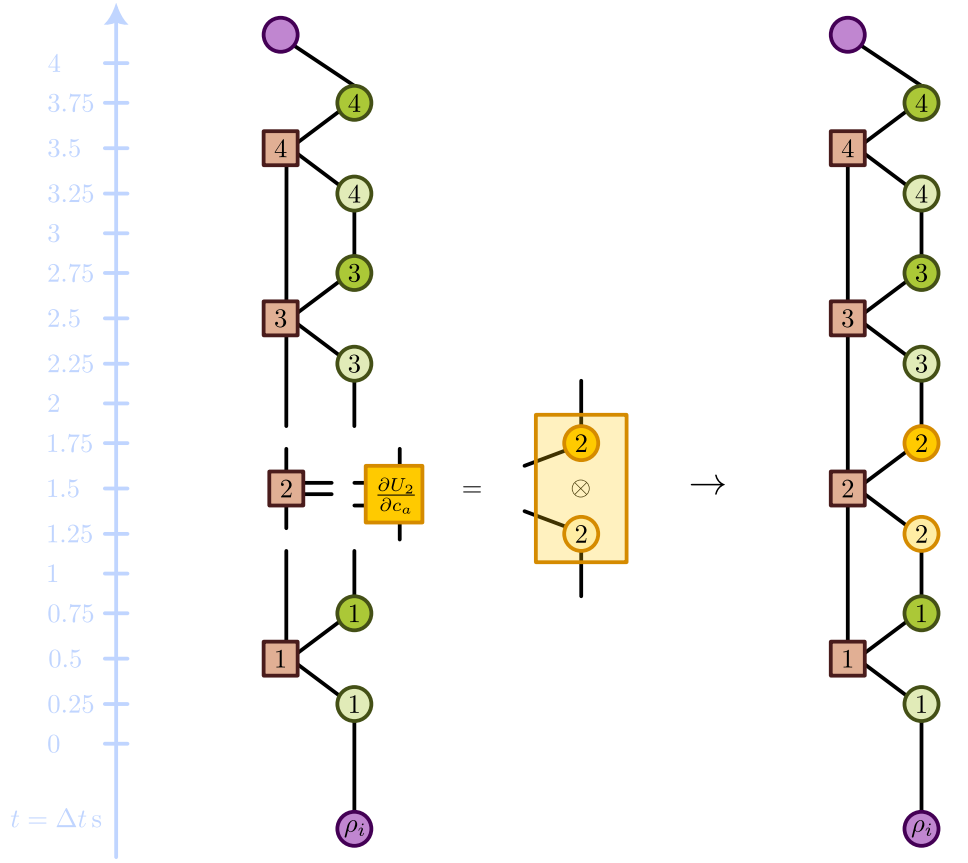


Figure A.4: Gradient by treating the half propagators as a rank four super-operator.

Bibliography

[1] P. Benioff, "The computer as a physical system: A microscopic quantum mechanical Hamiltonian model of computers as represented by Turing machines", Journal of Statistical Physics **22**, 563–591 (1980).

[2] C. H. Bennett and G. Brassard, "Quantum cryptography: Public key distribution and coin tossing", Theoretical Computer Science, Theoretical Aspects of Quantum Cryptography – Celebrating 30 Years of BB84 **560**, 7–11 (2014).

[3] C. P. Koch, U. Boscain, T. Calarco, et al., "Quantum optimal control in quantum technologies. Strategic report on current status, visions and goals for research in Europe", EPJ Quantum Technology **9**, 1–60 (2022).

[4] C. P. Koch, "Controlling open quantum systems: tools, achievements, and limitations", Journal of Physics: Condensed Matter **28**, 213001 (2016).

[5] S. J. Glaser, U. Boscain, T. Calarco, et al., "Training Schrödinger's cat: quantum optimal control", The European Physical Journal D **69**, 279 (2015).

[6] F. Arute, K. Arya, R. Babbush, et al., "Quantum supremacy using a programmable superconducting processor", Nature **574**, 505–510 (2019).

[7] A. Strathearn, *Modelling Non-Markovian Quantum Systems Using Tensor Networks*, Springer Theses (Springer International Publishing, Cham, 2020).

[8] F. A. Pollock, C. Rodríguez-Rosario, T. Frauenheim, M. Paternostro, and K. Modi, "Non-Markovian quantum processes: Complete framework and efficient characterization", Physical Review A **97**, 012127 (2018).

[9] I. de Vega and D. Alonso, "Dynamics of non-Markovian open quantum systems", Reviews of Modern Physics **89**, 015001 (2017).

[10] Á. Rivas, S. F. Huelga, and M. B. Plenio, "Quantum non-Markovianity: characterization, quantification and detection", Reports on Progress in Physics **77**, 094001 (2014).

[11] D. M. Reich, N. Katz, and C. P. Koch, "Exploiting Non-Markovianity for Quantum Control", Scientific Reports **5**, 12430 (2015).

[12] P. Rebentrost, I. Serban, T. Schulte-Herbrüggen, and F. K. Wilhelm, "Optimal Control of a Qubit Coupled to a Non-Markovian Environment", Physical Review Letters **102**, 090401 (2009).

[13] F. F. Floether, P. de Fouquieres, and S. G. Schirmer, "Robust quantum gates for open systems via optimal control: Markovian versus non-Markovian dynamics", New Journal of Physics **14**, 073023 (2012).

[14] B. Hwang and H.-S. Goan, "Optimal control for non-Markovian open quantum systems", Physical Review A **85**, 032321 (2012).

[15] E. Mangaud, R. Puthumpally-Joseph, D. Sugny, et al., "Non-Markovianity in the optimal control of an open quantum system described by hierarchical equations of motion", New Journal of Physics **20**, 043050 (2018).

[16] R. Schmidt, A. Negretti, J. Ankerhold, T. Calarco, and J. T. Stockburger, "Optimal Control of Open Quantum Systems: Cooperative Effects of Driving and Dissipation", Physical Review Letters **107**, 130404 (2011).

[17] Y. Ohtsuki, "Non-Markovian effects on quantum optimal control of dissipative wave packet dynamics", The Journal of Chemical Physics **119**, 661–671 (2003).

[18] J. Ying-Hua, H. Ju-ju, H. Jian-Hua, and K. Qiang, "Optimal control-based states transfer for non-Markovian quantum system", Physica E: Low-dimensional Systems and Nanostructures **81**, 77–82 (2016).

[19] W. Cui, Z. R. Xi, and Y. Pan, "Optimal decoherence control in non-Markovian open dissipative quantum systems", Physical Review A **77**, 032117 (2008).

[20] S. D. Mishra, R. Trivedi, A. H. Safavi-Naeini, and J. Vučković, "Control Design for Inhomogeneous-Broadening Compensation in Single-Photon Transducers", Physical Review Applied **16**, 044025 (2021).

[21] R.-E. Plessix, "A review of the adjoint-state method for computing the gradient of a functional with geophysical applications", Geophysical Journal International **167**, 495–503 (2006).

[22] F. A. Pollock, C. Rodríguez-Rosario, T. Frauenheim, M. Paternostro, and K. Modi, "Operational Markov Condition for Quantum Processes", Physical Review Letters **120**, 040405 (2018).

[23] M. R. Jørgensen and F. A. Pollock, "Exploiting the Causal Tensor Network Structure of Quantum Processes to Efficiently Simulate Non-Markovian Path Integrals", Physical Review Letters **123**, 240602 (2019).

[24] G. E. Fux, E. P. Butler, P. R. Eastham, B. W. Lovett, and J. Keeling, "Efficient Exploration of Hamiltonian Parameter Space for Optimal Control of Non-Markovian Open Quantum Systems", Physical Review Letters **126**, 200401 (2021).

[25] G. D. Mahan, "Exactly Solvable Models", in *Many-Particle Physics*, edited by G. D. Mahan, Physics of Solids and Liquids (Springer US, Boston, MA, 2000), pp. 187–294.

[26] D. Basilewitsch, J. Fischer, D. M. Reich, D. Sugny, and C. P. Koch, "Fundamental bounds on qubit reset", Physical Review Research **3**, 013110 (2021).

[27] H.-P. Breuer and F. Petruccione, *The Theory of Open Quantum Systems* (Oxford University Press, Jan. 25, 2007).

[28] A. K. Hansen, M. A. Sørensen, P. F. Staanum, and M. Drewsen, "Single-Ion Recycling Reactions", Angewandte Chemie International Edition **51**, 7960–7962 (2012).

[29] R. Rugango, J. E. Goeders, T. H. Dixon, et al., "Sympathetic cooling of molecular ion motion to the ground state", New Journal of Physics **17**, 035009 (2015).

[30] J. Clarke and F. K. Wilhelm, "Superconducting quantum bits", Nature **453**, 1031–1042 (2008).

[31] A. Blais, A. L. Grimsmo, S. M. Girvin, and A. Wallraff, "Circuit quantum electrodynamics", Reviews of Modern Physics **93**, 025005 (2021).

[32] A. J. Ramsay, "A review of the coherent optical control of the exciton and spin states of semiconductor quantum dots", Semiconductor Science and Technology **25**, 103001 (2010).

[33] A. J. Leggett, S. Chakravarty, A. T. Dorsey, et al., "Dynamics of the dissipative two-state system", Reviews of Modern Physics **59**, 1–85 (1987).

[34] L. Li, M. J. W. Hall, and H. M. Wiseman, "Concepts of quantum non-Markovianity: A hierarchy", Physics Reports, Concepts of Quantum Non-Markovianity: A Hierarchy **759**, 1–51 (2018).

[35] M. A. Nielsen and I. L. Chuang, *Quantum computation and quantum information*, 10th anniversary edition (Cambridge university press, Cambridge, 2010).
[36] Á. Rivas, S. F. Huelga, and M. B. Plenio, "Entanglement and Non-Markovianity of Quantum Evolutions", Physical Review Letters **105**, 050403 (2010).
[37] P. R. Eastham, A. O. Spracklen, and J. Keeling, "Lindblad theory of dynamical decoherence of quantum-dot excitons", Physical Review B **87**, 195306 (2013).
[38] C. Addis, E.-M. Laine, C. Gneiting, and S. Maniscalco, "Problem of coherent control in non-Markovian open quantum systems", Physical Review A **94**, 052117 (2016).
[39] C. J. Wood, J. D. Biamonte, and D. G. Cory, *Tensor networks and graphical calculus for open quantum systems*, (May 7, 2015) `http://arxiv.org/abs/1111.6950`.
[40] K. Kraus, *States, Effects, and Operations: Fundamental Notions of Quantum Theory* (Springer Berlin Heidelberg, Oct. 1983), 176 pp.
[41] I. Bengtsson and K. Zyczkowski, *Geometry of Quantum States: An Introduction to Quantum Entanglement* (Cambridge University Press, Cambridge, 2006).
[42] L. H. Pedersen, K. Molmer, and N. M. Moller, "Fidelity of quantum operations", Physics Letters A **367**, 47–51 (2007).
[43] A. Jamiołkowski, "Linear transformations which preserve trace and positive semidefiniteness of operators", Reports on Mathematical Physics **3**, 275–278 (1972).
[44] R. P. Feynman, A. R. Hibbs, and D. F. Styer, *Quantum Mechanics and Path Integrals* (Courier Corporation, July 21, 2010), 386 pp.
[45] J. W. Negele, *Quantum Many-particle Systems* (CRC Press, Mar. 5, 2018), 474 pp.
[46] R. P. Feynman, "Space-Time Approach to Non-Relativistic Quantum Mechanics", Reviews of Modern Physics **20**, 367–387 (1948).
[47] A. Altland and B. D. Simons, *Condensed Matter Field Theory*, 2nd ed. (Cambridge University Press, Cambridge, 2010).
[48] R. P. Feynman and F. L. Vernon, "The theory of a general quantum system interacting with a linear dissipative system", Annals of Physics **24**, 118–173 (1963).
[49] N. Makri and D. E. Makarov, "Tensor propagator for iterative quantum time evolution of reduced density matrices. II. Numerical methodology", The Journal of Chemical Physics **102**, 4611–4618 (1995).
[50] N. Makri and D. E. Makarov, "Tensor propagator for iterative quantum time evolution of reduced density matrices. I. Theory", The Journal of Chemical Physics **102**, 4600–4610 (1995).
[51] H. F. Trotter, "On the product of semi-groups of operators", Proceedings of the American Mathematical Society **10**, 545–551 (1959).
[52] A. Strathearn, P. Kirton, D. Kilda, J. Keeling, and B. W. Lovett, "Efficient non-Markovian quantum dynamics using time-evolving matrix product operators", Nature Communications **9**, 1–9 (2018).
[53] M. Suzuki, "Generalized Trotter's formula and systematic approximants of exponential operators and inner derivations with applications to many-body problems", Communications in Mathematical Physics **51**, 183–190 (1976).
[54] G. E. Fux, "Process tensor networks for non-Markovian open quantum systems", Thesis (The University of St Andrews, Nov. 29, 2022).
[55] R. Orús, "A practical introduction to tensor networks: Matrix product states and projected entangled pair states", Annals of Physics **349**, 117–158 (2014).
[56] G. White, F. Pollock, L. Hollenberg, K. Modi, and C. Hill, "Non-Markovian Quantum Process Tomography", PRX Quantum **3**, 020344 (2022).
[57] *OQuPy: Open Quantum Systems in Python*, Tempo Collaboration, Aug. 18, 2023.

[58] H.-P. Breuer, E.-M. Laine, and J. Piilo, "Measure for the Degree of Non-Markovian Behavior of Quantum Processes in Open Systems", Physical Review Letters **103**, 210401 (2009).
[59] E.-M. Laine, J. Piilo, and H.-P. Breuer, "Measure for the Non-Markovianity of Quantum Processes", Physical Review A **81**, 062115 (2010).
[60] S. Milz, F. A. Pollock, and K. Modi, "An Introduction to Operational Quantum Dynamics", Open Systems & Information Dynamics **24**, 1740016 (2017).
[61] S. Deffner and S. Campbell, "Quantum speed limits: from Heisenberg's uncertainty principle to optimal quantum control", Journal of Physics A: Mathematical and Theoretical **50**, 453001 (2017).
[62] L. Mandelstam and Ig. Tamm, "The Uncertainty Relation Between Energy and Time in Non-relativistic Quantum Mechanics", in *Selected Papers*, edited by I. E. Tamm, B. M. Bolotovskii, V. Y. Frenkel, and R. Peierls (Springer, Berlin, Heidelberg, 1991), pp. 115–123.
[63] J. Uffink, "The rate of evolution of a quantum state", American Journal of Physics **61**, 935–936 (1993).
[64] N. Margolus and L. B. Levitin, "The maximum speed of dynamical evolution", Physica D: Nonlinear Phenomena, Proceedings of the Fourth Workshop on Physics and Consumption **120**, 188–195 (1998).
[65] L. B. Levitin and T. Toffoli, "Fundamental Limit on the Rate of Quantum Dynamics: The Unified Bound Is Tight", Physical Review Letters **103**, 160502 (2009).
[66] W. K. Wootters, "Statistical distance and Hilbert space", Physical Review D **23**, 357–362 (1981).
[67] P. Pfeifer, "How fast can a quantum state change with time?", Physical Review Letters **70**, 3365–3368 (1993).
[68] S. Deffner and E. Lutz, "Energy–time uncertainty relation for driven quantum systems", Journal of Physics A: Mathematical and Theoretical **46**, 335302 (2013).
[69] P. J. Jones and P. Kok, "Geometric derivation of the quantum speed limit", Physical Review A **82**, 022107 (2010).
[70] P. M. Poggi, F. C. Lombardo, and D. A. Wisniacki, "Quantum speed limit and optimal evolution time in a two-level system", Europhysics Letters **104**, 40005 (2013).
[71] A. Uhlmann, "The "transition probability" in the state space of a $*$-algebra", Reports on Mathematical Physics **9**, 273–279 (1976).
[72] R. Jozsa, "Fidelity for Mixed Quantum States", Journal of Modern Optics **41**, 2315–2323 (1994).
[73] M. M. Taddei, B. M. Escher, L. Davidovich, and R. L. de Matos Filho, "Quantum Speed Limit for Physical Processes", Physical Review Letters **110**, 050402 (2013).
[74] S. Deffner and E. Lutz, "Quantum Speed Limit for Non-Markovian Dynamics", Physical Review Letters **111**, 010402 (2013).
[75] D. P. Pires, M. Cianciaruso, L. C. Céleri, G. Adesso, and D. O. Soares-Pinto, "Generalized Geometric Quantum Speed Limits", Physical Review X **6**, 021031 (2016).
[76] F. Campaioli, F. A. Pollock, F. C. Binder, and K. Modi, "Tightening Quantum Speed Limits for Almost All States", Physical Review Letters **120**, 060409 (2018).
[77] Z. Sun, J. Liu, J. Ma, and X. Wang, "Quantum speed limits in open systems: Non-Markovian dynamics without rotating-wave approximation", Scientific Reports **5**, 8444 (2015).

[78] A. del Campo, I. L. Egusquiza, M. B. Plenio, and S. F. Huelga, "Quantum Speed Limits in Open System Dynamics", Physical Review Letters **110**, 050403 (2013).
[79] F. Campaioli, F. A. Pollock, and K. Modi, "Tight, robust, and feasible quantum speed limits for open dynamics", Quantum **3**, 168 (2019).
[80] S. Deffner, "Geometric quantum speed limits: a case for Wigner phase space", New Journal of Physics **19**, 103018 (2017).
[81] D. Mondal and A. K. Pati, "Quantum speed limit for mixed states using an experimentally realizable metric", Physics Letters A **380**, 1395–1400 (2016).
[82] M. S. Byrd and N. Khaneja, "Characterization of the positivity of the density matrix in terms of the coherence vector representation", Physical Review A **68**, 062322 (2003).
[83] F. Nathan and M. S. Rudner, "Universal Lindblad equation for open quantum systems", Physical Review B **102**, 115109 (2020).
[84] M. Cygorek, M. Cosacchi, A. Vagov, et al., "Numerically exact open quantum systems simulations for arbitrary environments using automated compression of environments", Aug. 26, 2021.
[85] D. Gribben, A. Strathearn, G. E. Fux, P. Kirton, and B. W. Lovett, "Using the Environment to Understand non-Markovian Open Quantum Systems", Quantum **6**, 847 (2022).
[86] D. Gribben, D. M. Rouse, J. Iles-Smith, et al., "Exact Dynamics of Nonadditive Environments in Non-Markovian Open Quantum Systems", PRX Quantum **3**, 010321 (2022).
[87] G. E. Fux, D. Kilda, B. W. Lovett, and J. Keeling, "Tensor network simulation of chains of non-Markovian open quantum systems", Physical Review Research **5**, 033078 (2023).
[88] D. C. Reynolds and T. C. Collins, *Excitons: Their Properties and Uses* (Elsevier Science, Jan. 28, 1981), 312 pp.
[89] A. S. Bracker, D. Gammon, and V. L. Korenev, "Fine structure and optical pumping of spins in individual semiconductor quantum dots", Semiconductor Science and Technology **23**, 114004 (2008).
[90] D. J. Mowbray and M. S. Skolnick, "New physics and devices based on self-assembled semiconductor quantum dots", Journal of Physics D: Applied Physics **38**, 2059–2076 (2005).
[91] C. Murphy, "Quantum Control of Thermodynamic Processes in Semiconductors", Thesis (Trinity College Dublin).
[92] J. Duan, H. Huang, D. Jung, et al., "Semiconductor quantum dot lasers epitaxially grown on silicon with low linewidth enhancement factor", Applied Physics Letters **112**, 251111 (2018).
[93] Z. Yuan, B. E. Kardynal, R. M. Stevenson, et al., "Electrically Driven Single-Photon Source", Science **295**, 102–105 (2002).
[94] Y.-J. Wei, Y.-M. He, M.-C. Chen, et al., "Deterministic and robust generation of single photons from a single quantum dot with 99.5% indistinguishability using adiabatic rapid passage", Nano Letters **14**, 6515–6519 (2014).
[95] Y.-M. He, Y. He, Y.-J. Wei, et al., "On-demand semiconductor single-photon source with near-unity indistinguishability", Nature Nanotechnology **8**, 213–217 (2013).
[96] P. Chen, C. Piermarocchi, and L. J. Sham, "Control of Exciton Dynamics in Nanodots for Quantum Operations", Physical Review Letters **87**, 067401 (2001).

[97] T. Calarco, A. Datta, P. Fedichev, E. Pazy, and P. Zoller, "Spin-based all-optical quantum computation with quantum dots: Understanding and suppressing decoherence", Physical Review A **68**, 012310 (2003).

[98] H. Kamada and H. Gotoh, "Quantum computation with quantum dot excitons", Semiconductor Science and Technology **19**, S392 (2004).

[99] B. W. Lovett, A. Nazir, E. Pazy, et al., "Quantum computing with spin qubits interacting through delocalized excitons: Overcoming hole mixing", Physical Review B **72**, 115324 (2005).

[100] T. H. Stievater, X. Li, D. G. Steel, et al., "Rabi Oscillations of Excitons in Single Quantum Dots", Physical Review Letters **87**, 10.1103/PhysRevLett.87.133603 (2001).

[101] S. Lüker and D. E. Reiter, "A review on optical excitation of semiconductor quantum dots under the influence of phonons", Semiconductor Science and Technology **34**, 063002 (2019).

[102] A. J. Ramsay, A. V. Gopal, E. M. Gauger, et al., "Damping of Damping of Exciton Rabi Rotations by Acoustic Phonons in Optically Excited InGaAs/GaAs Quantum Dots", Physical Review Letters **104**, 017402 (2010).

[103] A. J. Ramsay, T. M. Godden, S. J. Boyle, et al., "Phonon-Induced Rabi-Frequency Renormalization of Optically Driven Single InGaAs/GaAs Quantum Dots", Physical Review Letters **105**, 177402 (2010).

[104] A. Nazir, "Photon statistics from a resonantly driven quantum dot", Physical Review B **78**, 10.1103/PhysRevB.78.153309 (2008).

[105] S. Lüker, K. Gawarecki, D. E. Reiter, et al., "Influence of acoustic phonons on the optical control of quantum dots driven by adiabatic rapid passage", Physical Review B **85**, 121302 (2012).

[106] J. Nocedal and S. J. Wright, *Numerical Optimization*, Springer Series in Operations Research and Financial Engineering (Springer New York, 2006).

[107] N. Khaneja, T. Reiss, C. Kehlet, T. Schulte-Herbrüggen, and S. J. Glaser, "Optimal control of coupled spin dynamics: design of NMR pulse sequences by gradient ascent algorithms", Journal of Magnetic Resonance **172**, 296–305 (2005).

[108] Y. Dauphin, R. Pascanu, C. Gulcehre, et al., *Identifying and attacking the saddle point problem in high-dimensional non-convex optimization*, (June 10, 2014) http://arxiv.org/abs/1406.2572.

[109] G. C. Hegerfeldt, "Driving at the Quantum Speed Limit: Optimal Control of a Two-Level System", Physical Review Letters **111**, 260501 (2013).

[110] M. Larocca, P. M. Poggi, and D. A. Wisniacki, "Quantum control landscape for a two-level system near the quantum speed limit", J. Phys. A: Math. Theor. **51**, 385305 (2018).

[111] E. P. Butler, G. E. Fux, C. Ortega-Taberner, et al., "Optimizing Performance of Quantum Operations with Non-Markovian Decoherence: The Tortoise or the Hare?", Physical Review Letters **132**, 060401 (2024).

[112] T. Caneva, M. Murphy, T. Calarco, et al., "Optimal Control at the Quantum Speed Limit", Physical Review Letters **103**, 240501 (2009).

[113] M. Popovic, M. T. Mitchison, A. Strathearn, et al., "Quantum Heat Statistics with Time-Evolving Matrix Product Operators", PRX Quantum **2**, 020338 (2021).

[114] A. W. Chin, S. F. Huelga, and M. B. Plenio, "Quantum Metrology in Non-Markovian Environments", Physical Review Letters **109**, 233601 (2012).

[115] T. Oksenhendler and N. Forget, "Pulse-Shaping Techniques Theory and Experimental Implementations for Femtosecond Pulses", in (Feb. 1, 2010).

[116] A. Weiner, D. Leaird, J. Patel, and J. Wullert, "Programmable shaping of femtosecond optical pulses by use of 128-element liquid crystal phase modulator", IEEE Journal of Quantum Electronics **28**, 908–920 (1992).

[117] A. M. Weiner, "Femtosecond pulse shaping using spatial light modulators", Review of Scientific Instruments **71**, 1929–1960 (2000).

[118] V. Malinovsky and J. Krause, "General theory of population transfer by adiabatic rapid passage with intense, chirped laser pulses", The European Physical Journal D - Atomic, Molecular, Optical and Plasma Physics **14**, 147–155 (2001).